UCRL-97200
PREPRINT

B.1.4

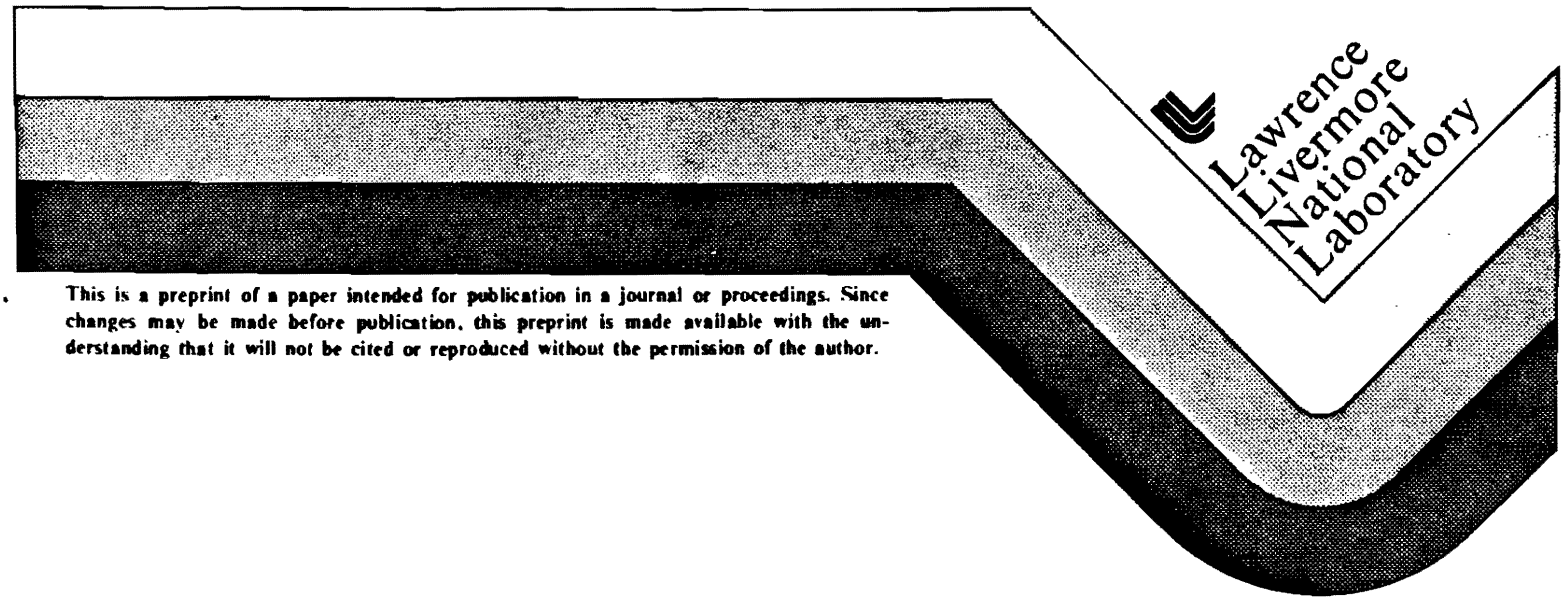
A CONSERVATIVE FRONT TRACKING METHOD FOR HYPERBOLIC CONSERVATION LAWS

I-Liang Chern
Institute of Mathematics
Academia Sinica

Phillip Colella
Lawrence Livermore National Laboratory

This paper was prepared for submittal to
Computational Physics
Journal

July, 1987



DISCLAIMER

A CONSERVATIVE FRONT TRACKING METHOD FOR HYPERBOLIC CONSERVATION LAWS

I-Liang Chern
Institute of Mathematics
Academia Sinica
Taipei, Taiwan, Republic of China

and

Phillip Colella
Lawrence Livermore National Laboratory
PO Box 808
Livermore, California 94550

July, 1987

Abstract

We present a hybrid front tracking / conservative finite difference method for computing discontinuous solutions to systems of hyperbolic conservation laws. In this method, the tracked front is allowed to move through the finite difference mesh, forming a free boundary at which the Rankine - Hugoniot conditions are applied as boundary conditions. The coupling to the finite difference method is done by performing a finite volume differencing in the irregular cells formed by the intersection of the finite difference grid wtih the regions on either side of the tracked front, leading to a method which is fully conservative up to and across the front. The CFL time step limitations which might come about from cell fractions on either side of the front being arbitrarily small are avoided by algebraically extending the range of influence of such volumes, with solution increments being redistributed into nearby cells in a volume-weighted way. Finally, the overall algorithm is designed to make minimum use of the global geometry of the tracked front; in particular, it can be coupled to a volume-of-fluid description of the tracked front. Numerical results using this method, with an unsplit second-order Godunov method for the conservative finite difference method, are presented for the case of gas dynamics in two space dimensions. The solutions computed are for the problem of self-similar shock reflection from an oblique surface, with the incident shock and the rigid shock tube wall treated as tracked fronts.

§ 1. Introduction

There are two approaches to the treatment of discontinuities in the numerical calculation of solutions to hyperbolic conservation laws. One is front capturing, where the discontinuities are represented as steep gradients spread over a small number of finite difference cells. The discrete divergence form of the difference equations and the addition of suitable dissipation operators are sufficient to insure that the solution converges to a weak solution of the conservation laws satisfying appropriate entropy conditions. The other approach is front tracking, where the discontinuity is treated as an interior free boundary coupled to a finite difference calculation for the smooth part of the solution on both sides of the discontinuity. The boundary conditions for the calculation on both sides of the discontinuity, as well as the evolution of the discontinuity itself, are given by the requirement that the solution satisfy the Rankine-Hugoniot relations across the discontinuity.

Each of the approaches has its advantages and disadvantages. The principal advantage of front capturing is its relative simplicity and its generality. Although a finite difference algorithm can, in general, be quite complicated, it is the same at all cells, and the finite difference cells themselves can have a simple logical structure. In spite of this simplicity, properly designed finite difference algorithms can resolve highly complicated patterns of discontinuities interacting with one another and with surrounding regions containing smooth solutions [33]. In contrast, tracking methods must contain mechanisms for predicting the formation and propagation of discontinuities, and their interaction with one another. Although considerable progress has been made in providing these mechanisms [3],[27], tracking methods by themselves have not been used successfully to solve problems of the same complexity as those accessible to capturing methods.

The advantages of front tracking are best understood by considering the limitations to front capturing. The accuracy with which strongly nonlinear discontinuities produced by systems of conservation laws are represented in a capturing algorithm depends on the establishment of a discrete traveling wave that represents the discontinuity, the existence of which in turn depends strongly on the regularity of the finite difference mesh. For example, in gas dynamics, one can easily construct meshes on which strong shocks are captured with unacceptably large errors, while smooth solutions, or even weak shocks, are calculated accurately [22],[32]. These problems disappear if the strong shock is tracked. Furthermore, there are problems in which front capturing fails on any mesh which resolves only the length scales associated with the hyperbolic waves, while front tracking gives acceptable results. An example of the latter are reaction fronts in gas dynamics, which can be treated being an infinitely thin discontinuity relative to the fluid dynamic length scales on either side, but whose jump relations depend on the details of the diffusive structure across the front. In that case, a capturing calculation on a mesh which resolves only the hydrodynamic length scales fails, since the numerical viscosity in the truncation errors in the method would vastly exceed the physical values, leading to incorrect wave speeds. This can also be the case even when the jump relations are independent of the diffusion coefficients, as long as the latter are sufficiently small, as is the case with detonation waves [10]. One approach is to use an adaptive grid to resolve the diffusive length scales by clustering mesh points at the front. However, if one does not need to resolve the details of the diffusive length scales on the multidimensional finite difference grid, but only the effects of the jump relations on the hydrodynamic scale flow, a tracking algorithm will treat these problems in a satisfactory fashion, since the discontinuity will be treated as a surface across which the appropriate jump relations are imposed explicitly.

In view of the above considerations, it is natural to attempt to combine the two approaches, tracking and capturing, in a single algorithm, tracking a few distinguished

discontinuities, while capturing the others. Such an idea is not new for compressible flow calculations. Indeed, the earliest numerical methods for gas dynamics were Lagrangian methods [19],[31], which automatically track contact discontinuities, while capturing shocks. This is a special case of a class of tracking methods where the tracked discontinuity coincides with some set of finite difference cell boundaries; such methods have also been used to track gas-dynamic shocks [17],[28]. The main disadvantage of these methods is that the mesh on either side of the tracked wave loses smoothness if there are large deformations of the tracked front. This can lead to a loss of stability and accuracy due to the distortions in the mesh, even for those modes which are only weakly coupled to the tracked discontinuity. An alternative to the tracking method described above is to allow the tracked discontinuity to move freely across a regular finite difference mesh. This type of tracking has been used extensively in calculations where all the discontinuities are tracked and the coupling of the tracked wave to the interior difference algorithm does not preserve conservation form ([3],[13],[20],[25],[27]). However, once the requirement of global conservation is imposed, the difficulty occurs that the tracked front generally divides cells into two pieces, one of which can be arbitrarily small, leading to the CFL time step approaching zero. The solution to this difficulty has been to locally rearrange the mesh geometry in the neighborhood of the front, by merging the small cell fragments with larger ones, for example, possibly giving up conservation for extremely small cell fragments [15],[21].

In this paper, we will follow the latter approach of allowing the tracked front to move through the mesh. The main new element in this work is our technique for dealing with the CFL restrictions arising from conservation at the front. In the previous work cited above, the CFL restrictions were circumvented by modifying the grid geometry in order to enlarge the range of influence of small control volumes adjacent to the tracked front. In the present work, we enlarge the range of influence algebraically by redistributing increments of the solution, as

given by the flux differences, in places near the front where the time step violates a CFL stability criterion. This flux difference redistribution is done in such a way so as to obtain a stable scheme which conserves globally. A major advantage of this method over the previous methods is that we need very little information about the geometry of the tracked front; in particular, this method can be used with a volume of fluid type description of the front. Such representations have proved useful in modeling unstable fronts such as material interfaces [7],[11],[16],[23] and flame fronts [5], where the volume of fluid description provides a simple and robust mechanism for representing large distortions and changes in topology of the tracked front.

§ 2. Front Tracking in One Dimension

We want to compute numerically solutions to the systems of conservation laws of the form

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=0 \qquad U(x,t)=U:\ \mathbf{R}\times[0,T]\to\mathbf{R}^N \tag{2.1}$$

where the system (2.1) is hyperbolic, i.e. the matrix $\nabla_U F = A(U)$ has N real eigenvectors $\lambda_1 \le \cdots \le \lambda_N$ with associated linearly independent sets of left and right eigenvectors $\{r_k\}_{k=1}^N, \{l_k\}_{k=1}^N$, $l_k \cdot r_{k'} = \delta_{kk'}$. We assume that the Riemann initial value problem for (2.1)

$$U(x,0) = \begin{cases} U_L & x<0 \\ U_R & x>0 \end{cases}$$

has a unique piecewise smooth solution $U(x,t)$ which is a function of the similarity variable $\frac{x}{t}$, and satisfies appropriate entropy conditions.

Our discrete approximation to the solution of (2.1) at time t^n is denoted by $U_j^n = \frac{1}{\Delta x}\int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} U(x,t^n)dx$, $j \in \mathbb{Z}$. In addition, in the $j^{n,J}$-th cell, there is a tracked front, i.e., a point $x^{n,J}$ dividing the $j^{n,J}$-th cell into two subintervals with lengths Λ_L^n, Λ_R^n to the left and right of $x^{n,J}$. The point $s^{n,J}$ is the approximate location of a discontinuity associated with the pth wave family at time step n. The average of U over each of these two subintervals is also given:

$$U_L^n = \frac{1}{\Lambda_L^n}\int_{(j-\frac{1}{2})\Delta x}^{x^{n,J}} U(x,t^n)dx$$

$$U_R^n = \frac{1}{\Lambda_R^n}\int_{x^{n,J}}^{(j+\frac{1}{2})\Delta x} U(x,t^n)dx .$$

We want to advance this description of the problem in time, calculating $x^{n+1,J}$ and the location of the front at time $t^{n+1} = t^n + \Delta t$, and U_L^n, U_R^n, the average value of the solution on either side of the front in the cell containing it, as well as the values of U_j^{n+1}, $j \neq j^{n+1,J}$. The combined algorithm should be conservative overall, i.e. $\sum_j U_j^{n+1} = \sum_j U_j^n + (F_L - F_R)\Delta t$, for some F_L, F_R. Away from the tracked front, the solution is to be completed with an explicit conservative finite difference algorithm of the form

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}(F_{j-1/2} - F_{j+1/2}) \tag{2.2}$$

$$F_{j+1/2} = F(U_{j-r}^n, \ldots, U_{j+r}^n). \tag{2.3}$$

Our strategy to do this will be to treat the tracked front as a moving interior boundary, using the solution to the Riemann problem to move the tracked front, and to obtain a flux boundary condition at the front. The fluxes obtained by this procedure and from the flux formula (2.2) are then differenced to obtain conservative increments to the solution. In the neighborhood of the front, these increments are redistributed to neighboring cells to avoid CFL stability restrictions on the time step in cases where one of $\Lambda_{L,R} \ll \Delta x$ holds.

