UCRL- 100880
PREPRINT

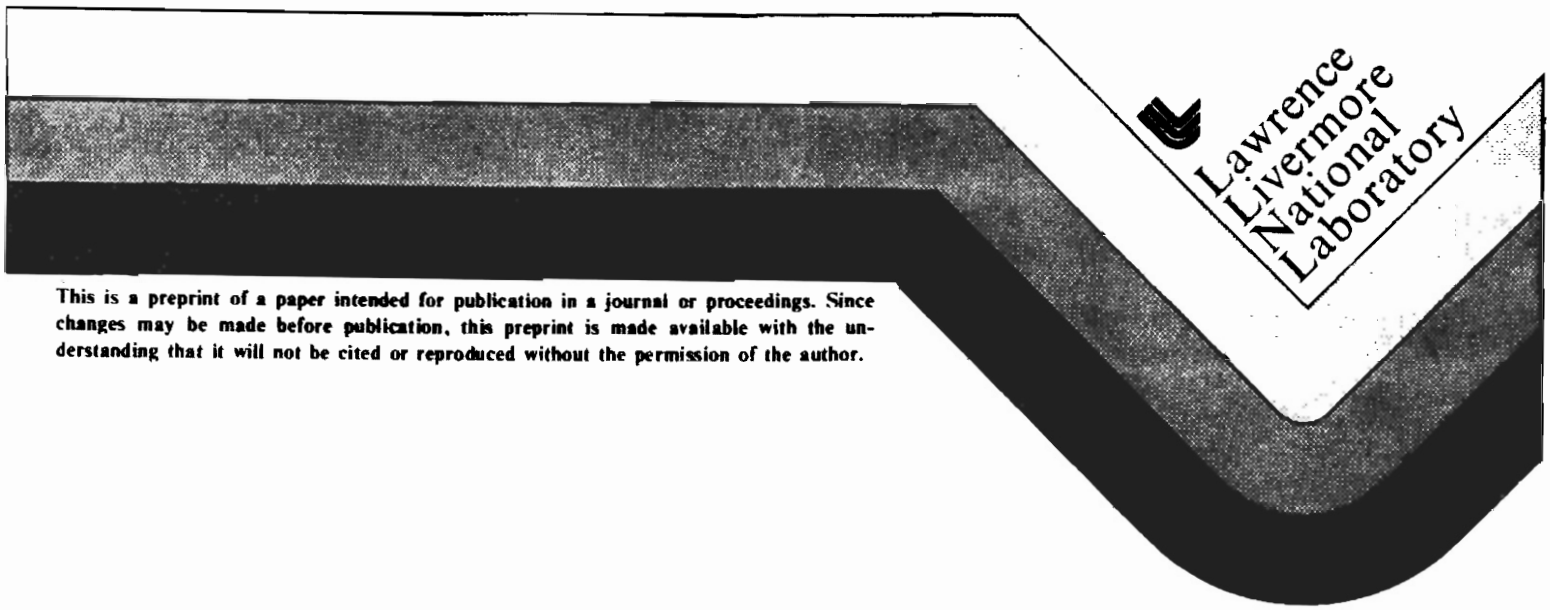
Adaptive Mesh Refinement on Moving Quadrilateral Grids

John B. Bell
Phillip Colella
John A. Trangenstein
Michael Welcome

Lawrence Livermore National Laboratory
Livermore, CA 94550

This Paper was Prepared for Presentation at the AIAA 9th Computational Fluid Dynamics Conference in Buffalo, New York June 14-16, 1989.

April 1989



Adaptive Mesh Refinement on Moving Quadrilateral Grids†

John Bell
Phillip Colella‡
John Trangenstein
Michael Welcome

Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94550

Abstract

This paper describes an adaptive mesh refinement algorithm for unsteady gas dynamics. The algorithm is based on an unsplit, second-order Godunov integration scheme for logically-rectangular moving quadrilateral grids. The integration scheme is conservative and provides a robust, high resolution discretization of the equations of gas dynamics for problems with strong nonlinearities. The integration scheme is coupled to a local adaptive mesh refinement algorithm that dynamically adjusts the location of refined grid patches to preserve the accuracy of the solution, preserving conservation at interfaces between coarse and fine grids while adjusting the geometry of the fine grids so that grid lines remain smooth under refinement. Numerical results are presented illustrating the performance of the algorithm.

Introduction

Over the last decade, there has been substantial effort in the development of upwind finite difference methods for invsicid gas dynamics. Higher-order versions of Godunov's scheme are among the more successful methods of this type for solving unsteady problems. These methods incorporate the nonlinear wave propagation properties of the compressible Euler equations, in the form of characteristic equations and Riemann problems into robust conservative difference algorithms. The second- and higher-order versions of these methods resolve discontinuities accurately, while providing accurate representation of smooth flows. Particularly when combined with adaptive methods, they have been effective in resolving complex combinations of discontinuities and smooth solutions in simple rectangular geometries.

The goal of the research described in this paper is the development of an adaptive mesh refinement algorithm for use with higher-order Godunov methods on moving quadrilateral grids. Our algorithm is an extension of that of of Berger and Colella [1]. In their approach the grid is locally refined in space and time in response to the appearance of large errors or other features in the solution. Richardson extrapolation based on coarsening the grid is used to identify and tag cells where refinement is required based on a specified error criterion. The tagged cells are covered by the union of a small number of rectangular patches, that are then refined. Refined regions may change, appear or disappear as a function of time. Previously, Berger and Jameson [2] developed an adaptive mesh refinement algorithm of the type discussed here for quadrilateral meshes. However, they took advantage of the fact that that they were computing steady flows to simplify the algorithm. In the present work, our target application is genuinely unsteady flows with strong shocks.

The quadrilateral adaptive mesh refinement algorithm (QAMR) is based on a composite of several coarse, logically-rectangular grids. These grids are assumed to fit together smoothly with no overlap. Typically, coarse grids are generated using an elliptic grid generation technique modified to ensure smoothness of the grid in time and across grid boundaries. The principal difficulty in the method is the necessity to maintain smoothness of the refined grids in order to preserve the accuracy of the integrator. Refined grids inherit their geometry from the coarse grid; however, a simple subdivision of the coarse cells is inadequate to maintain sufficient smoothness of the refined grids. Instead, we

† This work of this author 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 IACROs 89-848 and 89-849.

