JOURNAL OF COMPUTATIONAL PHYSICS 115, 153–168 (1994)

An Unsplit 3D Upwind Method for Hyperbolic Conservation Laws

JEFF SALTZMAN

Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received October 27, 1989; revised August 2, 1993

An unsplit upwind method for solving hyperbolic conservation laws in three dimensions is developed. This paper derives the algorithm by generalizing a two-dimensional advection algorithm of Van Leer and Colella to three dimensions and then making appropriate modifications. The method is implemented using the equations of gas dynamics. Several test problems are computed to both verify and display the behavior of the method. These test problems include a 1D blast wave, a 2D shock reflection off a 30° ramp, and a 3D astrophysical jet.

INTRODUCTION

This article describes an unsplit upwind method for solving hyperbolic conservation laws in three dimensions. The numerical solution of systems of hyperbolic conservation laws has a rich and long history. Even a book-length discussion of this topic would not sufficiently cover the subject because of the amount of work invested in the subject area. The reason why such a great deal of work has been done is the large number of times physical models yield mathematical descriptions in conservation form. Even models that do not strictly meet the structural criteria of hyperbolic conservation laws can often be studied using a simpler model in the correct form. A well-known example of this kind of simplification is the Navier–Stokes equations and their inviscid limit, the Euler equations.

In this paper, one path in the development of upwind methods is outlined. Its only purpose is to give the necessary background for this paper. Therefore the description of how upwind methods were developed and the many ways in which they have evolved is very incomplete. A more thorough description can be found in the review article of Lax, Harten, and van Leer [1]. The starting point in this summary is the method of Courant, Issacson, and Rees [2]. Although their paper is probably not the first to use upwind differencing (also called donor-cell differencing), it is the first to recognize the connection between characteristics of partial differential equations and proper differencing. Later Godunov generalized the method of Courant, Isaccson, and Rees to systems of hyperbolic conservation laws [3]. Godunov solved the one-dimensional Lagrangian equations of inviscid fluid dynamics using the nonlinear generalization of upwinding—the solution of the Riemann problem. Methods using solutions of Riemann problems are often collectively referred to as Godunov methods. It is important to note the methods described in this paragraph are all first order.

Only after much time did work continue along the path described above. Because of the slow rates of convergence of first-order methods and the apparent complexity of the Riemann problem, numerical calculations were usually made with second-order methods augmented with an artificial dissipation (Richtmyer and Morton [4]). Although artificial dissipation methods have a mathematical basis, they all seem to require adjustment from problem to problem. If there is too much dissipation added to the problem, structures of interest are diffused or eliminated. If too little dissipation is added, numerically generated solutions contain unphysical oscillations. Van Leer decided to attack the problem of numerical oscillations using higher order monotone schemes. Monotone schemes, by their design, do not allow spurious oscillations. In his search for higher order monotone schemes, he resurrected the work of Godunov. Van Leer's work generalizes Godunov's method to second order using a method called MUSCL [5]. Colella and Woodward simplified the method and increased the spatial accuracy to third order [6]. Colella and Woodward have further argued that their methods are competitive with or superior to other numerical methods for the solution of the equations of gas dynamics in their review article [7].

In [8] Colella introduced an advection scheme and its corresponding generalization to hyperbolic systems in two dimensions. The scheme for the advection equation is an unsplit second-order advection method set within a predictor–corrector framework. The advection scheme is for tensor product meshes. The method for hyperbolic systems not only generalizes the advection scheme but also generalizes the geometry to any smooth two-dimensional curvilinear coordinate system. Colella's rationale for this departure from operator splitting is twofold. Unsplit schemes are customarily used in a variety of applications, yet these same

0021-9991/94 $6.00

areas are not benefiting from the advantages of upwind methods. Colella also has specific target applications which are front tracking and implicit–explicit algorithms. In addition, unsplit algorithms can preserve symmetries normally destroyed by splitting. In the 3D test problem presented in this paper, symmetries are preserved using the unsplit method that are not preserved in calculations found in the literature. This paper is an extension to three-dimensional tensor product grids of Colella's unsplit method given in [8]. The following sections describes Colella's 2D unsplit advection scheme and its generalization to tensor-product grids in three dimensions. The 3D advection scheme is then generalized to 3D systems of conservation laws. The use of curvilinear coordinates is left for future work. Van Leer [9] outlined unsplit advection schemes in two and three dimensions identical to those presented here and in [8]. His generalization to systems takes a different direction.