In order to calculate the trajectory of the discontinuity of the pth wave family over the time step, we solve the Riemann problem for (2.1) with left and right states

$$\overline{U}_L = U_L^n \frac{\Lambda_L^n}{\Delta x} + U_{j^{n,J}+1}^n \left[1 - \frac{\Lambda_L^n}{\Delta x}\right]$$

$$\overline{U}_R = U_R^n \frac{\Lambda_R^n}{\Delta x} + U_{j^{n,J}+1}^n \left[1 - \frac{\Lambda_R^n}{\Delta x}\right].$$

We use these averaged values, rather than U_L^n, U_R^n, because our values for U_L^n, U_R^n, may not be very accurate if one of $\Lambda_{L,R}^n \ll \Delta x$. The only two pieces of data we require from the Riemann problem are s, the speed of the pth discontinuity, and $F^J = F(U) - s^J U$, the flux across that discontinuity which, according to the Rankine-Hugoniot discontinuity relations, is continuous at

the discontinuity. The location of the shock at $t^n+\Delta t$ is then given by $x^{n+1,f}=x^{n,f}+s^f\Delta t$.

To obtain the values of the solution at the new time for cells not containing $x^{n,f}$ or $x^{n+1,f}$, we use (2.2) - (2.3), except that for fluxes sufficiently close to the tracked front that the flux formula (2.3) would involve values of U on the other side of the front, we replace those values with $\overline{U}_L$ for fluxes to the left of the front, $\overline{U}_R$ for fluxes to the right of the front.

To update the cells containing the tracked front, there are two cases, either the tracked front crosses into the next cell ($j^{n,f}=j^{n+1,f}$), or not ($j^{n,f}=j^{n+1,f}\pm 1$). We consider first the case $j^{n,f}=j^{n+1,f}$.

We calculate

$$\delta M_L = (\Lambda_L^n-\Lambda_L^{n+1})U_L^n+\Delta t(F^n_{j+\frac{1}{2}}-F^f) \tag{2.4}$$
$$\delta M_R = (\Lambda_R^n-\Lambda_R^{n+1})U_R^n+\Delta t(F^f-F^f_{j+\frac{1}{2}})$$

The significance of these quantities is that they give the conservative update formula for U_L^{n+1},U_R^{n+1},

$$\Lambda_S^{n+1}U^{n+1}=\Lambda_S^{n+1}U_S^n+\delta M_S \qquad S=L,R. \tag{2.5}$$

obtained from differencing the fluxes at the cell edge and along the tracked front. The difficulty with this formula is that it is unstable for time steps satisfying the usual CFL condition $\max_{j,k}\frac{\Delta t}{\Delta x}|\lambda_k(U_j^n)|\le\sigma<1$. This is most easily seen from the fact that, to obtain U_S^{n+1} from (2.5). division by Λ_S^{n+1} is required, leading to a time step requirement that $\Delta t\le\min_{S=L,R}\frac{\Lambda_S^{n+1}}{\max_{k=1,\dots,N}\lambda_k(U_S^n)}$. This has long been recognized to be a problem with tracking methods of this type. The two ways which had been used to deal with this difficulty in the case of a general tracked front were: to give up conservation at the tracked front, or to rearrange the mesh geometry in the neighborhood of the tracked front so that the resulting $\Lambda_{L,R}^{n+1}$ are large

enough so as not to impose a catastrophic time step restriction. There is a third approach which has been used in the special case of a tracked material interface in gas dynamics, which involves differencing across the tracked front in a fashion which does not mix the two materials [11]. The algorithm which we describe here is in a generalization of the latter approach to the case of a general tracked front. In order to calculate $U_{L,R}^{n+1}$, we replace δM_S by $\delta M_S \eta_S = \frac{\Lambda_S^{n+1}}{\Delta x}$ in (2.5) to obtain

$$U_S^{n+1} = U_S^n + \frac{1}{\Delta x}\delta M_S. \tag{2.6}$$

This solves the stability problem, but in order to have global conservation, we must dispose of $(1-\eta_S)\delta M_S$, elsewhere on the grid. We do so by distributing these flux differences into nearby cells into which they can be absorbed without loss of stability. This redistribution is done based on a decomposition of the flux differences in terms of right eigenvectors, so that the various components of $(1-\eta_S)\delta M_S$ are distributed to where they are being propagated in the sense of characteristics, although possibly sooner than expected.

We expand

$$(1-\eta_S)\delta M_S = \sum_k \alpha_k r_k^S \ , \ r_k^S = r_k(\bar{U}_S) \tag{2.7}$$

and define

$$\begin{aligned}
\delta M_S^+ &= \sum_{k>p} \alpha_k r_k^S, \\
\delta M_S^- &= \sum_{k<p} \alpha_k r_k^S, \\
\delta M_S^0 &= \alpha_p r_p^S \ , \\
\delta M_+^{wi} &= \delta M_L^+ + \delta M_R^+ + \delta M_R^0, \\
\delta M_-^{wi} &= \delta M_L^- + \delta M_R^+ + \delta M_L^0.
\end{aligned} \tag{2.8}$$

We then modify the results obtained from (2.2),(2.6) to be

$$U^{n+1}_{j^{n+1}J+1} := U^{n+1}_{j^{n+1}J+1} + \frac{1}{\Lambda^{tot}_{+}}\delta M^{tot}_{+}$$

$$U^{n+1}_{R} := U^{n+1}_{R} + \frac{1}{\Lambda^{tot}_{+}}\delta M^{tot}_{+} \tag{2.9}$$

$$U^{n+1}_{j^{n+1}_{tr}-1}, U^{n+1}_{L} := U_{j^{n+1}_{tr}-1}, U^{n+1}_{L} + \frac{1}{\Lambda^{tot}_{-}}\delta M^{tot}_{-}$$

where

$$\Lambda^{tot}_{+} = \Lambda^{n+1}_{R} + \Delta x$$

$$\Lambda^{tot}_{-} = \Lambda^{n+1}_{L} + \Delta x,$$

and we use := to denote assignment in place. It is not difficult to check that (2.6) - (2.9) leads to an algorithm which globally conserves. The case where $j^{n+1J} = j^{nJ} + 1$ proceeds similarly, except that we must deal with the complication that the length of the cell fraction to the right of the tracked front contained in cell j^{nJ} is zero at the new time. If we define $\bar{F}_{j^{nJ}+1/2} = F(\bar{U}_L)$, then

$$U^{n+1}_{j^{n}J} = \frac{1}{\Delta x}(\Lambda^{n}_{L}U^{n}_{L} + \Delta t F_{j^{n}J-1/2} - (\tau F^{J} + (\Delta t - \tau)\bar{F}_{j^{n}J+1/2}))$$

$$\delta M_L = -\Lambda^{n+1}_{L}\bar{U}_L + (\Delta t - \tau)(\bar{F}_{j^{n}J+1/2} - F^{J})$$

$$\delta M_R = \Lambda^{n}_{r}U^{n}_{R} + \tau(F^{J} - F_{j^{n}J+1/2}) + (\Delta x - \Lambda^{n+1}_{R})U^{n}_{j^{n+1}J} + (\Delta t - \tau)(F^{J} - F_{j^{n+1}J+1/2})$$

$$U^{n+1}_{R} = U^{n}_{j^{n+1}J} + \frac{1}{\Delta x}\delta M_R$$

$$U^{n+1}_{L} = \bar{U}_L + \frac{1}{\Delta x}\delta M_L.$$

$$\tau = \frac{\Lambda^{n+1}_{L}}{s}$$

Then the redistribution algorithm proceeds as before, with (2.7)-(2.9) applied without modification.

There are a number of remarks to be made about the algorithm described above. The first is that, if the solution at time t^n is given by $U(x,t^n) = U_L$, $x < x^{nJ}$, $U(x,t^n) = U_R$, $x > x^{nJ}$

where U_L, U_R satisfy the Rankine-Hugoniot jump relations for a p-shock, then the solution produced by this algorithm at the new time is also given by the same two constant states separated by a jump at $x^{n+1/2}$ up to roundoff; in particular, $\delta M_L = \delta M_R = 0$. More generally, if the solution at time t^n consists of a k-shock separating smooth regions, then $\delta M_L, \delta M_R = 0(\Delta x \Delta t)$. Thus the algorithm redistributes flux differences representing the effect of the smooth wave interacting with the tracked discontinuity. by the same argument, it follows that the local error in the approximation is second order in the mesh spacing for a tracked discontinuity interacting with a smooth background, at least in the sense of distributions. That is, if U_ϵ is the exact solution to (2.1) with such initial data, then for Δt sufficiently small

$$\int_{(j_l-1/2)\Delta x}^{(j_r+1/2)\Delta x} U_\epsilon(x, \Delta t)\, dx - \Delta x \sum_{j=j_l}^{j_r} U_j^1 = O(\Delta x \Delta t) \tag{2.10}$$

$$j_l \Delta x \ , \ j_r \Delta x \to x_l \ , \ x_r \text{ as } \Delta x \ , \ \Delta t \to 0.$$

Where the discrete initial data is obtained by taking appropriate averages of the exact initial data over the discrete mesh. Formally, this follows from applying the divergence theorem to the integral of (2.1) over the rectangle $[(j_l - 1/2)\Delta x \ , \ (j_r + 1/2)\Delta x] \times [0 \ , \ \Delta t]$, comparing the exact and approximate fluxes at the endpoints, and the observation that the effect of the redistribution algorithm is $O(\Delta t \Delta x)$. Notice that this argument holds independent of whether the discontinuity crosses one of the endpoints. In contrast, there is no analogous estimate known for the exact and approximate fluxes for any shock capturing method for systems of equations, i.e. one that says that their averages differ by $O(\Delta x \Delta t)$ in the neighborhood of a captured shock; in fact, the numerical evidence in [22], [32] for the case of a gas-dynamic shock separating constant states is a counterexample, since the result is independent of Δx, Δt, depending only on their ratio.

A second comment is that the redistribution given above is not the only possible one. In the case where the tracked front is a linearly degenerate discontinuity, the redistribution algorithm given above reproduces in the discretization the qualitative behavior of the solution to the differential equation that information carried by the pth characteristic family does not cross the discontinuity. If the pth wave is genuinely nonlinear, then the increments $\delta M^0_{L,R}$ could, in principle, be placed on either side of the discontinuity, since in any case they would eventually be swept up by discontinuity due to the convergence of the characteristics of that family. This fact leaves one free to redistribute the fluxes based on other considerations. For example, consider the case of a gas-dynamic shock facing to the right. In that case, $N=p=3$, and $\delta M^+_R=0$. However, δM^0_R is, in general, nonzero, due either to smooth 3-wave approaching from the right or due to errors in the estimate of the shock speed. If the shock is propagating into a uniform low pressure ambient, then it is desirable to redistribute δM^0_R in back of the shock, since small perturbations of the preshock state (coming necessarily from errors in the shock speed) could produce negative pressures. We will discuss this point further in § 4.

§ 3. Front Tracking in Two Dimensions

We now want to extend the ideas in the previous section to yield algorithms for solving numerically

$$\frac{\partial U}{\partial t} + \frac{\partial F^x}{\partial x} + \frac{\partial F^y}{\partial y} = 0 \tag{3.1}$$

$$U(x,y,t) = U:\ \mathbf{R}^2 \times [0,T] \to \mathbf{R}^N.$$

We assume this system to be hyperbolic in the sense that the system (3.1) projected in the $\hat{n}$ direction

$$\frac{\partial V}{\partial t} + \frac{\partial F^{\hat{n}}}{\partial \xi}(V) = 0 \tag{3.2}$$

$$F^{\hat{n}}(V) = n_x F^x(V) + n_y F^y(V)$$

satisfies the conditions given for systems in one space dimension given in the previous section, for any choice of unit vector $\hat{n} = (n_x, n_y)$.

We will use a conservative finite difference algorithm for solving (3.1) away from the tracked front:

$$U_{ij}^{n+1} = U_{ij}^n + \frac{\Delta t}{\Delta x}(F^x_{i-\frac{1}{2},j} - F^x_{i+\frac{1}{2},j}) + \frac{\Delta t}{\Delta y}(F^y_{i,j-\frac{1}{2}} - F^y_{i,j+\frac{1}{2}}). \tag{3.3}$$

Here U_{ij}^n is the discrete approximation to the cell average of U on a uniform rectangular grid at time t^n

$$U_{ij}^n \approx \int_{\Delta_{ij}} U(x,y,t^n)\,dx\,dy,$$

$$\Delta_{ij} = [(i-\tfrac{1}{2})\Delta x, (i+\tfrac{1}{2})\Delta x] \times [(j-\tfrac{1}{2})\Delta y\,(j+\tfrac{1}{2})\Delta y]$$

$F^x_{i+\frac{1}{2},j}, F^y_{i,j+\frac{1}{2}}$ approximate the time averaged fluxes at the cell edges, and are assumed to be explicit functions of U^n of the form

$$
\begin{aligned}
F^x_{i+\frac{1}{2},j} &= F^x(U^n_{i,j-1},\ldots,U^n_{i+1,j+1};\,(D_x^- U)_{i-r,j-r},\\
&\qquad (D_y^- U)_{i-r,j-r},\ldots,(D_x^- U)_{i+r,j+r},(D_y^- U)_{i+r,j+r}) \qquad (3.4)\\
F^y_{i,j+\frac{1}{2}} &= F^y(U^n_{i-1,j},\ldots,U^n_{i+1,j+1};\,(D_x^- U)_{i-r,j-r},\\
&\qquad (D_y^- U)_{i-r,j-r},\ldots,(D_x^- U)_{i+r,j+r},(D_y^- U)_{i+r,j+r})
\end{aligned}
$$

where $(D_x^- U)_{i,j} = U^n_{i,j} - U^n_{i-1,j}$, $(D_y^- U)_{i,j-1} = U^n_{i,j} - U^n_{i,j-1}$. In other words, we assume that the dependence of $F^x_{i+\frac{1}{2},j}, F^y_{i,j+\frac{1}{2}}$ on U^n has the form of a general dependence for the 6 cells nearest the cell edge where the flux is defined, plus a possible dependence on values of U^n farther away which appears only as one sided differences in U^n. Clearly, such a representation is not unique, since dependences on differences of values at the 6 nearest neighbors can be absorbed in both the first or second sets of arguments. However, we will also restrict the representations considered in (3.4) to be of one of two types. We require the algorithm obtained either by setting in (3.4) an arbitrary subset of the $D_x^- U, D_y^- U$ arguments to zero, or by setting all of the $D_x^- U, D_y^- U$ arguments to zero, to be a stable algorithm which is at least first order accurate. The class of algorithms which satisfy our requirements is a fairly broad one, including the two-step Lax-Wendroff algorithm [25], the Flux-Corrected Transport algorithms in [34], and Godunov's method [14] and its second order extensions [6],[30]. We are restricting our attention to unsplit algorithms, for reasons we will discuss in § 6.