‡ Member, AIAA. Permanent address: Dept. of Mechanical Engineering, University of California at Berkeley

use a smooth interpolation based on Hermite cubic splines to define refined grid patches. Thus, the refinement of a coarse cell need not fit exactly inside the original coarse cell. This necessitates particular care in communicating information between coarse and fine grids to guarantee local conservation.

The integration scheme that forms the basis for our adaptive mesh refinement algorithm is the second-order, unsplit Godunov method for logically-rectangular, moving quadrilateral grids developed by Trangenstein and Colella [3]. This algorithm is based on an upstream-centered predictor-corrector formalism, with the conservative corrector step performed using finite-volume differencing. The method is second-order accurate for smooth solutions on smooth grids, has a robust treatment of discontinuities with minimal dissipation, and is freestream-preserving.

Equations of motion

The unsteady two-dimensional Euler equations can be written in conservation law form as follows:

$$U_t + \nabla\cdot\mathbf{F} = 0 \qquad (2.1)$$

$$U(x,t) = U:\mathbf{R}^2\times[0,T]\rightarrow\mathbf{R}^M$$

$$\mathbf{F} = (F^x,F^y) \in \mathbf{R}^M\times\mathbf{R}^M$$

with $M = 4$, and where the dependent variables U and the fluxes F^x, F^y are

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

$$F^y(U) = \begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vE+vp\end{bmatrix}$$

where ρ is the density, $(u,v) = \mathbf{u}$ are the x and y components of velocity, and E is the total energy per unit mass. The pressure is derived from these quantities *via* an equation of state, $p=p(\rho,e)$, where e is the internal energy per unit mass, given by $e=E-\frac{1}{2}(u^2+v^2)$.

For each $\mathbf{n}\in\mathbf{R}^2$ we define the projected equations along $\mathbf{n}$ to be the one-dimensional system of conservation laws

$$\frac{\partial U}{\partial t} + \frac{\partial F^{\mathbf{n}}}{\partial\chi} = 0 \quad ; \quad F^{\mathbf{n}}(U) = \mathbf{n}\cdot\mathbf{F}(U) \qquad (2.2)$$

The projected equations for the system (2.1) are essentially those of gas dynamics in one dimension. If we project the equations in the $\mathbf{n}$ direction for $\mathbf{n}$ a unit vector, we can make a change of variables to obtain the following system equivalent to (2.2):

$$\frac{\partial W}{\partial t} + \frac{\partial G(W)}{\partial\chi} = 0 \qquad (2.3)$$

$$W = \begin{bmatrix}\rho\\ \rho u^{\mathbf{n}}\\ \rho u^{\mathbf{t}}\\ \rho E\end{bmatrix}\ ,\ G(W) = \begin{bmatrix}\rho u^{\mathbf{n}}\\ \rho u^{\mathbf{n}2}+p\\ \rho u^{\mathbf{n}}u^{\mathbf{t}}\\ \rho u^{\mathbf{n}}E+u^{\mathbf{n}}p\end{bmatrix}$$

where $u^{\mathbf{n}} = \mathbf{u}\cdot\mathbf{n}$, $u^{\mathbf{t}} = \mathbf{u}\cdot\mathbf{n}^{\perp}$ with the other variables defined as before. (Here, $(a,b)^{\perp} = (b,-a)$.) Since $\mathbf{n}$ is a unit vector, $u^2+v^2 = (u^{\mathbf{n}})^2+(u^{\mathbf{t}})^2$ so the formula for the internal energy e can use either quantity. From these equations, it is clear that the eigenvectors and eigenvalues of the linearized system, as well as the solution to the Riemann problem, are given by those for the one-dimensional gas dynamics equations, with $u^{\mathbf{t}}$ being treated as a passively advected quantity. Hence, we can use the techniques in [4] for calculating solutions to the Riemann problem and for manipulating characteristic variables.

Integration scheme

We assume that the computational domain is divided into a time-dependent set of quadrilateral cells with the corners located at $(x^n_{i+\frac{1}{2},j+\frac{1}{2}}, y^n_{i+\frac{1}{2},j+\frac{1}{2}})$. Here i,j are discrete spatial indices, and n is a discrete time index. Furthermore, we assume that there is a smooth transformation $(x(\xi,\eta,t),y(\xi,\eta,t))$ from some coordinate space to physical space and a uniform rectangular grid $(\xi^{i+\frac{1}{2}},\eta^{j+\frac{1}{2}})$ in the coordinate space such that the computational grid points are mapped to the physical grid points at each time n. The equations of gas dynamics (2.1) transformed to the (ξ,η) coordinate space, can be expressed as follows.

$$\frac{\partial JU}{\partial t} + \frac{\partial F^{\xi}}{\partial\xi} + \frac{\partial F^{\eta}}{\partial\eta} = 0 \qquad (3.1)$$

$$F^{\xi} = \mathbf{n}^{\eta}\cdot(\mathbf{F}-sU)\ ,\ F^{\eta} = \mathbf{n}^{\xi}\cdot(\mathbf{F}-sU),$$

$$\mathbf{n}^{\xi} = (y_{\eta},-x_{\eta})\quad \mathbf{n}^{\eta} = (-y_{\xi},x_{\xi})\quad \mathbf{s} = (x_t,y_t)$$

$$J = \det(\nabla_{\xi,\eta}(x,y)) > 0\ .$$

We define finite difference approximations to the derivatives of the coordinate map as follows.

$$(\Delta^{\xi}\mathbf{x})_{i,j+\frac{1}{2}} = \mathbf{x}_{i+\frac{1}{2},j+\frac{1}{2}}-\mathbf{x}_{i-\frac{1}{2},j+\frac{1}{2}} \approx \Delta\xi\frac{\partial\mathbf{x}}{\partial\xi}(\xi_i,\eta_{j+\frac{1}{2}})$$

$$(\Delta^{\eta}\mathbf{x})_{i+\frac{1}{2},j} = \mathbf{x}_{i+\frac{1}{2},j+\frac{1}{2}}-\mathbf{x}_{i+\frac{1}{2},j-\frac{1}{2}} \approx \Delta\eta\frac{\partial\mathbf{x}}{\partial\eta}(\xi_{i+\frac{1}{2}},\eta_j)$$

$$\mathbf{s}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} = (\mathbf{x}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \mathbf{x}^{n}_{i+\frac{1}{2},j+\frac{1}{2}})/\Delta t$$

$$\sigma^n_{i,j} = \Sigma(\mathbf{x}^n_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{x}^n_{i+\frac{1}{2},j-\frac{1}{2}}, \mathbf{x}^n_{i-\frac{1}{2},j-\frac{1}{2}}, \mathbf{x}^n_{i-\frac{1}{2},j+\frac{1}{2}}) \qquad (3.2)$$

where $\Sigma(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) \equiv \frac{1}{2}(\mathbf{c}-\mathbf{a})\times(\mathbf{b}-\mathbf{d})$ is the signed area of the quadrilateral defined by the four points. Using these finite differences, we can make the connection between the mapping derivative appearing in the

transformed equations and the geometry of the finite difference grid in physical space: $\sigma_{i,j}^n \approx J(\xi_i,\eta_j,t^n)\Delta\xi\Delta\eta$ is the area of the $(i,j)^{th}$ zone, and $\mathbf{n}^\xi\Delta\xi_i \approx (\Delta^\xi \mathbf{x}^n)^\perp_{i+\frac{1}{2},j}$, $\mathbf{n}^\eta\Delta\eta_j \approx -(\Delta^\eta \mathbf{x}^n)^\perp_{i,j+\frac{1}{2}}$, are perpendicular to the cell edges with length equal to the length of the edge.

As in [5] we seek a conservative, finite volume, predictor-corrector discretization of (3.1), in which the predictor step calculates the flux at the cell edges, and the corrector step performs the conservative, finite-volume upate of U.

The form of the corrector is suggested by the following evolution equation for J.

$$J_t - (\mathbf{s}\cdot\mathbf{n}^\xi)_\eta - (\mathbf{s}\cdot\mathbf{n}^\eta)_\xi = 0 \quad . \tag{3.3}$$

The form of the discretization of (3.2) is suggested by the interpretation of $J_{i,j}\Delta\xi\Delta\eta$ as the volume of the $(i,j)^{th}$ cell:

$$\sigma_{i,j}^{n+1} = \sigma_{i,j}^n - \delta\sigma_{i-\frac{1}{2},j} + \delta\sigma_{i+\frac{1}{2},j} - \delta\sigma_{i,j-\frac{1}{2}} + \delta\sigma_{i,j+\frac{1}{2}} \tag{3.4}$$

where

$$\delta\sigma_{i+\frac{1}{2},j} = \Sigma(\mathbf{x}^n_{i+\frac{1}{2},j-\frac{1}{2}},\mathbf{x}^n_{i+\frac{1}{2},j+\frac{1}{2}},\mathbf{x}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}},\mathbf{x}^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}})$$

$$\delta\sigma_{i,j+\frac{1}{2}} = \Sigma(\mathbf{x}^n_{i+\frac{1}{2},j+\frac{1}{2}},\mathbf{x}^n_{i-\frac{1}{2},j+\frac{1}{2}},\mathbf{x}^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}},\mathbf{x}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}})$$

The difference approximation (3.4) is a discrete conservation law for the volume of the system. This formula also motivates the conservative corrector step

$$(\sigma U)_{i,j}^{n+1} = (\sigma U)_{i,j}^n + F_{i-\frac{1}{2},j} - F_{i+\frac{1}{2},j} + F_{i,j-\frac{1}{2}} - F_{i,j+\frac{1}{2}} \tag{3.5}$$

where

$$F_{i+\frac{1}{2},j} = \Delta t\,(\Delta^\eta \mathbf{x}^{n+\frac{1}{2}})^\perp_{i+\frac{1}{2},j}\cdot\mathbf{F}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - \delta\sigma_{i+\frac{1}{2},j}U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$$

and

$$F_{i,j+\frac{1}{2}} = -\Delta t\,(\Delta^\xi \mathbf{x}^{n+\frac{1}{2}})^\perp_{i,j+\frac{1}{2}}\cdot\mathbf{F}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \delta\sigma_{i,j+\frac{1}{2}}U^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$$

If $U^n = U^{n+\frac{1}{2}} \equiv U_0$ and $\mathbf{F}^{n+\frac{1}{2}} \equiv \mathbf{F}(U_0)$ then $U^{n+1} = U^n$. This is most easily seen by the fact that if $\mathbf{F}^{n+\frac{1}{2}}$ is a constant, then the terms involving $\mathbf{F}^{n+\frac{1}{2}}$ on the right hand side of (3.5) sum to zero. Given that, then the assertion follows directly from (3.4). Thus, if our predictor step for computing $U^{n+\frac{1}{2}}$ and $\mathbf{F}^{n+\frac{1}{2}}$ also preserves constants, then the overall algorithm is freestream-preserving.

Predictor step

Our overall strategy for computing the quantities $U^{n+\frac{1}{2}}$, $\mathbf{F}^{n+\frac{1}{2}}$ required for the fluxes follows closely that given in [5]. The principal change involves properly accounting for the grid motion in the predictor step so as to maintain freestream preservation. The algorithm is only outlined here; for details see [3].

We want to extrapolate time-centered values

$$U^{n+\frac{1}{2},S}_{i+\frac{1}{2},j} \quad , \quad U^{n+\frac{1}{2},S}_{i,j+\frac{1}{2}} \quad S=L,R$$

given the values $U^n_{i,j}$ at cell centers. The values so obtained will then be used as left and right states for the Riemann problem at the cell edges for the system (2.3) projected in the direction of the time-centered normal.

The construction of $U^{n+\frac{1}{2},L}_{i+\frac{1}{2},j}$ is obtained starting from a Taylor series expansion.

$$\begin{aligned} U(\xi_{i+\frac{1}{2}},\eta_j,t^n+\frac{\Delta t}{2}) &= U^n_{i,j} + \frac{\Delta\xi}{2}\frac{\partial U}{\partial\xi} + \frac{\Delta t}{2}\frac{\partial U}{\partial t} \\ &= U^n_{i,j} + \frac{\Delta\xi}{2}\frac{\partial U}{\partial\xi} - \frac{\Delta t}{2J}\left[U\frac{\partial J}{\partial t} + \frac{\partial F^\xi}{\partial\xi} + \frac{\partial F^\eta}{\partial\eta}\right] \\ &= U^n_{i,j} + \frac{1}{2}\left[I - \frac{\Delta t}{\Delta\xi J}(\mathbf{n}^\eta\cdot\mathbf{A} - \mathbf{s}\cdot\mathbf{n}^\eta I)\right]\frac{\partial U}{\partial\xi}\Delta\xi \\ &\quad - \frac{\Delta t}{2J}\left[\frac{\partial \mathbf{n}^\xi}{\partial\xi}\cdot\mathbf{F} + U\frac{\partial}{\partial\eta}(\mathbf{s}\cdot\mathbf{n}^\xi) + \frac{\partial F^\eta}{\partial\eta}\right] \end{aligned}$$

$$\mathbf{A} = \nabla_U \mathbf{F}$$

In the last step, we have put $\frac{\partial F^\xi}{\partial\xi}$ in nonconservation form and used (2.5) to eliminate $\frac{\partial J}{\partial t}$.