We assume that our tracked front divides our computational domain into two components labeled 1 and 2. At time t^n, we assume that, in each cell, we know the area $\Lambda^{n,l}_{i,j}$, $l = 1,2$, $\Lambda^{n,1}_{i,j} + \Lambda^{n,2}_{i,j} = \Delta x \Delta y$ of each of the two components along with $U^{n,l}_{i,j}$, the value of the solution for that component in that cell. If $\Lambda^{n,l}_{i,j}$ is zero, then there is no value assigned to $U^{n,l}_{i,j}$. We then wish to calculate the updated values of the areas $\Lambda^{n+1,l}_{i,j}$, and of the solution $U^{n+1,l}_{i,j}$. In this section, we will assume that $\Lambda^{n+2,l}_{i,j}$ is known, and give the algorithm for calculating $U^{n+1,l}_{i,j}$. The reason for this is that what constitutes an acceptable choice of algorithm for updating $\Lambda^{n+1,l}_{i,j}$ depends strongly on the type of front being tracked, whereas the extension of the algorithms

discussed in the previous section is independent of the method of calculating $\Lambda_{i,j}$. In particular, we will assume that, in each cell, $\Lambda_{i,j}^{n,j}, \Lambda_{i,j}^{n+1,j}$ are given, in addition to $\hat{n}_{i,j}$, a value for the unit normal to the front, pointing from 2 to 1, as well as $\tilde{s}_{i,j}$, an estimate of the velocity of the front in the direction of $\hat{n}_{i,j}$, the latter two quantities being specified for mixed cells, i.e., ones for which both $\Lambda_{i,j}^{\ell}$ are nonzero for one of the time levels. Given this information, our strategy for calculating $U_{i,j}^{n+1,j}$ will be to reconstruct in each mixed cell a local approximation to the geometry in space-time of the tracked front; to use this information, along with the difference algorithm (3.3)-(3.4) to calculate increments to the solution analogous to (2.4) in one dimension; and to distribute those increments in a way that has a minimum impact on the accuracy of the solution away from the tracked front, while satisfying the requirements of stability and global conservation form.

Given $\Lambda_{i,j}^{n,j}, \Lambda_{i,j}^{n+1,j}$, we can find a plane S_0 in $(\vec{x},t)$ space of the form $S_0 = \{(\vec{x},t): \vec{x}\cdot\hat{n} - s\,t + a = 0\}$ which represents locally the trajectory of the tracked front through the finite difference cell $\Delta_{i,j}$. If we define

$$S_1 = \{(\vec{x},t): \vec{x}\cdot\hat{n} - st + a > 0\}$$

$$S_2 = \{(\vec{x},t): \vec{x}\cdot\hat{n} - st + a < 0\}$$

then s and a can be chosen so that

$$\Lambda_{i,j}^{n,j} = \text{area of } S_l \cap \Delta_{i,j} \times \{t^n\}$$

$$\Lambda_{i,j}^{n,j} = \text{area of } S_l \cap \Delta_{i,j} \times \{t^{n+1}\}$$

$$l = 1,2.$$

In the case that $\Lambda_{i,j}^{n,j}, \Lambda_{i,j}^{n+1,j}$ are all nonzero S_0 will be uniquely determined from the Λ's and $\hat{n}$. In the case when one of the Λ's is zero, then the speed $\tilde{s}$ is required to determine S_0.

The only information we will need about S_0 are s and the areas of the surfaces in space-time on which fluxes of each of the components are imposed. We define $A_{i,j}^{q,j}$ to be the area of

$(S_l \cap L_q \times [t^n, t^n+\Delta t])$, $q = 1, \ldots, 4$ where the L_q are the sides of $\Delta_{i,j}$ listed in clockwise order, starting from $(i-\frac{1}{2}, j-\frac{1}{2})$. We also define

$$A^l_{i,j} = \text{area of } S_0 \cap \Delta_{i,j} \times [t^n, t^n+\Delta t].$$

The calculation of $A^{q,l}, A^l$ is a straightforward exercise in trigonometry, and will be sketched in an appendix. However, we do note here the consistency requirements $A^{p,1}+A^{p,2}=\Delta x \Delta t$ if $p=1,3$, $=\Delta y \Delta t$ if $q=2,4$. We also mention that A^l is easily calculated from the A^p's and Λ's using the divergence theorem.

We now can define our finite difference fluxes. First, if $h_{i,j}$ is a variable defined on the grid, we denote by

$$\sum_{\text{adj}(i,j)} h = \sum_{\substack{-1 \le r_1, r_2 \le 1 \\ (r_1, r_2) \neq (0,0)}} h_{i+r_1, j+r_2}.$$

Using this notation, we define an averaging procedure related to the redistribution algorithm. We define

$$\bar{U}^l_{i,j} = \left[\frac{\sum_{\text{adj}(i,j)} \Lambda^l U^{n,l}}{\sum_{\text{adj}(i,j)} \Lambda^{n,l}} \right] \left[1 - \frac{\Lambda^{n,l}_{i,j}}{\Delta x \Delta y} \right] + U^{n,l}_{i,j} \frac{\Lambda^{n,l}}{\Delta x \Delta y}. \tag{3.5}$$

Then for all edges which are adjacent to cells containing component l at either the old or new time, we define $F^{x,l}_{i+\frac{1}{2},j}, F^{y,l}_{i,j+\frac{1}{2}}$ to be values obtained from the formulas (3.4) with $U^n_{i,j}$ replaced by

$$\begin{aligned} \tilde{U}^l_{i,j} &= U^{n,l}_{i,j} \quad \text{if} \quad \Lambda^{n,l}_{i,j} > 0 \\ &= \bar{U}^l_{i,j} \quad \text{if} \quad \Lambda^{n,l}_{i,j} = 0, \end{aligned}$$

and $D_x^- U, D_y^- U$ replaced by values which depend on the algorithms being used. For algorithms such as the FCT and the second order Godunov methods, $(D_x^- U)_{i,j}$ is replaced by $(D_x^- U^{n,l})_{i,j}$ if there are values of $U^{n,l}$ defined with which to calculate it; otherwise, $(D_x^- U)_{i,j}$ is replaced

by zero. With algorithms for which such a choice might lead to stability problems, all the values of $(D_{x^-y}U)$ are replaced by zero unless $U^{n,J}$ is defined at enough cells to replace all arguments $D_{x^-y}U$ by $D_{x^-y}U^{n,J}$.

Having calculated these fluxes, we then can calculate the values of $\delta M^l_{i,j}$ analogous to (2.4) in one dimension:

$$\begin{aligned}\delta M^l_{i,j} \;=\; (\Lambda^{n,J} - \Lambda^{n+1,J})\tilde{U}^{n,J}_{i,j} \;+\; A^l_{i-\frac{1}{2},j}F^{x,l}_{i-\frac{1}{2},j} \;-\; A^l_{i+\frac{1}{2},j}F^{x,l}_{i+\frac{1}{2},j} \\ +\; A^l_{i,j-\frac{1}{2}}F^{y,l}_{j-\frac{1}{2}} \;-\; A^l_{i,j+\frac{1}{2}}F^{y,l}_{j+\frac{1}{2}} \;\pm\; A^f_{i,j}F^f_{i,j}\,,\end{aligned} \tag{3.6}$$

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

$$A^l_{i+\frac{1}{2},j} = \frac{1}{2}(A^{2,l}_{i+1,j} + A^{4,l}_{i,j}) \;,\; A^l_{i,j+\frac{1}{2}} = \frac{1}{2}(A^{1,l}_{i,j+1} + A^{3,l}_{i,j}), \tag{3.7}$$

and the $\pm$ is taken to be + if $l=1$, − if $l=2$. A^f and F^f are taken to be nonzero only for those cells which contain nonzero volumes of both regions at either the old or new times. In that case, F^f is the flux across the tracked front in the cell, given by

$$F^f_{i,j} = F^{\hat{n}}(U^{\hat{n}}_{i,j}) - s_{i,j}U^{\hat{n}}_{i,j}, \tag{3.8}$$

where $U^{\hat{n}}$ is the value of the solution to the Riemann problem for the system (3.2) projected in the $\hat{n}_{i,j}$ direction along the ray $\frac{\xi}{\tau} = s_{i,j}$, with left and right states $\tilde{U}^2_{i,j}, \tilde{U}^1_{i,j}$.

The preliminary evolution in time, corresponding to (2.6) in one dimension, is given by

$$U^{n+1,J}_{i,j} = U^{n,J}_{i,j} + \frac{1}{\Delta x\,\Delta y}\delta M^l_{i,j} \tag{3.9}$$

for cells where $\Lambda^{n+1,J}_{i,j} \neq 0$. As was the case in one dimension, (3.8) does not preserve discrete conservation form, which would have been the case if $(\Delta x\,\Delta y)$ had been replaced by $\Lambda^{n+1,J}_{i,j}$. In order to have conservation, we must distribute $\left[1 - \frac{\Lambda^{n,J}_{i,j}}{\Delta x\,\Delta y}\right]\delta M^l_{i,j}$ onto our grid. We do so by decomposing these increments into characteristic variables and distributing them to nearby cells in a volume-weighted fashion. If we expand

$$\left[1-\frac{\Lambda_{i,j}^{n,l}}{\Delta x\,\Delta y}\right]\delta M_{i,j}^{l}=\sum_{k}\alpha_k r_k^{l}$$

where $r_k(U)$, $k=1,\ldots,N$ are the linearized right eigenvectors of the system (3.2) projected in the $\hat{n}_{i,j}$ direction, with $r_k^l = r_k(\overline{U}_{i,j}^l)$, then we can define, analogous to (2.8)

$$\begin{aligned}
\delta M^{+,l} &= \sum_{p>k}\alpha_k r_k^l \\
\delta M^{-,l} &= \sum_{p<k}\alpha_k r_k^l \\
\delta M^{0,l} &= \alpha_p r_p^l \\
\delta M^{tot,1} &= \delta M^{+,1}+\delta M^{+,2}+\delta M^{0,1} \\
\delta M^{tot,2} &= \delta M^{-,1}+\delta M^{-,2}+\delta M^{0,2} \\
\Lambda_{i,j}^{tot,l} &= \sum_{ext(i,j)}\Lambda^{n+1,l}
\end{aligned} \tag{3.10}$$

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

$$U_{i,j}^{n+1,l} := U_{i,j}^{n+1,l}+\sum_{ext(i,j)}\frac{\delta M^{tot,l}}{\Lambda^{tot,l}}. \tag{3.11}$$

§ 4. Gas Dynamics in Two Dimensions

The formulation of the conservative tracking algorithm given in the previous two sections, while quite general, is in two ways less than completely satisfactory. The first way is in the time step control. If one considers the special case of a tracked front coinciding with one of the mesh lines, one immediately sees that the flux at the front is given by that for Godunov's first order method [14]. In general, this method has a more restrictive time step stability limitation than that for any of the higher order predictor-corrector methods mentioned at the beginning of the the previous section. It is not difficult to construct examples of systems of equations for which the above algorithm generates oscillations for time steps violating the CFL time step limit for the first order Godunov method, even though the interior scheme is satisfies a maximum principle for that choice of time step. The second difficulty is that of a lack of an obvious choice of redistribution algorithm for the component in the characteristic decomposition of the flux difference corresponding to the same family of the tracked wave in the case where the wave is genuinely nonlinear. This is a particular problem in two dimensions, where there is a component of the flux difference which is purely an artifact of the impossibility of representing exactly in a numerical calculation the spatial structure of a curved discontinuity. One needs to identify that component of the flux difference, and decide on which side of the tracked front to distribute it. Rather than attempting to give general solutions to these difficulties, we will deal with them in the context of a specific problem, tracking a shock for Euler's equations for compressible flow, and a specific interior algorithm, the second order Godunov method discussed in [6]. For the time step problem, we will intertwine the predictor-corrector algorithm with the tracking algorithm in such a way so that, when the tracked front coincides with one of the mesh lines, the algorithm for the flux is that of a difference method with the same time step restriction as the interior algorithm. We will also identify the component of the flux difference corresponding to the errors in the representation of a curved front,

and distribute it to the high pressure, post-shock region for the reasons discussed in § 2.

We wish to solve the equations of gas dynamics in conservation form in Cartesian geometry, i.e. the system (3.1) with

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad F^x(U) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uE + up \end{bmatrix} \quad F^y(U) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vE + vp \end{bmatrix}. \tag{4.1}$$

Here ρ is the density, $(u,v) = \vec{u}$ the x- and y-components of velocity, and E the total energy per unit mass. The pressure p is given by an equation of state $p = p(\rho, e)$, where $e = E - \frac{1}{2}(u^2+v^2)$ is the internal energy per unit mass. To simplify the exposition, we will assume that the equation of state is that of a polytropic gas, i.e., $p = (\gamma-1)\rho e$, where γ is a constant, $\gamma > 1$. The extension to a general convex equation of state can be carried out using the techniques in [8].