As was done in [5], we split the predictor into two steps.

$$\hat{U}^L_{i+\frac{1}{2},j} = U^n_{i,j} + \frac{1}{2}\left[I - \frac{\Delta t}{\Delta\xi J}(\mathbf{n}^\eta\cdot\mathbf{A} - \mathbf{s}\cdot\mathbf{n}^\eta I)\right]\frac{\partial U}{\partial\xi}\Delta\xi \tag{3.6}$$

$$U^{n+\frac{1}{2},L}_{i+\frac{1}{2},j} = \hat{U}^L_{i+\frac{1}{2},j} - \frac{\Delta t}{2J}\left[\frac{\partial \mathbf{n}^\xi}{\partial\xi}\cdot\mathbf{F} + U\frac{\partial}{\partial\eta}(\mathbf{s}\cdot\mathbf{n}^\xi) + \frac{\partial F^\eta}{\partial\eta}\right]$$

In the first part of the predictor step, we approximate $\frac{\partial U}{\partial\xi}$ by central differences to which monotonicity constraints have been applied. In addition, the components of $\frac{\partial U}{\partial\xi}$ corresponding to the waves moving away from the cell edge are projected out, and the overall calculation is performed in terms of primitive variables. In the second part of the predictor step, we approximate $\frac{\partial F^\eta}{\partial\eta}$ by a Godunov-type finite-volume differencing similar to that used in the corrector step. The geometric terms $\frac{\partial \mathbf{n}^\xi}{\partial\xi}$ and $\frac{\partial}{\partial\eta}(\mathbf{s}\cdot\mathbf{n}^\xi)$ are differenced so that they exactly cancel the difference approximation to $\frac{\partial F^\eta}{\partial\eta}$ if there are no gradients of U in the η direction. Thus, the predictor step is itself freestream-preserving in the sense that $U^{n+\frac{1}{2},S}_{i+\frac{1}{2},j} = U_0$ if $U^n = U_0$ in all zones sufficiently close to the $(i,j)^{th}$ zone.

AMR on quadrilaterals

QAMR is based on using a hierarchy of nested, logically rectangular grids on which the equations (2.1) are discretized. The grid hierarchy is defined logically by D_l, $l = 1, \ldots, l_{max}$, where each D_l is a collection of points in $\mathbf{Z}^2$. Each D_l is represented as the union of overlapping rectangles in $\mathbf{Z}^2$; however, the results obtained from the calculation are independent of the particular representation. We identify points at successive levels l, $l+1$ by a map $\phi: D_{l+1} \to D_l$, $\phi(\mathbf{i}) = \mathbf{i} / r$, where r is an even integer, called the refinement ratio. Thus D_{l+1} corresponds to a grid refined by a factor of r in each coordinate direction over D_l, and

$$\phi^{-1}(\mathbf{i}) = \{r\mathbf{i} + \mathbf{p} : \mathbf{p} = (0,0), \ldots, (r-1,r-1)\}$$

is the set of cells in D_{l+1} contained in the $\mathbf{i}^{th}$ cell of D_l. The grids are nested, in the sense that $\phi(D_{l+1}) \subset D_l$, $\phi^{-1}(\phi(D_{l+1})) = D_{l+1}$. In addition, we require them to be properly nested, i.e., that, for $l>1$, a one-cell wide buffer separate $\phi(D_{l+1})$ from the boundary of D_l, except for those parts of the boundary coinciding with boundaries of the physical domain of the problem. Refinement is done in time, as well as in space: if Δt_l is the time step at level l, then $\Delta t_l = \Delta t_1 / r^{l-1}$. In particular, the time steps can be chosen so that our explicit finite difference algorithm is stable on the entire grid hierarchy.

We take our basic dependent variables to be U^l, the conserved quantities, and σ^l, the cell volumes, defined for all points in D_l. The grid corners $\mathbf{x}^l$ for D_l are also computed at each time step for which (U^l, σ^l) is defined, with $\mathbf{x}^{l+1}_{r\mathbf{i}-(\frac{1}{2},\frac{1}{2})} = \mathbf{x}^l_{\mathbf{i}-(\frac{1}{2},\frac{1}{2})}$. At the lowest level, $l=1$, the location of the grid corners $\mathbf{x}^1$ are initially determined and moved by some type of mesh generation algorithm. Possible approaches include some form of variational mesh generation scheme or a simple algebraic or conformal mapping. Grids points at levels $l>1$ are determined by interpolation of the level $l-1$ grid point locations. To compute the location of the corner of grid cells in $\phi^{-1}(\mathbf{i})$ for some $\mathbf{i}$ we first define approximation to $\mathbf{x}_\xi$ and $\mathbf{x}_\eta$ at the corners of the $\mathbf{i}^{th}$ cell using a centered difference approximation. These derivatives are then used to define a Hermite bicubic spline mapping the unit square to physical space that maps the corners of the square to the corners of the $\mathbf{i}^{th}$ cell. The image of an $r \times r$ grid on this unit square defines the location of the grid point in $\phi^{-1}(\mathbf{i})$. This interpolation scheme is completely local, using only the cell and its four neighbors; however, the spline is sufficiently smooth across cell boundaries to ensure that the refined grid point locations remain smooth after refinement. A similar approach is used to smooth the trajectories of the level $l>1$ grid points in time.

We emphasize that it is not necessarily the case that σ^l is related to $\mathbf{x}^l$ by (3.2). In particular, in advancing the solution in time, we assume that the cell volumes σ^l are equal to the volume enclosed by the finest grid edges surrounding that cell.

Integration step

The AMR algorithm for advancing the solution on the grid hierarchy described above can be formulated as being recursive in the level of refinement. On a given level of refinement l, the algorithm can be broken up into three steps.

Step 1. Advance (U^l, σ^l) by one time step, applying the finite difference scheme (3.5) - (3.6) to the data on each of the rectangular grids making up D_l. However, we modify the scheme by having the cell volumes be given by σ^l, rather than by (3.2), at the beginning of the time step, and by updating them using (3.4). With that modification, the scheme remains well-defined, and is freestream-preserving. Boundary values for the individual rectangular grids are obtained from the rest of the level l solution, or from user-specified boundary conditions if the grid abuts a physical boundary. Any remaining boundary values are interpolated in space and time from the solution at levels $l' < l$.

Step 2. Advance (U^{l+1}, σ^{l+1}) by r time steps, so that the latest values of the level $l+1$ solution are known at the same time as the level l solution obtained in Step 1.

Step 3. Modify the solution values obtained in Step 1 to be consistent with the level $l+1$ solution. This is necessary for (i): cells that have been refined, or (ii): cells which have not been refined, but share one or more edges with refined cells. In case (i), we the value at level l is defined to be the conservative average over the level $l+1$ cells contained in it of the level $l+1$ solution.

$$\sigma^l_{\mathbf{i}} = \sum_{\mathbf{p}=(0,0)}^{(r-1,r-1)} \sigma^{l+1}_{r\mathbf{i}+\mathbf{p}} \qquad (4.1)$$

$$(\sigma U)^l_{\mathbf{i}} = \sum_{\mathbf{p}=(0,0)}^{(r-1,r-1)} (\sigma U)^{l+1}_{r\mathbf{i}+\mathbf{p}}$$

In case (ii), we modify the solution value in a coarse cell abutting a fine grid, in order to maintain conservation across grid refinement boundaries. For example, if cell $(i+1,j)$ is refined, then the correction to the values in (i,j) has the following form:

$$\sigma^l_{i,j} := \sigma^l_{i,j} - \delta\sigma^l_{i+\frac{1}{2},j} + \sum_{p=0}^{r-1}\sum_{q=0}^{r-1} \delta\sigma^{l+1,q}_{ri+\frac{1}{2},rj+p}$$

$$(\sigma U)^l_{i,j} := (\sigma U)^l_{i,j} + F^l_{i+\frac{1}{2},j} - \sum_{p=0}^{r-1}\sum_{q=0}^{r-1} F^{l+1,q}_{ri+\frac{1}{2},rj+p}$$

Here, q is a discrete time index on the level $l+1$ grid. In performing this correction, we are effectively using

the appropriate sum of the fine grid fluxes at $(i+\frac{1}{2},j)$ to perform the conservative difference step at (i,j).

Creating the mesh hierarchy

The logical structure of the grid generation is a straightforward generalization of the procedure discussed in [1]. A subset of the cells at level l are identified as requiring refinement; these are covered by the union of logical rectangles on which the coarse grid data must be interpolated in a conservative fashion, and data from the same level copied from the old grids. The principal difficulty in QAMR stems from the fact that when the location of the refined regions change, the cell volumes change at all coarser levels. This requires us to modify the conserved quantities, for example, to maintain freestream preservation. Our strategy for dealing with this problem is to initially preprocess the old grid data on each level so that it is consistent with cell volumes defined by (3.2). We then use the inverse of this preprocessing step to obtain values on the new grid hierarchy that are consistent with the grid geometry one level above. Those values are used to interpolate new values as needed at the next level; in addition, values are copied from the old grids at the next level which overlap the new ones. After the new grids at all the levels have been set in this fashion, we recursively apply (4.1), starting at the finest level. The overall construction is conservative, freestream-preserving, and transforms the cell volumes so that the volume of a cell at any level is equal to the volume of the region enclosed by the edges of the finest grids surrounding that cell.

In the preprocessing step, we adjust the conserved quantities on each level so that they are consistent with cell volumes given by (3.2). This is done recursively in level, starting at the finest level.

$$\sigma_{\mathrm{i}}^{l} = \sum_{\mathrm{p}=(0,0)}^{(r-1,r-1)} \sigma_{r\mathrm{i}+\mathrm{p}}^{l+1,old}$$

$$(\sigma U)_{\mathrm{i}}^{l} = \sum_{\mathrm{p}=(0,0)}^{(r-1,r-1)} (\sigma U)_{r\mathrm{i}+\mathrm{p}}^{l+1,old}$$

$$\sigma_{i,j}^{l} := \sigma_{i,j}^{l} - \delta\sigma_{i+\frac{1}{2},j}^{l,l+1} + \delta\sigma_{i-\frac{1}{2},j}^{l,l+1} - \delta\sigma_{i,j+\frac{1}{2}}^{l,l+1} + \delta\sigma_{i,j-\frac{1}{2}}^{l,l+1}$$

$$(\sigma U)_{i,j}^{l} := (\sigma U)_{i,j}^{l} - \delta U_{i+\frac{1}{2},j}^{l} + \delta U_{i-\frac{1}{2},j}^{l} - \delta U_{i,j+\frac{1}{2}}^{l} + \delta U_{i,j-\frac{1}{2}}^{l}$$

$$l = l_{max}-1, \ldots, 1$$

Here, $\delta\sigma^{l,l+1}$ is the signed area between the cell edges given by x^{l} and a cell edge given by x^{l+1}. (See Figure 1.) On cell edges that are not adjacent to refined cells, $\delta\sigma^{l,l+1}$ is set to zero. The δU^{l}'s are computed to be the amount of conserved quantity contained in $\delta\sigma^{l,l+1}$.

$$\delta U_{i+\frac{1}{2},j} = \delta\sigma_{i+\frac{1}{2},j}^{l,l+1} U_{i+s,j}^{l}$$

$$s = \begin{cases} 1 & \text{if } \delta\sigma_{i+\frac{1}{2},j}^{l,l+1} < 0 \\ 0 & \text{otherwise} \end{cases}$$

We emphasize that this procedure was performed on the old grid hierarchy. To generate values for the new grid hierarchy, we begin by performing a partial inverse to the preprocessing step described above.