The difference algorithm which we will couple to our front tracking method is a second order extension of Godunov's method. We review this method briefly here; for further details, see [6]. This method is formulated as a predictor-corrector method. The conservation corrector step is of the form (3.3) with $F^x_{i+\frac{1}{2},j} = F^x(U^{n+\frac{1}{2}}_{i+\frac{1}{2},j})$, $F^y_{i,j+\frac{1}{2}} = F^y(U^{n+\frac{1}{2}}_{i,j+\frac{1}{2}})$. The values $U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$, $U^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$ are given as solutions to the Riemann problems for (4.1) projected in the x and y coordinate directions, with left and right states $(U^{n+\frac{1}{2}}_{i+\frac{1}{2},j,L}, U^{n+\frac{1}{2}}_{i+\frac{1}{2},j,R})$, $(U^{n+\frac{1}{2}}_{i,j+\frac{1}{2},L}, U^{n+\frac{1}{2}}_{i,j+\frac{1}{2},R})$.

A repeated construction in these methods, both with and without tracking, is the calculation of upstream-centered extrapolations from the left and right at cell edges; for example, the left and right states for the Riemann problem used to obtain the fluxes at the cell edges. In what follows, we will give the prescription for calculating only such quantities only for the extrapolation from the left at the cell edge $(i+\frac{1}{2},j)$, e.g., $U^{n+\frac{1}{2}}_{i+\frac{1}{2},j,L}$. The extension to the other three cases are straightforward, and are given in [6] for the algorithm without tracking.

The left and right states are given by upstream-centered approximations to the solution of the nonconservative form of (4.1) to obtain time-centered values, which is itself carried out in two steps. First, we calculated the effect of spatial derivative in the direction normal to the cell edge, using the primitive variables $V = (\rho, u, v, p)^t$

$$V^n = V(U^n)$$

$$\hat{V}_{i+\frac{1}{2},j,L} = V^n_{i,j} + \sum_{k=1}^{4} (\lambda^k_{i,j} - \lambda^k_{i,j}) \frac{\Delta t}{\Delta x} (l^k_{i,j} \cdot \Delta V_{i,j}) r^k_{i,j} \tag{4.2}$$

where $\Delta^1 V_{i,j}$ is a difference approximation to $\frac{\partial V}{\partial x} \Delta x$ to which monotonicity constraints have been applied. $\Delta V_{i,j}$ can be written as a function of $(D_x^- V)_{i+r,j}$ $r = -1, \ldots, 3$. Here $\lambda^k_{i,j}, l^k_{i,j}, r^k_{i,j}$ are the eigenvalues and left and right eigenvectors of $\nabla_U V \cdot \nabla_U F^x \cdot \nabla_V U$, evaluated at $U^n_{i,j}$. The second step of the predictor step estimates the effect on the extrapolated states at the cell edge of spatial gradients in the direction parallel to that cell edge. It is given by

$$\hat{U} = U(\hat{V}) \tag{4.3}$$

$$U^{n+\frac{1}{2}}_{i+\frac{1}{2},j,L} = \hat{U}_{i+\frac{1}{2},j,L} - \frac{\Delta t}{2\Delta y} (\hat{F}^y_{i,j+\frac{1}{2}} - \hat{F}^y_{i,j-\frac{1}{2}})$$

where $\hat{F}^y_{i,j+\frac{1}{2}}$ is obtained by solving the Riemann problem for the system (4.1) projected in the y direction, for left and right states $(\hat{U}_{i,j+\frac{1}{2},L}, \hat{U}_{i,j+\frac{1}{2},R})$.

We now describe the extension of the predictor-corrector algorithm given above to the case of a tracked gas-dynamic shock. We assume the choice of normal pointing from 2→1 used in the previous section, and that the shock we are tracking, with that choice of normal, is that associated with the λ^4 characteristic family, so that the component 2 contains the high-pressure, "post-shock" state. We also assume that the local geometry of the front has been calculated as in the previous section. In particular, we know the volumes $\Lambda^{n,l}, \Lambda^{n+1,l}$ and the apertures $A^l_{i+\frac{1}{2},j}, A^l_{i,j+\frac{1}{2}}$, $l = 1,2$ (for the algorithm described here, it turns out that $A^f_{i,j}$ is not required). The extension of the conservative corrector step in the absence of a tracked front is

given by (3.4) where $F^x_{i+\frac{1}{2}j} = F^x(U^{n+\frac{1}{2}j}_{i+\frac{1}{2}j})$, $F^y_{ij+\frac{1}{2}} = F^y(U^{n+\frac{1}{2}j}_{ij+\frac{1}{2}})$ are defined for $A^l_{i+\frac{1}{2}j} > 0$, $A^l_{ij+\frac{1}{2}} > 0$. The values $U^{n+\frac{1}{2}j}_{i+\frac{1}{2}j}, U^{n+\frac{1}{2}j}_{ij+\frac{1}{2}}$ are given as solutions to Riemann problems for (4.1) projected in the x,y directions, with left and right states given by $(U^{n+\frac{1}{2}j}_{i+\frac{1}{2}jL}, U^{n+\frac{1}{2}j}_{i+\frac{1}{2}jL})$, $(U^{n+\frac{1}{2}j}_{ij+\frac{1}{2}L}, U^{n+\frac{1}{2}}_{i+\frac{1}{2}j+\frac{1}{2}R})$. However, F^l_{ij}, the flux at the tracked boundary, is not given by a solution to the Riemann problem directly, but by

$$A^l_{ij}F^l_{ij} = F^{\sharp}(U^{n+\frac{1}{2},1}_{ij}),$$

where $U^{n+\frac{1}{2},1}_{ji}$ is the result of a predictor calculation similar to that given by (4.2),(4.3) with $F^{\sharp}$ given by

$$F^{\sharp}(U) = (A^2_{i+\frac{1}{2}j} - A^2_{i-\frac{1}{2}j})F^x(U) + (A^2_{ij+\frac{1}{2}} - A^2_{ij-\frac{1}{2}})F^y(U) + (\Lambda^{n2}_{ij} - \Lambda^{n+1,2}_{ij})U \tag{4.4}$$

By (3.6) the formula for $A^l_{ij}F^l_{ij}$ differs from that given in (3.8) by $\delta M^1 + O(\Delta x \Delta y \Delta t) = O(\Delta x \Delta y \Delta t)$, so that it is formally first order accurate in smooth regions. Another consequence of (4.4) is that if $U^{n+\frac{1}{2},1}_{ij} = U_0 = U^{n+\frac{1}{2},1}_{i\pm\frac{1}{2}j}, U^{n+\frac{1}{2},1}_{ij\pm\frac{1}{2}}$, then $\delta M^1 = 0$. Thus, by using (4.4) as the flux across the front, we are effectively assigning all of the contributions to the flux difference due to the numerical errors in representing the shock geometry to the post shock region.

The left and right states are given by suitable modifications of the predictor step (4.2),(4.3). We first calculate

$$\hat{V}^l_{i+\frac{1}{2}jL} = V(\bar{U}^l_{ij}) + \sum(\lambda^{kl}_{ij} - \lambda^{kl}_{ij})\frac{\Delta t}{\Delta x}(l^{kl}_{ij}\cdot\bar{\Delta}V^{nl}_{ij}) \tag{4.5}$$

where $\bar{\Delta}V^l_{ij} = \Delta V(DV^{nl}_{i-r,j}, \ldots, DV^{nl}_{i+r,j})$ is expressed as a function of one-sided differences of V^l in the x direction, with those differences which are not defined due to the absence of the appropriate component in a cell set to zero. It is possible because of (3.7) for $A^l_{i+\frac{1}{2}j}$ to be positive while $\Lambda^{nl}_{ij} = \Lambda^{n+1,l}_{ij} = 0$; in that case, $\hat{V}^l_{i+\frac{1}{2}jL}$ is set to be $\hat{V}^l_{i+\frac{1}{2}jR}$, which is defined (otherwise $A^l_{i+\frac{1}{2}j}$ would be zero).

In the second half of the predictor step, we estimate the effect of solution gradients parallel to the cell edges and to the tracked front on the time-centered left and right states at those locations.

$$U^{n+\frac{1}{2},1}_{i+\frac{1}{2},j,L} = \hat{U}^{1}_{i+\frac{1}{2},j,L} - \frac{1}{2\Delta x\,\Delta y}(A^{1}_{j+\frac{1}{2}}(F^{y}(\hat{U}^{1}_{i,j+\frac{1}{2}}) - F^{y}(\bar{U}^{n,1}_{i,j})) - A^{1}_{j-\frac{1}{2}}(F^{y}(\hat{U}^{1}_{i,j-\frac{1}{2}}) - F^{y}(\bar{U}^{n,1}_{i,j}))) \tag{4.6}$$

Here $\hat{U}^{1}_{i,j+\frac{1}{2}}$ is the solution to the Riemann problem with left and right states $(\hat{U}^{1}_{i,j+\frac{1}{2},L}, \hat{U}_{i,j-\frac{1}{2},R})$ for the system projected in the y direction. For the postshock state, there is an additional contribution to the predictor step due to the flux through the tracked wave:

$$U^{n+\frac{1}{2},2}_{i+\frac{1}{2},j,L} := U^{n+\frac{1}{2},2}_{i+\frac{1}{2},j,L} - \frac{n^{x}_{i,j}}{2\Delta x\,\Delta y}(F^{f}(\bar{U}^{n,2}_{i,j}) - F^{f}(\bar{U}^{n,1}_{i,j})). \tag{4.7}$$

Note that, in this part of the predictor step, we subtract from each flux a flux evaluated at the cell center. We do this because we only want contributions due to solution gradients, as opposed to those due to the nonrectangular finite volume differencing near the front; in particular, if the solution consists of a straight shock separated by two constant states, the predictor step yields a zero increment. The particular form is motivated by the extension of the second order Godunov method to general quadrilateral grids discussed in [6], where the predictor step takes a similar form. The additional term in the postshock predictor given by (4.7) has the property that if the tracked front coincides with one of the mesh directions then the predictor is that given by the second order Godunov method in [6], with $F^{f}(U^{n,1}_{i,j})$ acting as the flux through the boundary, and the flux $F^{f}(U^{n,2}_{i,j})$ cancelling the reference flux terms in (4.6). The analogous term in $U^{n+\frac{1}{2},1}_{i+\frac{1}{2},j,L}$ is identically zero, since $F^{f}_{j}(U^{n,1}_{i,j})$ takes the place of $F^{f}(U^{n,2}_{i,j})$ in (4.7).

We must also calculate a predictor step for the tracked front. Since the flux depends only on the state in front of the shock, we need only calculate a predictor step for $U^{n+\frac{1}{2},1}_{i,j}$. It is

given by

$$
\begin{aligned}
U_{ij}^{n+1/2,1} = U_{ij}^{n,1} - \frac{1}{2\Delta x \Delta y} \Big(& - n_{ij}^{y}(A_{i+1/2,j}^{1}(F^{x}(\hat{U}_{i+1/2,j}^{1}) - F^{x}(\bar{U}_{ij}^{n,1}))) \\
& - (A_{i-1/2,j}^{1}(F^{x}(\hat{U}_{i-1/2,j}^{1}) - F^{x}(\bar{U}_{ij}^{n,1}))) \\
& + n_{ij}^{x}(A_{i,j+1/2}^{1}(F^{y}(\hat{U}_{i,j+1/2}^{1}) - F^{y}(\bar{U}_{ij}^{n,1}))) \\
& - (A_{i,j-1/2}^{1}(F^{y}(\hat{U}_{i,j-1/2}^{1}) - F^{y}(\bar{U}_{ij}^{n,1})))\Big).
\end{aligned}
\tag{4.8}
$$

In the case of the tracked wave coinciding with one of the cell edges, this algorithm reduces to the second part (4.3) of the predictor algorithm for the unsplit method without tracking, and the combined algorithm is given by the interior algorithm (4.2) - (4.4) with the slopes adjacent to the front and in the direction of the normal of the front set to zero. Since this corresponds to a hybridization of two schemes having the same time step restrictions, i.e. that of the interior scheme, we expect that the overall calculation will be able to run at the same time step as the interior scheme. Also, the effects of incompatibilities in the representation of the shock geometry are effectively assigned to δM^2. In particular, $\delta M^1 \equiv 0$ if the tracked shock is moving into a constant preshock state.

§ 5. Numerical Results

We use as a test problem the problem of self-similar reflection of a shock by an oblique surface [12]. The initial data consists of a planar shock moving into a uniform fluid with pressure p_0, density ρ_0, and with preshock velocity equal to zero, incident on a reflecting surface at an oblique angle. The surface in back of the incident shock parallel to its direction of propagation is also a reflecting surface. The solution to this problem for positive times consists of some reflected wave propagating into the uniform postshock medium, with a combination of smooth and discontinuous waves contained in region bounded by the incident shock, the reflected wave, and the solid wall boundaries. Normally, one expects the solution to be self-similar, i.e., depending on (x,y,t) only in the combination $\left[\frac{x}{t},\frac{y}{t}\right]$, so that the reflected wave patterns at different times are identical, modulo a linear rescaling of the coordinates. In that case, the reflection patterns obtained are functions of the incident shock Mach number M_S, the ramp angle α, and the equation of state γ.

We use the hybrid tracking and capturing algorithm described in the previous section to calculate solutions to this problem. We track the incident shock, while capturing all the waves found behind it. In particular, we compute Mach triple points adjacent to the undisturbed medium in front of the incident shock as kinks forming in the incident shock. This is a particularly appropriate use of the hybrid algorithm, since the combination of the incident shock and Mach stem attached to it are the strongest shocks in the problem, with pressure ratios of arbitrary magnitude. In contrast, the reflected shocks are typically in the weak shock regime, with pressure ratios ≤ 10. In addition, our interior finite difference grid is oriented so that one of the grid directions is parallel to the surface oblique to the shock; consequently, the other reflecting surface intersects the grid obliquely, and is treated as a tracked impermeable surface, but one along which the fluid is allowed to slip, using a suitable version of the redistribution algorithm.