$$\tilde{\sigma}_{i,j}^{l} := \sigma_{i,j}^{l} + \delta\sigma_{i+\frac{1}{2},j}^{l,l+1} - \delta\sigma_{i-\frac{1}{2},j}^{l,l+1} + \delta\sigma_{i,j+\frac{1}{2}}^{l,l+1} - \delta\sigma_{i,j-\frac{1}{2}}^{l,l+1}$$

$$(\tilde{\sigma U})_{i,j}^{l} := (\sigma U)_{i,j}^{l} - \delta U_{i+\frac{1}{2},j}^{l} + \delta U_{i-\frac{1}{2},j}^{l} - \delta U_{i,j+\frac{1}{2}}^{l} + \delta U_{i,j-\frac{1}{2}}^{l}$$

Here, $\delta\sigma^{l,l+1}$ and δU are defined as in the preprocessing step, except that the U^{l}'s consistent with the geometric definition of σ^{l} are used, and the new grid hierarchy replaces the old one. Both this step, and the preprocessing step, are manifestly conservative, being in flux form, and are freestream-preserving. The $\tilde{\sigma}^{l}$, $\tilde{U}^{l}$'s are used to perform a conservative piecewise linear interpolation onto the level $l+1$ grid.

$$\tilde{U}_{r\mathrm{i}+\mathrm{p}}^{l+1} = \tilde{U}_{\mathrm{i}}^{l} + \frac{\sigma_{\mathrm{i}}^{l}}{r^{3}\sigma_{r\mathrm{i}+\mathrm{p}}^{l+1}} (\Delta^{\xi}U \ , \Delta^{\eta}U)\cdot(\mathrm{p}-\tfrac{1}{2}(r-1,r-1))$$

$$\mathrm{p}=(0,0), \ldots, (r-1,r-1)$$

Here, $\Delta^{\xi}U$, $\Delta^{\eta}U$ are undivided central differences to which monotonicity constraints have been applied. Finally, the values of U^{l+1} from the old grid hierarchy are copied onto the new level $l+1$ grids in regions where the two overlap.

The above procedure is applied recursively in l, beginning at the coarsest level being refined, until all the refined grid values have been generated. Then a final recursive pass through the grid hierarchy is made, so that the new values are consistent with the fine grid geometry.

$$(\sigma,U)^{l_{max}^{new}} = (\tilde{\sigma},\tilde{U})^{l_{max}}$$

$$\sigma_{\mathrm{i}}^{l,new} = \sum_{\mathrm{p}=(0,0)}^{(r-1,r-1)} \sigma_{r\mathrm{i}+\mathrm{p}}^{l+1,new}$$

$$(\sigma U)_{\mathrm{i}}^{l,new} = \sum_{\mathrm{p}=(0,0)}^{(r-1,r-1)} (\sigma U)_{r\mathrm{i}+\mathrm{p}}^{l+1,new}$$

$$l = l_{max}-1, \ldots, 1$$

Numerical results

Our first example is the reflection of a shock wave by a ramp. For this case, the ramp angle was 30°. The fluid is described by a γ-law gas with $\gamma = 1.4$. with an incident Mach 5.5 shock. This combination of shock strength and ramp angle are in the double Mach reflection regime. For this computation we began with a 50×40 level 1 grid. We used two levels of refinement with a refinement ratio of 4 for each level; thus, the effective resolution of the calculation corresponds to an 800×640 grid. In Figure 2 we show density contours for the composite grid structure. The location of the grids is indicated by dashed lines with long dashes for

the level 2 grids and short dashes for the level 3 grids. Note that the level three grids surround the reflected shock and the incident shock with very little wasted computation. There are also level 3 grids surrounding the region around the two triple points and the associated slip lines and Mach stems. The net result is an accurate depiction of the complete shock structure with a minimum of wasted computational effort. To estimate the savings associated with the adaptive mesh algorithm we can compare the total number of cells advanced in time with the number of cells that would have been advanced on a single grid calculation of comparable resolution. For this computation, the adaptive mesh algorithm advanced a total of 6.8×10^7 cells compared to 5.7×10^9 for a single grid computation. Adjusting the ratio of cells advanced by the 10% overheard associated with the adaptive mesh algorithm yields an reduction in computational effort by a factor of 7.5. Additional savings could be obtained by selectively restricting the location of the level 3 grids. For example, one could turn off the 2^{nd} level of refinement for the reflected shock away from the two triple points.

Our second example shows the diffraction of a shock over a 45° wedge. For this case the incident Mach number was 2.85. In Figure 3 a) - c). we show density contours for at three times as the flow expands around the top of the wedge. Here, there are two level 1 grids, each 50×50, one on the left and the other on the right. As in the first case, we used two levels of refinement by a factor of 4. For this example there is a strong expansion of the flow at top of the wedge. As the flow exands two unstable slip surfaces emerge. One corresponds to the stretched contact discontinuity from first triple-point. The second corresponds to the shear layer that forms as the flow separates at the top of the wedge. Each of these slip lines are Kelvin-Helmholtz unstable and are beginning to roll up. As in the previous example, the finest grids are clustered around the dominant discontinuities.

References

1. M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," UCRL-97196, LLNL, to appear in J. Comp. Phys..
2. M. J. Berger and A. Jameson, "Automatic Adaptive Refinement for the Euler Equations," *AIAA J.*, vol. 23, pp. 561-568, 1984.
3. J. A. Trangenstein and P. Colella, *An Unsplit, Second-Order Godunov Method for Moving Quadrilateral Grids*, in preparation.
4. P. Colella and H. M. Glaz, "Efficient Solution Algorithms for the Reimann Problem for Real Gases," *J. Comput. Phys.*, vol. 59, pp. 264-289, 1985.
5. P. Colella, "A Multidimensional Second Order Godunov Scheme for Conservation Laws," LBL-17023, Lawrence Berkeley Laboratory, to appear in J. Comp. Phys..

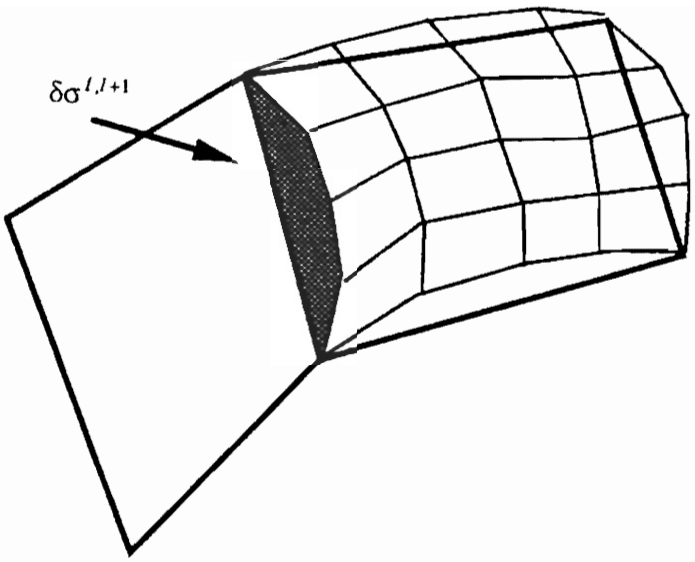


Fig. 1. Area adjustment at coarse-fine interface.

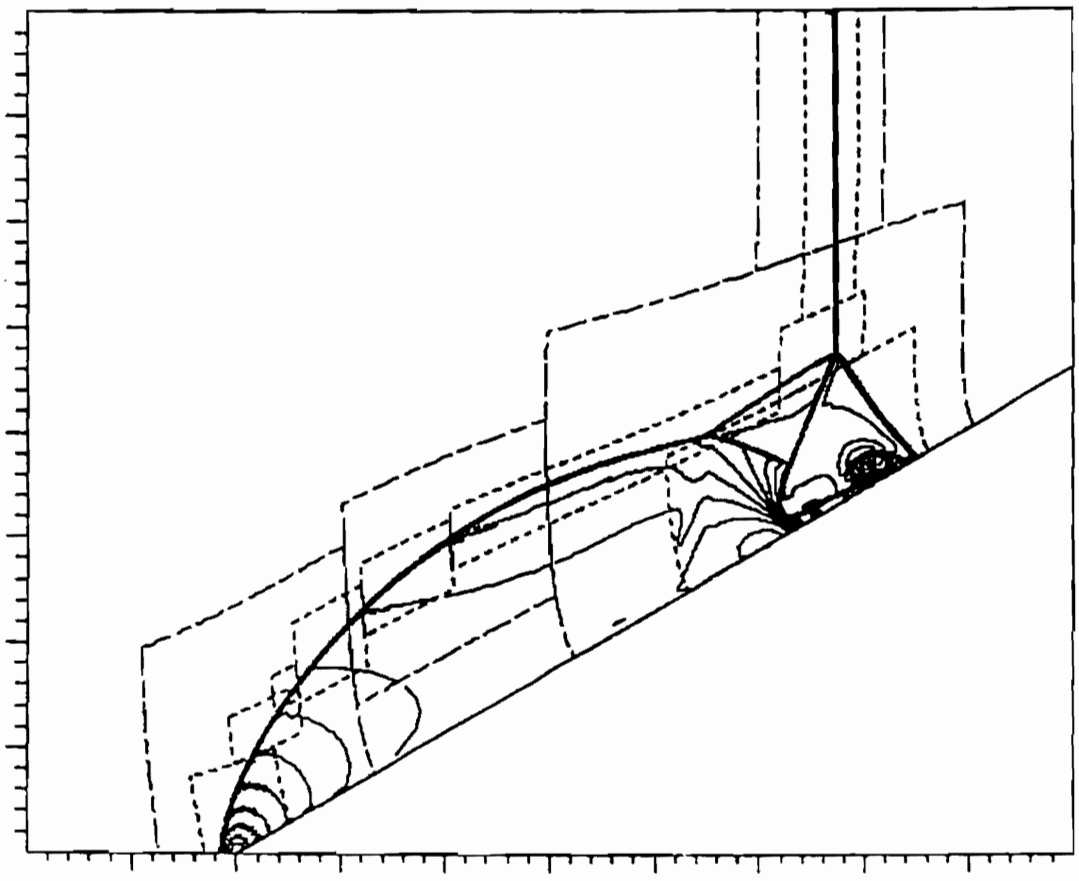


Fig. 2. Shock reflection by a 30° ramp, $M_\infty = 5.5$.

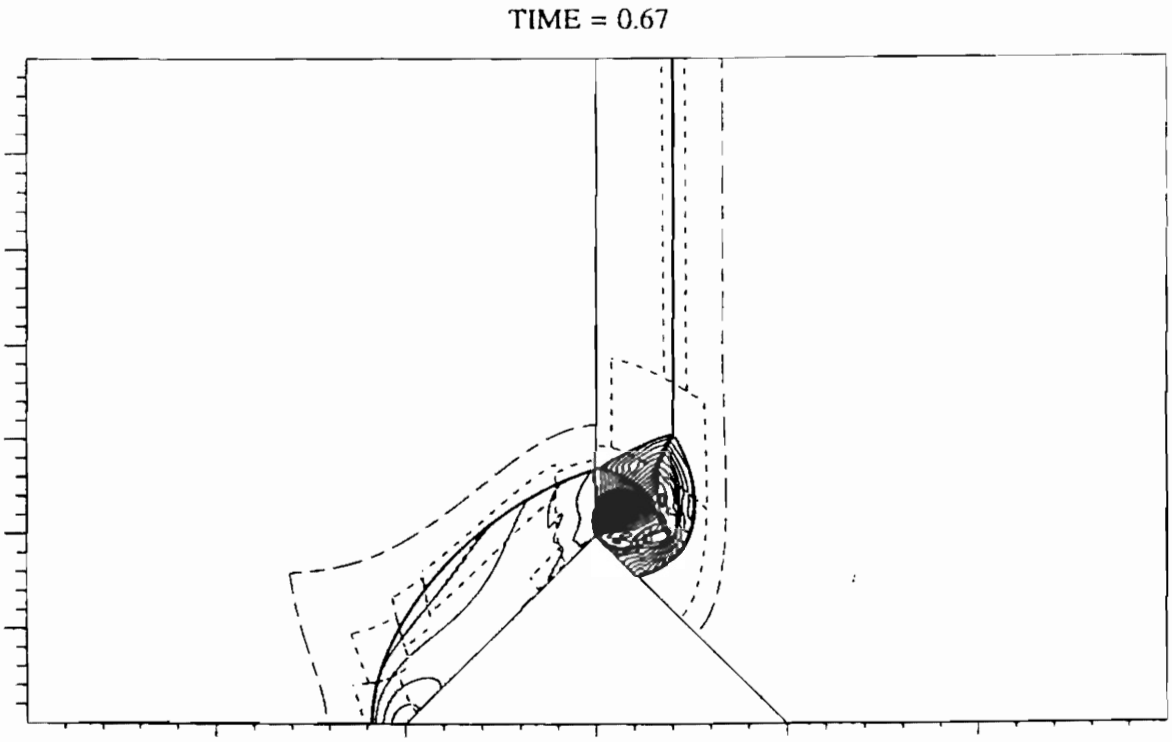


Fig. 3a. Shock diffraction by a wedge, t=0.67, M_∞ = 2.85.