In these calculations, the volumes representing the tracked front are obtained by propagating a polygonal curve through the mesh. The vertices (x_k^n, y_k^n) of this curve are located such that at least one coordinate is located on a grid line: $x_k^n = (i-\frac{1}{2})\Delta x$ or $y_k^n = (j-\frac{1}{2})\Delta y$; furthermore, any point on the curve which intersects a grid line is a vertex. Finally, we assume that $y_{k-1}^n < y_k^n$, an assumption which is reasonable for our shock reflection problem, and simplifies the logic of the calculation considerably. We use this representation of our tracked shock front to calculate the partial volumes Λ_{ij}^n with the region to the left of the curve taken to be in the post shock region, the calculation of which, given such a representation, is straightforward. Thus we must give an algorithm for updating the array (x_k^n, y_k^n), to obtain a representation of the front at time t^{n+1}. Once that is done, then the volumes Λ_{ij}^{n+1} can be computed, and the solution on either side of the front updated. The evolution of the array proceeds in the following four steps:

1) Delete all points (x_k^n, y_k^n) which lie on x coordinate lines, i.e. for which x_k is of the form $(i-\frac{1}{2})\Delta x$.

2) Move all the remaining points $(x_k, y_k) = (x_k^n + s_x \Delta t, y_k^n + s_y \Delta t)$. Here the shock speed $(s_x, s_y) = s\,\hat{n}$ is calculated using the jump relations on either side in the following fashion. Since we have deleted the points located on x grid lines, and since the y coordinates are strictly monotone functions of k, the cells $(i, j-1)$, (i, j) above and below (x_k^n, y_k^n) have nonzero postshock volumes $\Lambda^{n,2}$. For each of these cells we compute $\bar{U}^{n,2}$ using (3.5) from $\bar{U}^{n,2}$ compute values for p and $\vec{u}$, and an estimate for the shock speed (s_x, s_y):

$$(s_x, s_y) = s\,\frac{\vec{u}}{|\vec{u}|}\,,\quad s = \left[\frac{\gamma p_0}{\rho_0}\left(1 + \frac{\gamma+1}{2\gamma}\,\frac{(p-p_0)}{p_0}\right)\right]^{1/2}. \tag{5.1}$$

To obtain a speed for (x_k^n, y_k^n), we we take a volume weighted average of the two speeds above and below:

$$(s_x, s_y)_k = f_{i,j}(s_x, s_y)_{i,j} + f_{i,j-1}(s_x, s_y)_{i,j-1}$$

$$f_{i,j-1;\,i,j} = \frac{\Lambda^{n}_{i,j-1;\,i,j}}{\Lambda^{n}_{i,j-1} + \Lambda^{n}_{i,j}}$$

3) Delete all points (x_k, y_k) which are contained in the same cell as (x_{k-1}, y_{k-1}).

4) Create the new set of points (x_k^{n+1}, y_k^{n+1}). consisting of the points where the line segments connecting successive points of (x_k, y_k) intersect the coordinate lines $\{(x,y)\colon x = (i - \tfrac{1}{2})\Delta x \;\; or \;\; y = (j - \tfrac{1}{2})\Delta y\}$.

It is assumed that the arrays obtained in the deletion steps 1) and 3) have the same ordering as the array input to them. If that is the case, and if the time step satisfies the constraints $|s_y \Delta t|$, $|((s_y)^n_{k-1} - (s_y)^n_k)\Delta t| < \Delta y$, then the array of points produced at the end of step 3) has the properties that $y_k > y_{k-1}$, and that no more than one point (x_k, y_k) is in each cell. From this it follows that y_k^{n+1} is a strictly monotone function of k. Having calculated $\Lambda^{n,J}$, $\Lambda^{n+1,J}$, we then use the procedure describing in § 3 and the Appendix to calculate the local geometry of the front in space-time, using (5.1) to calculate the speed and unit normal.

Our first test case is that of a $M_S = 10$, $\alpha = 30°$, $\gamma = 1.4$ (figure 1). This problem was used in [33] as a test problem to compare a number of methods for gas dynamics. In these plots, and those that follow, we plot 30 equally spaced contours of the various dependent variables, with the values of the the dependent variables for points under the ramp and in front of the tracked shock set equal to the values in the uniform state behind the incident wave. In addition, we plot the polygonal curve defined by the (x_k^n, y_k^n). In order to compare the tracking method with the results obtained by shock capturing, we include a fully capturing calculation of the same problem using the second order Godunov method presented in [6] (figure 2). Overall, the results look quite similar; the only major difference is that the error generated where the shock exits the top boundary in the capturing calculation is, of course, absent from the tracking calculation. Consequently, the weak contact surface emanating from the second triple point is

more easily seen in the tracking calculation as a slight kink in the density contours, and a single contour in the entropy.

The second test problem is that of $M_S = 8$, $\alpha = 35°$, $\gamma = 1.2$ (figure 3), taken from [1],[9]. This problem tests the ability of the tracking algorithm to represent strongly curved shocks. Because of the low γ, the jet formed in the double Mach configuration is accelerated sufficiently strongly to push the original Mach stem out in front of it, causing a portion of the tracked shock to be strongly curved, and another Mach triple point to form at the end of the curved segment. The combined tracking and capturing algorithm has no difficulty representing the complicated combination of smooth and discontinuous flow generated by the curved Mach stem. We call particular attention to the combination of smooth and discontinuous entropy variation along the Mach stem.

The final test problem is the case $M_S = 2.05$, $\alpha = 60°$, $\gamma = 5/3$ (figure 4). These conditions result in a regular reflection pattern. This is a problem for which there is no particular advantage in tracking the incident shock, since the it interacts with the nonuniform part of the flow field at only one point, the reflection point, where the amplitude of the wave interacting with the tracked shock is large. Nonetheless, this algorithm yields quite acceptable results, with the exception of a four cell wide error in the entropy at the wall. This error is analogous to wall heating in one dimension [22],[32], and comes from capturing the reflected shock. If the reflected shock were also tracked, as in [3], for example, then this error would not occur; as is, it is smaller than is observed in fully captured calculations of this problem [12]. Also of interest in this case is the fact that the reflected shock is detached, leading to nonuniform flow along the left ramp boundary. The tracking algorithm is able to resolve the tangential flowfield along that boundary, without evidence of instability, even at the reflected shock.

§ 6. Discussion and Conclusions

We have presented here a hybrid method for computing multidimensional solutions to hyperbolic conservation laws, in which we track a single distinguished discontinuity, while capturing all others, even if they intersect the tracked front. The essential new idea here is that of redistributing the flux differences (2.4),(3.6) in a way that is stable and preserves global conservation. This technique is quite general: it can be applied to fronts represented as polygonal lines, or by a volume-of-fluid formulation, as long as the method for updating the volumes is exact for flat fronts. Thus we are free to choose a representation of the geometry of the front appropriate to the stability properties of the front being calculated. We intend to apply this technique to a number of multidimensional problems, including the formation of Mach stems in detonation fronts [24],[26], the interaction of flame fronts and compressibility effects, and the dynamics of unstable slip surfaces and material interfaces. In all these cases, there is a distinguished front — a reaction front, contact or slip surface — whose dynamics would be incorrectly calculated if captured on a uniform finite difference grid whose mesh spacing resolves only the smooth hydrodynamic length scales. Of course, it will be necessary to solve the one dimensional Riemann problem normal to the front, which is still an open research question in the case of a general reacting fluid; however, solutions are known in the case of some simplified models [4],[29].

Even for discontinuities which are adequately represented on uniform grids by capturing methods, there are situations where tracking is a useful technique. For example, one often wishes to compute shocks on meshes which change discontinuously, such as in local adaptive mesh refinement [2]. For gas-dynamic shocks which intersect the mesh discontinuity, one finds unacceptably large errors generated at the mesh discontinuity if the shock is very strong, while those errors are quite small for weak shocks. Thus for problems in which there is only

one strong shock, such as the incident shock in the ramp reflection problem in § 5, or the bow shock in a supersonic steady flow, it is possible, by tracking that distinguished shock, to apply adaptive mesh refinement only in the regions where it is required to resolve accurately the smooth or weakly shocked flow, allowing the tracked wave to intersect the mesh discontinuity. This combination of refinement and tracking has proved useful in investigating the details of the transition between regular and Mach reflection in [1].

One question which we have not addressed is the use of a conservative tracking method such as this with an operator split interior method. We believe that this is going to require a different approach than the one used here. The basis for this assertion is most easily seen in the following example. Consider a stationary tracked slip line for the equations of gas dynamics (4.1), i.e. a line across which the pressure is constant, and the velocity on either side is parallel to the front. If one naively applies an operator split method to this problem, i.e. by incrementing sequentially the solution to the left of the tracked front, due to the fluxes in each of the coordinate directions, pressure variations of $O(1)$ amplitude are generated, since the velocity jump in each of the coordinate directions is compressing or expanding the fluid. Only by differencing all the fluxes simultaneously does one see that the net compression is zero. Of course, one could deal with this problem in a variety of ways, such as hybridizing the operator split method with an unsplit method in the neighborhood of the tracked wave, or not evaluating the points of the increments of the solution corresponding to a tracked wave separating constants until all of them are available, similar to the modification of the predictor step for the second order Godunov method (4.6).

Another area where further work is required is that of the numerical analysis of this method. In more than one dimension, this is likely to be a quite substantial undertaking, since the mathematical theory for the for the well-posedness of solutions to the initial / boundary

problem being approximated by this method is quite difficult [18]. In addition, the formal analysis given by (2.10) does not even yield pointwise error estimates in the neighborhood of the tracked wave. This latter problem is easily remedied by the following modification to the algorithm. If one replaces (2.6) by

$$\delta M_S := \delta M_S + \Lambda_S^{n+1}(U_S^n - \tilde{U}_S)$$

$$U_S^{n+1} = \tilde{U}_S + \frac{1}{\Delta x}\,\delta M_S$$

where δM_S on the right of of the first assignment is given by (2.4), and $\tilde{U}_S$ is some stable non-conservative estimate to U_S^{n+1} having a local truncation error of $O(\Delta x \Delta t)$, then one obtains a method which conserves and is first order accurate pointwise, even in the neighborhood of the tracked wave. This technique can also be used to accelerate convergence to steady state near the tracked wave, e.g. by setting $\tilde{U}_S = \bar{U}_S$.

An open question concerning this technique is its extension to the case where there are two or more discontinuities to be tracked. Currently, our algorithm applies only to the case where a single front which separates the computational domain into two components; in general, neither component need be of fixed connectivity as a function of time. So our technique can be applied to the collision of two strong shocks, or to a combustion front with multiple ignition centers. However, the technique as currently formulated cannot be applied to the calculation of two or more tracked fronts of different characteristic families, such as a tracked front interacting with a tracked slip surface. One idea which has promise is to derive a modification of the flux redistribution algorithm which simply redistributes fluxes in a volume-weighted way inside a cell where the fronts intersect, thus reverting to a redistribution algorithm similar to that used in multifluid calculations in [7],[11],[33]. The redistribution algorithm could also be used to obtain a conservative coupling for methods where the intersection points are explicitly tracked, as in [3].

Acknowledgements

This research was supported by the US Department of Energy, Office of Energy Research Applied Mathematical Sciences Program at the Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48, and at the Lawrence Berkeley Laboratory under contract no. DE-AC03-76SF0098; and by the Air Force Office of Scientific Research under contract no. AFOSR-ISSA-870016. Parts of this work were done while the first author was a postdoctoral research associate at the Mathematical Sciences Research Institute, and while the second author was a visiting staff member at the University of Minnesota Mathematics Department, and at the Courant Institute for the Mathematical Sciences.

Appendix

We give here the construction of the apertures $A^{p,l}$ discussed in § 3. Since this is a cell-centered calculation, we omit subscripts (i,j) and assume without loss of generality that the lower left corner of our cell is located at the origin of our coordinates. As input data, we are given $\Lambda^{n,1},\Lambda^{n+1,1}$ with $\Lambda^{n,1}+\Lambda^{n,2}$, $\Lambda^{n+1,1}+\Lambda^{n+1,2}=\Delta x\Delta y$, as well as $\hat{n}$, the unit normal pointing from region r into region l. By rotations, reflections, and reversal of the time direction, we can transform the problem into one for which $\Lambda^{n+1,2}\geq\Lambda^{n,2}$, and $\hat{n}=(\cos\theta,\sin\theta)$, $0\leq\theta<\frac{\pi}{4}$. We also have at least one of $\Lambda^{n,2}>0$ $\Lambda^{n+1,2}<1$ holding.