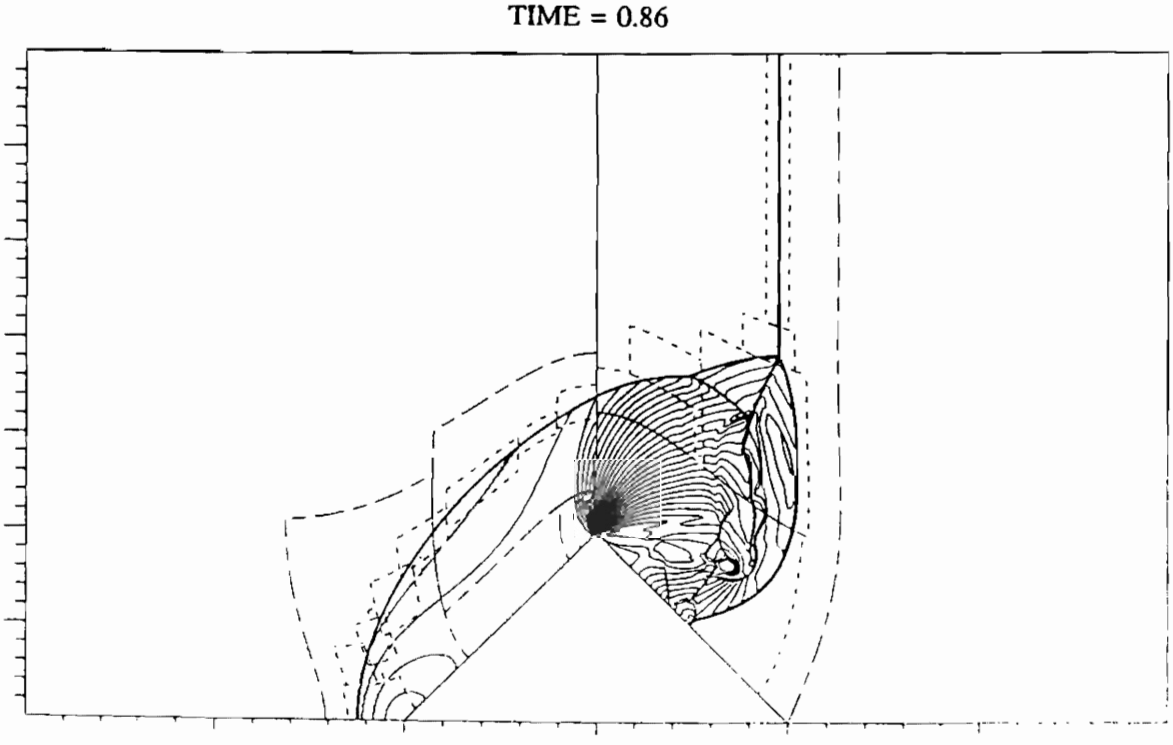


Fig. 3b. Shock diffraction by a wedge, t=0.86, M_∞ = 2.85.

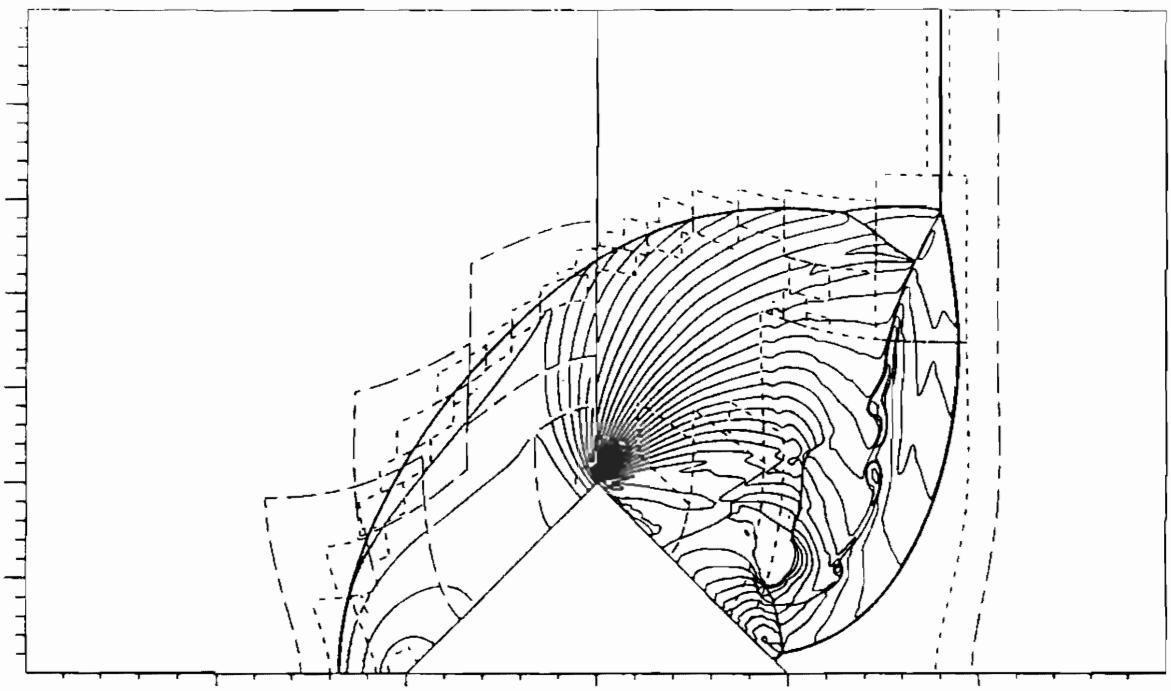


Fig. 3c. Shock diffraction by a wedge, t=1.16, M_∞ = 2.85.