We will first consider the case where $0<\Lambda^{n,2}\leq\Lambda^{n+1,2}<1$. In this case, the part of the cell containing region 2 is below and to the left of the front, and increases in area as the front moves up and to the right. If we define $\Lambda_1=\tan\theta\frac{\Delta y^2}{2}$, $\Lambda_2=\Delta x\Delta y-\Lambda_1$, then region 2 occupies a triangle in the lower left corner if $\Lambda^2(t)\leq\Lambda_1$, a trapezoid if $\Lambda_1<\Lambda^2(t)<\Lambda_2$, and the complement of a triangle in the upper right corner if $\Lambda^2(t)\geq\Lambda_2$ (figure A1). Given this geometric information, it is easy to specify $x(\Lambda)$, the location of the point where the tracked wave intersects the x-axis as a function of Λ, the area of the cell in region 2, to be

$$\begin{aligned} x(\Lambda) &= (2\Lambda\tan\theta)^{1/2} && \text{if } \Lambda\leq\Lambda_1 \\ &= \frac{\Lambda}{\Delta y}+\frac{\Delta y}{2}\tan\theta && \text{if } \Lambda_2\geq\Lambda\geq\Lambda_1 \\ &= \Delta x+\Delta y\tan\theta-(2\tan\theta(\Delta x\Delta y-\Lambda))^{1/2} && \text{if } \Lambda>\Lambda_2. \end{aligned}$$

From this we can calculate the shock speed s

$$s=\frac{x(\Lambda^{n+1,2})-x(\Lambda^{n,2})}{\Delta t\cos\theta},$$

and the times t_1,t_2 when $\Lambda^2(t)=\Lambda_1,\Lambda_2$:

$$t_1=\frac{x(\Lambda_1)-x(\Lambda^{n,2})}{s\cos\theta}+t^n$$

$$t_2 = \frac{x(\Lambda_2) - x(\Lambda^{n,2})}{s\cos\theta} + t^n .$$

Finally, we can calculate $A^{1,2}A^{2,2}$ to be

$$A^{1,2} = \frac{1}{(t^{n+1}-t^n)} \int_{t^n}^{t^{n+1}} A(t)dt$$

$$A^{2,2} = \frac{1}{(t^{n+1}-t^n)} \int_{t^n}^{t^{n+1}} A^{2,2}(t)dt$$

where

$$\begin{aligned} A^{1,2}(t) &= x(\Lambda^n) + (t-t^n)s\cos\theta \quad \text{if } t^n < t < t_2 \\ &= \Delta x \qquad\qquad\qquad\qquad \text{if } t > t_2 \end{aligned}$$

$$\begin{aligned} A^{2,2}(t) &= \frac{x(\Lambda^n)}{\tan\theta} + (t-t^n)s\sin\theta \quad \text{if } t^n < t < t_1 \\ &= \Delta y \qquad\qquad\qquad\qquad \text{if } t > t_1 \end{aligned}$$

$A^{1,1}$ and $A^{2,1}$ are computed using $A^{p,1} + A^{p,2} = \Delta y \Delta t$ for p even, $\Delta x \Delta t$ for p odd; $A^{3,1}, A^{4,1}$ are calculated by a similar procedure, or the algorithm identical to that given above for a suitably time-reversed and rotated problem.

In the case where $\Lambda^{n,2}=0$, or $\Lambda^{n+1,2}=1$ the construction is modified slightly, since we no longer have sufficient information to uniquely determine s. We consider only the case $\Lambda^{n,2}=0$, the other case being obtainable from the first by rotation and time reversal. We calculate t_B, the time when the front first enters the cell, to be

$$t_B = t^{n+1} - \min\left[\frac{x(\Lambda^{n+1,2})}{\bar{s}}, \Delta t\right]$$

so that $t_B \geq t^n$. Then $x(\Lambda)$, s, t_1, t_2, $A^{p,2}(t)$, $p = 1,2$ are defined as before, but replacing t^n by t_B. Finally we define

$$A^{p,2} = \frac{1}{(t^{n+1} - t^n)} \int_{t^n}^{t^{n+1}} A^{p,2}(t)dt \qquad p = 1,2$$

$A^{1,1} = \Delta x \Delta t - A^{1,2}, \quad A^{2,1} = \Delta y \Delta t - A^{2,2}.$

Having computed all of the apertures on the cell edges, the aperture at the front is given by

$$A^f = A^{1,2} + A^{2,2} - A^{3,2} - A^{4,2} + \Lambda^{n,2} - \Lambda^{n+1,2}.$$

This is most easily seen by applying the divergence theorem to

$$0 = \int_{\Delta x [t^n, t^{n+1}] \cap S_2} (\nabla \cdot 1)\, dxdydt.$$

References

1 M. Berger, I.-L. Chern, P. Colella, and H.M. Glaz, "Transitions and Bifurcations in Self-Similar Shock Reflections", in preparation.

2 M. Berger and P. Colella,"Adaptive Mesh Refinement for Shock Hydrodynamics", pre-print.

3. I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, J. Comput. Phys. **62** (1986), 83.

4. A.J. Chorin, J. Comput. Phys. **25** (1977), 253.

5. A.J. Chorin, J. Comput. Phys. **35** (1980), 1.

6. P. Colella, Multidimensional "Upwind Methods for Hyperbolic Conservation Laws", Lawrence Berkeley Laboratory report LBL-17023, to appear in J. Comput. Phys.

7. P. Colella, R. Ferguson, H.M. Glaz, "A Single Step Eulerian Algorithm for Multimaterial Compressible Flow Calculations", in preparation.

8. P. Colella and H. M. Glaz, J. Comput. Phys. **59** (1985), 264.

9. P. Colella and H.M. Glaz, "Numerical Calculation of Complex Shock Reflections in Gases", Springer Lecture Notes in Physics **218** , 154.

10. P. Colella, A. Majda, and V. Roytburd, SIAM J. Sci. Stat. Computing 7 (1986), 1059.

11. R. DeBar, "Fundamentals of the KRAKEN Code", Lawrence Livermore Laboratory Report UCIR-760, 1974.

12. H.M. Glaz, P. Colella, I.I. Glass, and R. Deschambeault, Proc. Royal Soc. London A **398** (1985), 117.

13. J. Glimm, B. Lindquist, O. McBryan, B. Plohr, and S. Yaniv, SPE J. (1983) 41.

14. S.K. Godunov, A.W. Zabrodyn and G.P. Prokopov, USSR Comp. Math. and Math. Phys. **1** (1961), 1187.

15. J.M. Hyman, Physica **D, 12D** (1985), 396.

16. C.W. Hirt, and B.D. Nichols, J. Comput. Phys. **39** (1981), 201.

17. P. Kutler, S.R. Chakravarthy, and C. Lombard, "Supersonic Flow Over Ablated Nosetips Using an Unsteady Implicit Numerical Procedure", AIAA paper 78-213, AIAA 16th Aerospace Sciences Meeting, Huntsville, Alabama, January 16-18, 1978.

18. A. Majda, "Compressible Fluid Flow and Systems of Conservation Laws in Several Space Variables", Springer Applied Mathematical Sciences **53** , 1984.

19. Methods in Computational Physics **3** , Academic Press, New York, B. Alder and S. Fernbach, eds.

20. G. Moretti, "Thoughts and Afterthoughts About Shock Computations", Report No. PIBAL-72-37, Polytechnic Institute of Brooklyn, 1972.

21. W.F. Noh, ref. 19, 117.

22. W. F. Noh, "Errors for Calculating Hydrodynamic Shocks Using an Artificial Viscosity and Artificial Heat Flux", Lawrence Livermore National Laboratory report UCRL-53669, December, 1985.

23. W.F. Noh and P. R. Woodward, Springer Lecture Notes in Physics **59** (1976), 330.

24. A.K. Oppenheim, "Gasdynamics of Explosions", CIME Lecture Notes, 1970.

25. R.D. Richtmyer and K.W. Morton, "Difference Methods for Initial Value Problems" Interscience, New York, 1967.

26 R. Rosales and A. Majda, SIAM J. Appl. Math. **43** (1983), 1086.

27. M. Salas, AIAA J. **14** (1976), 583.

28. T.D. Taylor and B.S. Masson, J. Comput. Phys. **5** (1970), 443.

29. Z.-H. Teng, A.J. Chorin, and T.-P. Liu, SIAM J. Appl. Math. **42** (1982), 964.

30. B. van Leer, "Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws", in Computing Methods in Applied Sciences and Engineering VI, R. Glowinski and J.-L. Lions eds., North Holland, (1984), 493.

31. J. von Neumann and R.D. Richtmyer, J. Appl. Phys. **21** (1950), 232.

32. P. R. Woodward, "Piecewise Parabolic Methods for Astrophysical Fluid Dynamics", in Astrophysical Radiation Hydrodynmics, K. - H. Winkler and M. Norman, eds., D. Reidel, 1986, 245.

33. P. R. Woodward and P. Colella, J. Comput. Phys. **54** (1984), 115.

34. S.T. Zalesak, J. Comput. Phys. **31** (1979), 335.

Figure Captions

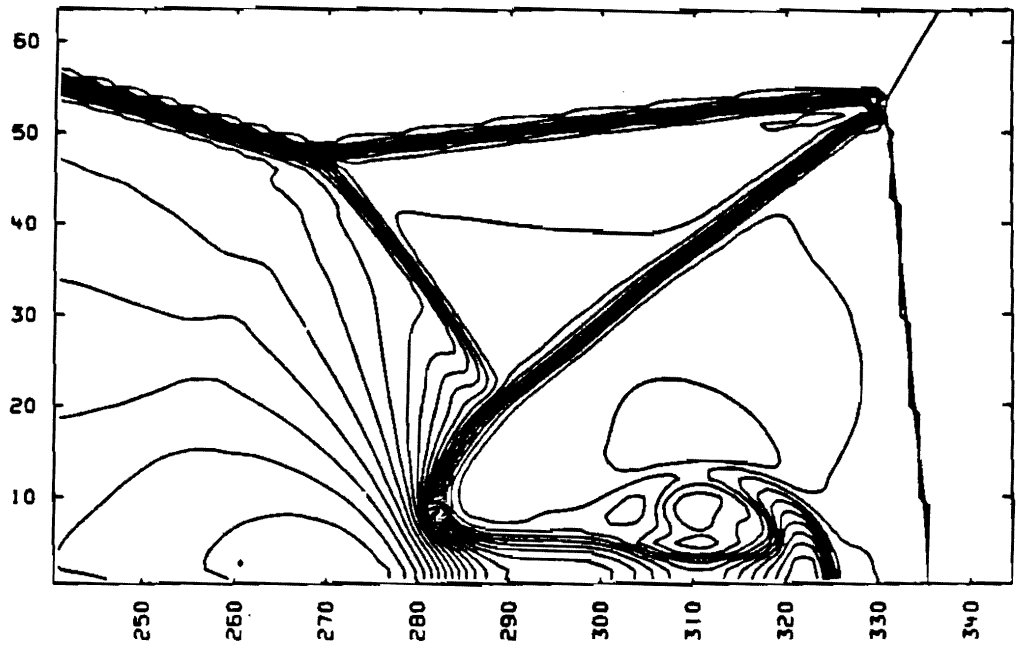
Figure 1. Contour plots for $M_s = 10$, $\alpha = 30$, $\gamma = 1.4$ ramp reflection problem computed using the hybrid track and capturing algorithm. Figure 1 (a) - density, figure 1 (b) - pressure, figure 1 (c) - entropy.

Figure 2. Density contour plots for $M_s = 10$, $\alpha = 30$, $\gamma = 1.4$ ramp reflection problem computed using the second order Godunov method, with all discontinuities captured.

Figure 3. Contour plots for $M_s = 8$, $\alpha = 35$, $\gamma = 1.2$ ramp reflection problem computed using the hybrid track and capturing algorithm. Figure 3 (a) - density, figure 3 (b) - pressure, figure 3 (c) - entropy.

Figure 4. Contour plots for $M_s = 2.05$, $\alpha = 60$, $\gamma = \frac{5}{3}$ ramp reflection problem computed using the hybrid track and capturing algorithm. Figure 4 (a) - density, figure 4 (b) - pressure, figure 4 (c) - entropy.

Figure A1. Examples of local configurations of the tracked front in a finite difference cell.


60
50
40
30
20
10
250
260
270
280
290
300
310
320
330
340

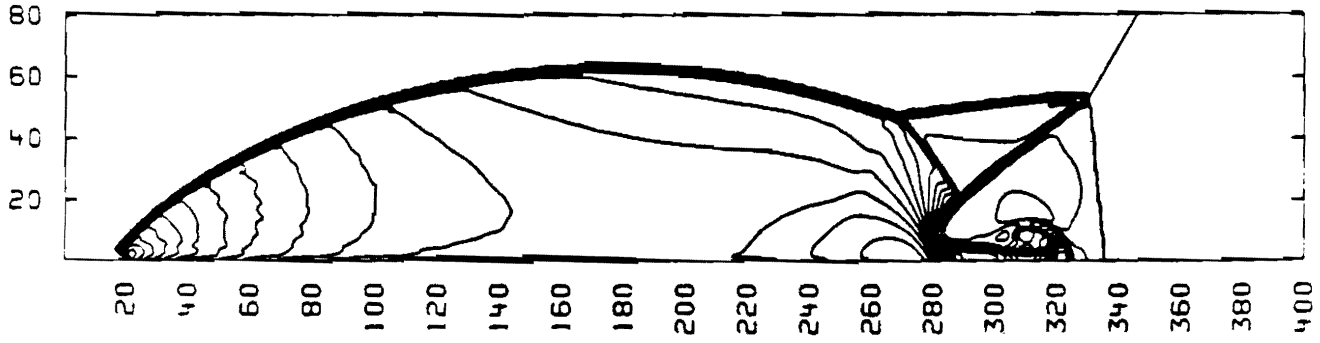
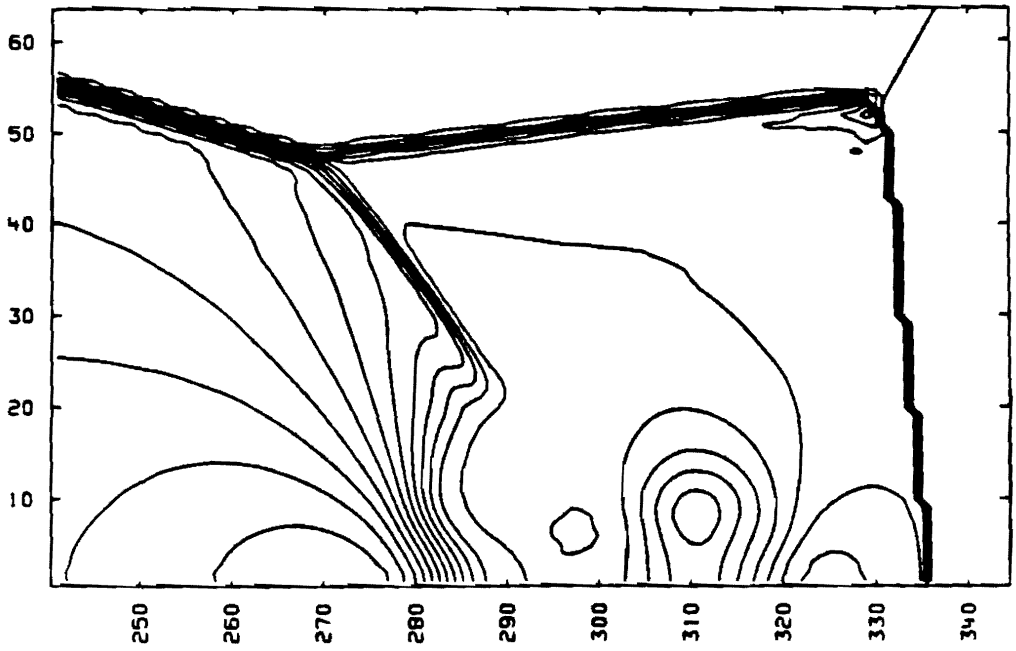

80
60
40
20
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400

Figure 1 (a)


60
50
40
30
20
10
250
260
270
280
290
300
310
320
330
340

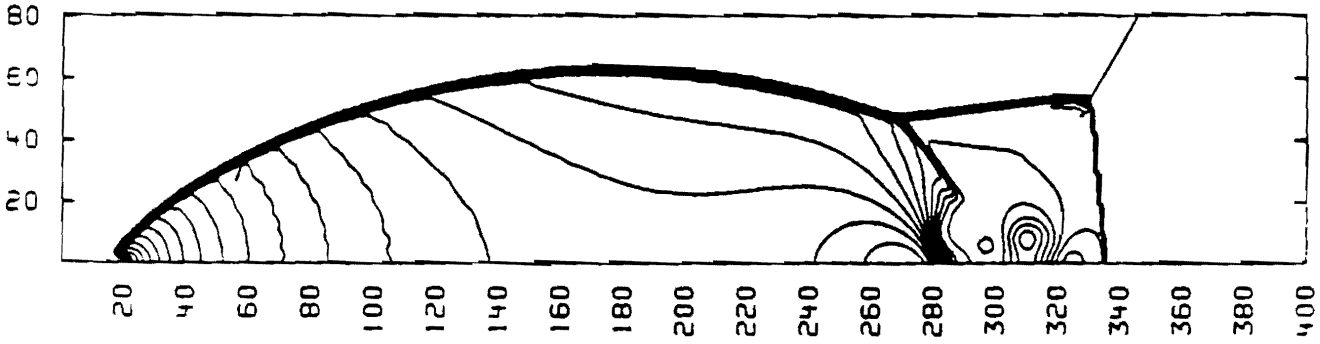
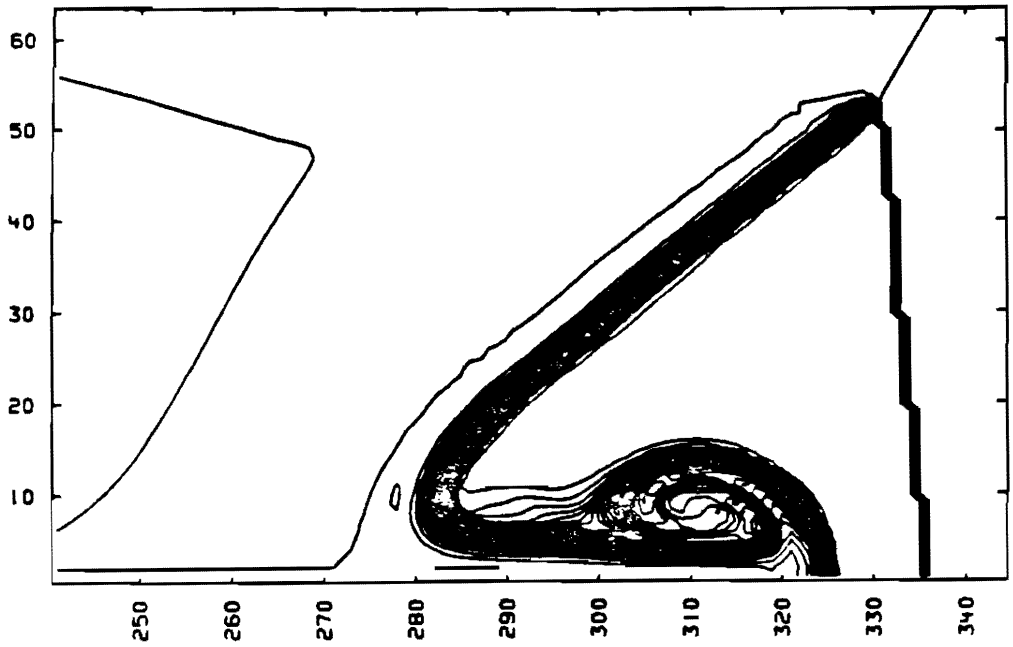

80
60
40
20
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400

Figure 1 (b)


60
50
40
30
20
10
250
260
270
280
290
300
310
320
330
340

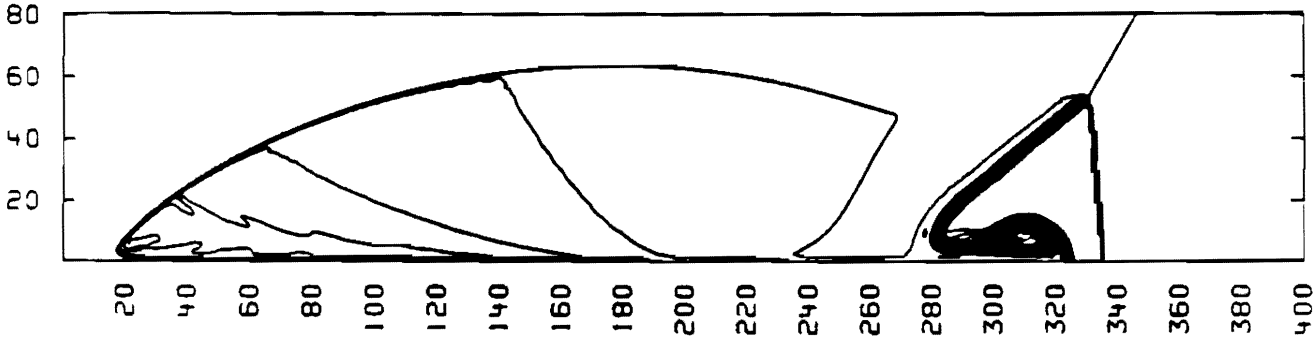
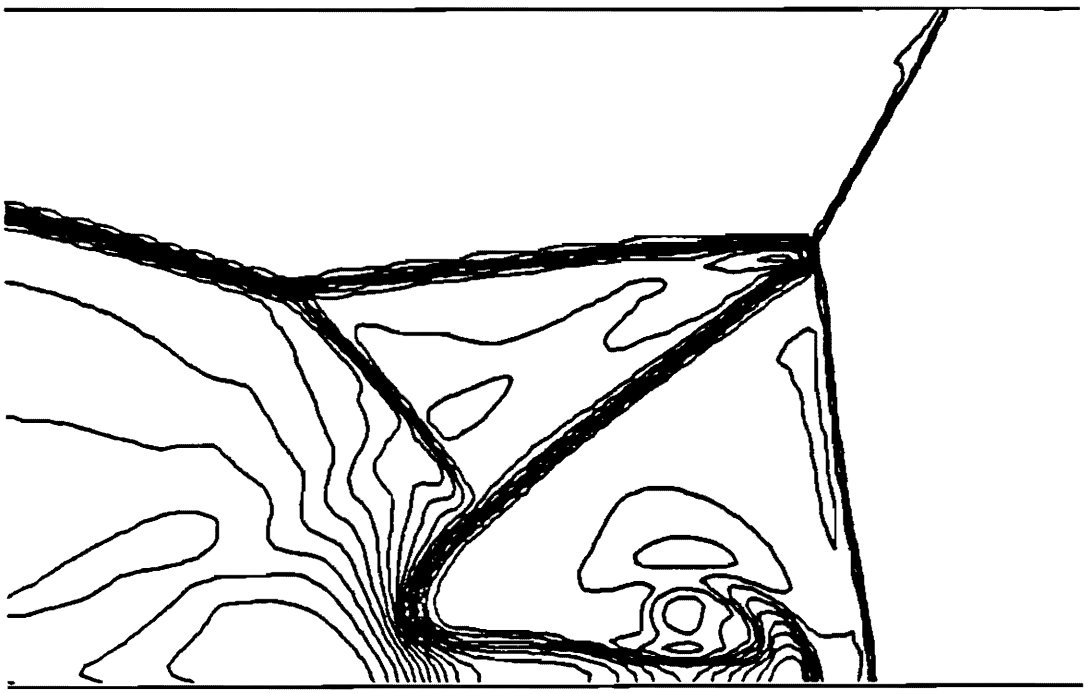

80
60
40
20
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400

Figure 1 (c)



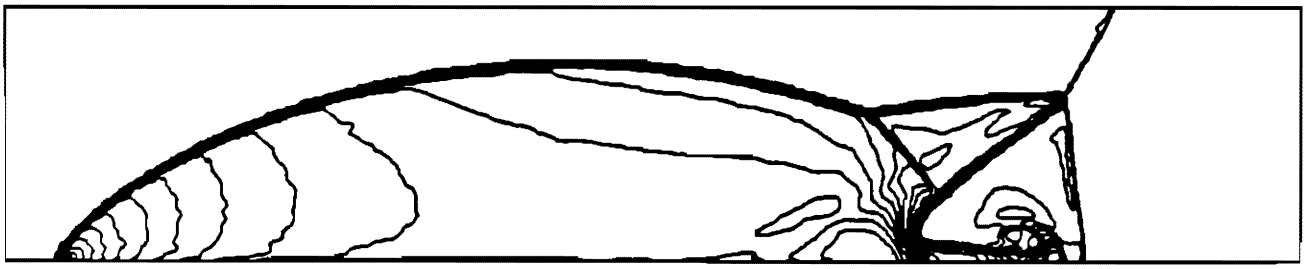


Figure 2

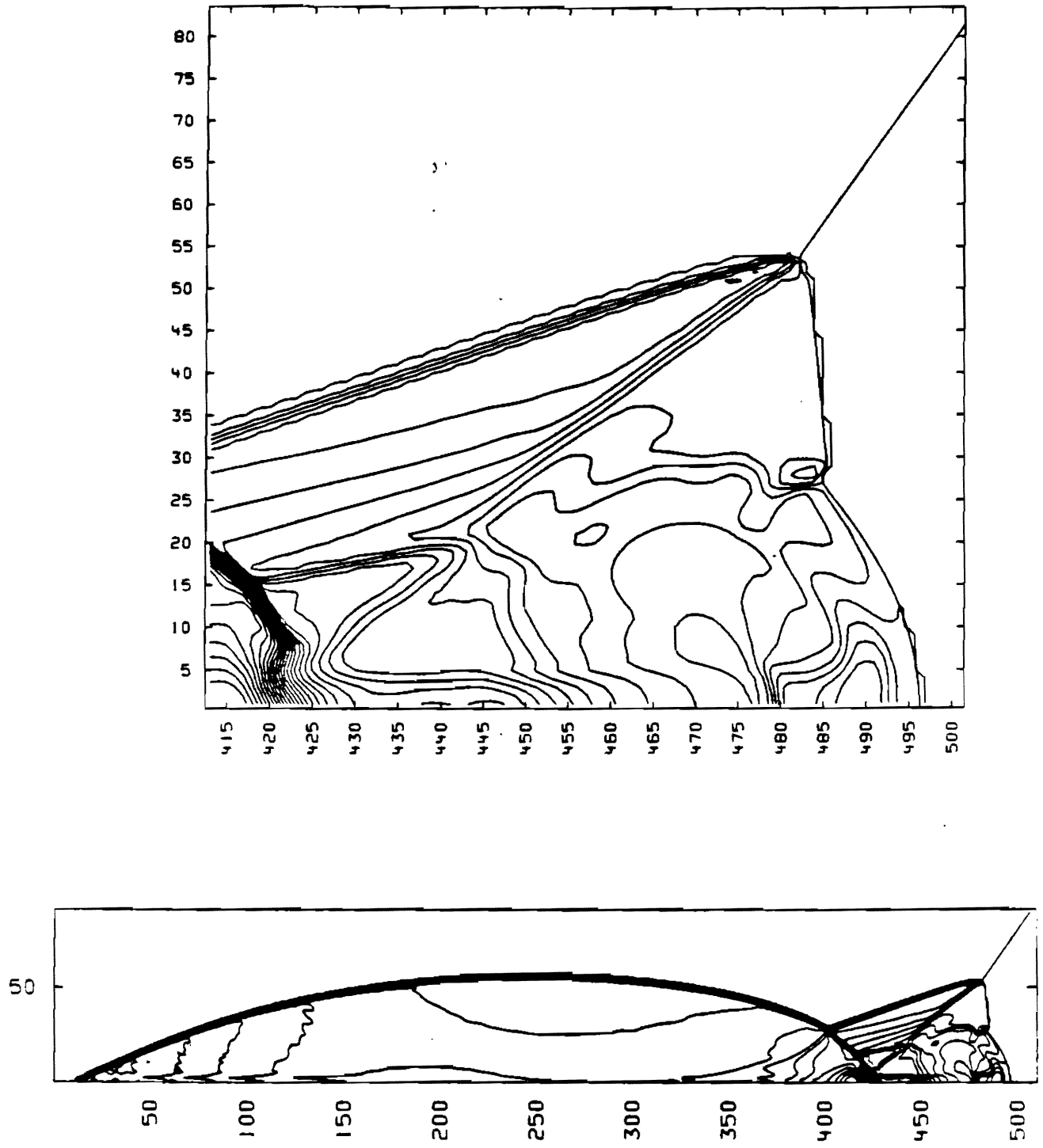

80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
50
50
100
150
200
250
300
350
400
450
500

Figure 3 (a)

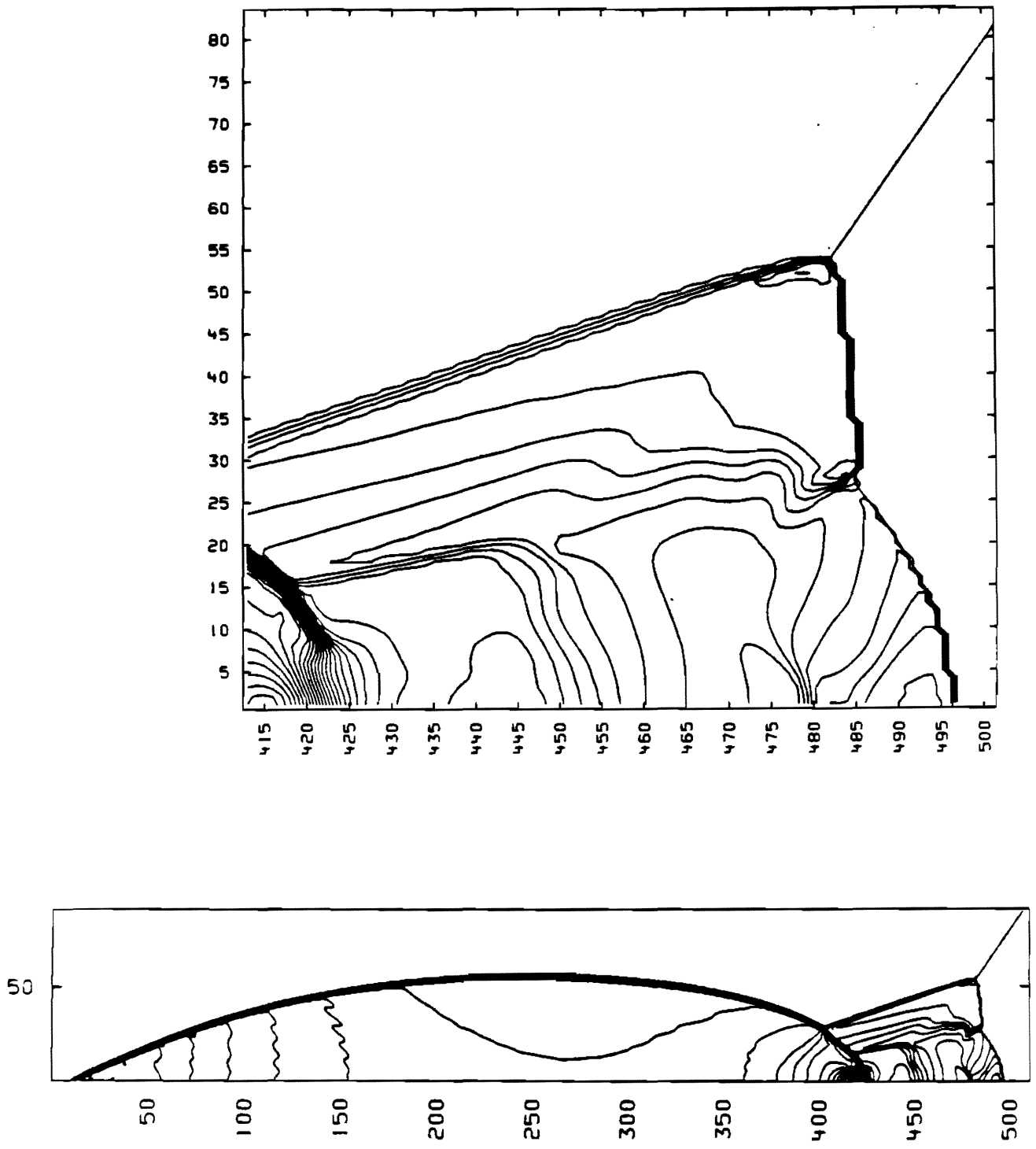

80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
50
50
100
150
200
250
300
350
400
450
500

Figure 3 (b)

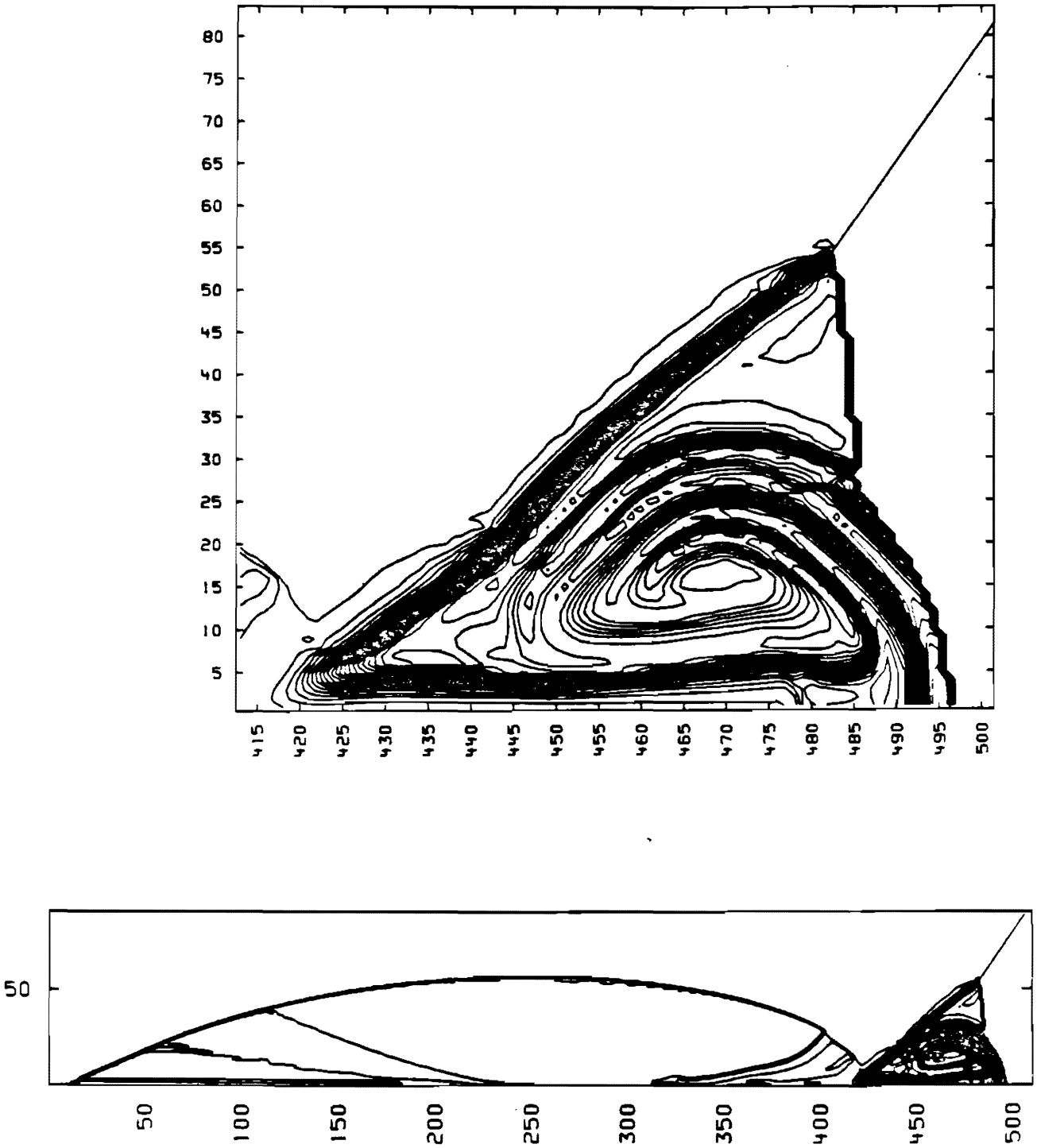
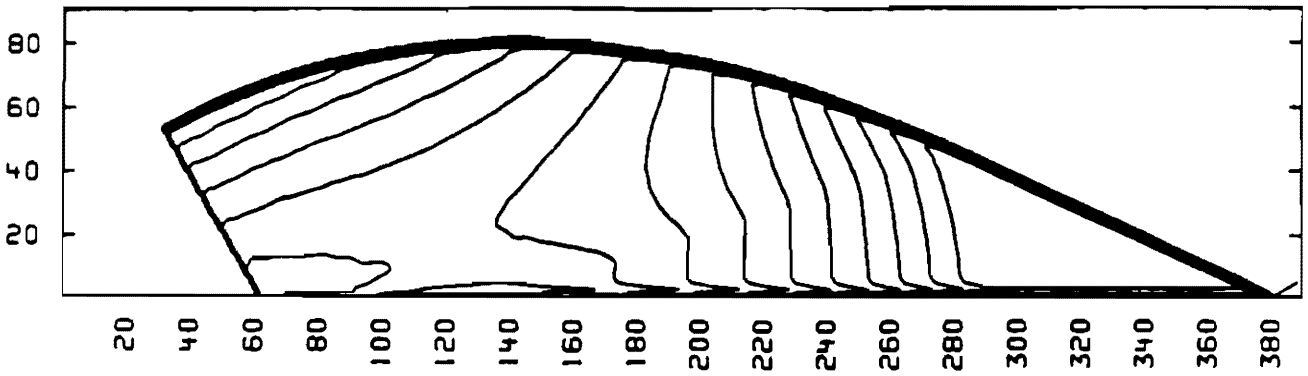
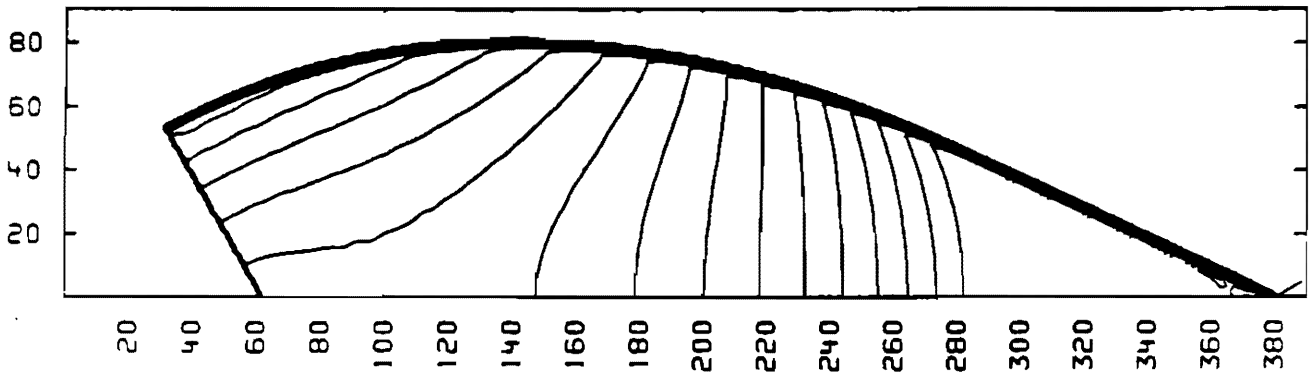

80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
50
50
100
150
200
250
300
350
400
450
500

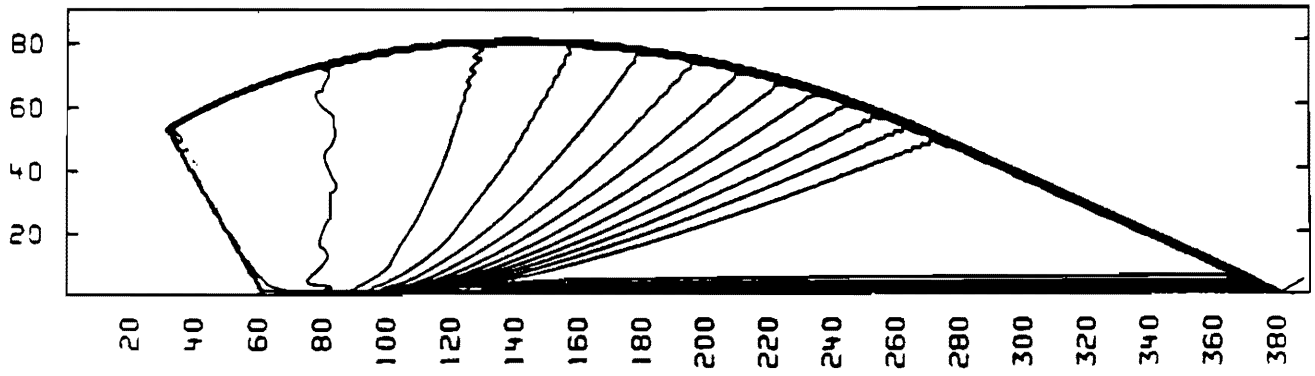
Figure 3 (c)


80
60
40
20
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380

(a)


80
60
40
20
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380

(b)


80
60
40
20
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380

(c)

Figure 4