The remainder of this paper is divided into several sections. The first section briefly describes hyperbolic conservation laws and a solution with special initial conditions called the Riemann problem. Because of the complexity of the three-dimensional difference schemes a section is devoted to notation. The notation is designed in a manner similar to indicial notation in tensor calculus where the same problem of notational complexity is present. After the section on notation the brief overview of advection schemes and their corresponding generalizations is retraced in more detail. This section is augmented with the details of the three-dimensional difference scheme and the corresponding hyperbolic system solver. The method is implemented for the Euler equations of gas dynamics in the following section. Finally, some examples are presented for calculations in one, two, and three dimensions.

HYPERBOLIC SYSTEMS

Three-dimensional hyperbolic systems of equations arise in many areas. Our attention is restricted to those systems which have the form

$$\frac{\partial U}{\partial t}+\frac{\partial F_1(U)}{\partial x}+\frac{\partial F_2(U)}{\partial y}+\frac{\partial F_3(U)}{\partial z}=0, \tag{1}$$

where

$$U\colon R^3\times[0,\infty]\Rightarrow R^m, \qquad F_{1,2,3}\colon R^m\Rightarrow R^m$$

and whose initial conditions are

$$U(x, y, z, t=0)=U_0(x, y, z).$$

These equations are categorized as being in conservation form and are often called conservation laws. The solutions that arise from these equations can be discontinuous and nonunique. Care must be taken to choose the correct physical solutions. An extensive discussion of hyperbolic systems can be found in [10].

An important subproblem of (1), given the initial conditions

$$U(x,0)=\begin{cases}U_{\mathrm{R}}, & \text{if } x>0,\\ U_{\mathrm{L}}, & x<0,\end{cases} \tag{2}$$

is called the Riemann problem. The Riemann problem plays a central role in the difference schemes derived below. The solution is a nonlinear combination of left and right moving waves that can be quite complicated. Analytic solutions for many interesting systems of equations are unknown. For strictly hyperbolic systems with initial conditions that are sufficiently close to each other, there does exist a unique solution.

Fortunately only a small amount of information is required from the solution of the Riemann problem and many problems that have very complicated Riemann problems can still be efficiently approximated. There is a large body of literature available for the approximate solution of the Riemann problem [11–15].

NOTATION

Notation is introduced which allows the concise expression of difference schemes in one, two, or three dimensions. The domains of these finite difference methods are lattices. The lattices may be as general as tensor products of nonuniform one-dimensional grids. All of the schemes can be written for these general grids. For simplicity the lattices are restricted to be tensor products of uniform but possibly unequally spaced one-dimensional grids. The cartesian grid spacings are indexed as

$$\Delta x,\ \Delta y,\ \Delta z \Leftrightarrow \Delta x_1,\ \Delta x_2,\ \Delta x_3.$$

Lattice functions use subscripts for spatial descriptions and superscripts for temporal specification. A lattice function, possibly vector, can be represented as

$$U(i\,\Delta x, j\,\Delta y, k\,\Delta z, n\,\Delta t)\Leftrightarrow U^n_{i,j,k}.$$

The convention that spatial integer-valued subscripts are cell-centered while half integer-valued subscripts are face-centered is used. The special case where all the spatial indices are half integer values is called vertex centered. To simplify notation further, implied indexing is used. This means that if an index is missing, it has some default value. The default values are integer values with the following rule:

$$U=U^n_{i,j,k}.$$

An application of this rule is

$$U_{j-1}^{n+1/2} \Leftrightarrow U_{i,j-1,k}^{n+1/2}.$$

Periodicities in the indices appear in the two- and three-dimensional difference schemes. To exploit these periodicities, indexing functions mapping integers to index place holders are used. In three spatial dimensions the following index functions are introduced.

$$\alpha = \alpha(l)\colon \alpha(1) = i, \alpha(2) = j, \alpha(3) = k$$
$$\beta = \beta(l)\colon \beta(1) = j, \beta(2) = k, \beta(3) = i$$
$$\gamma = \gamma(l)\colon \gamma(1) = k, \gamma(2) = i, \gamma(3) = j.$$

The corresponding index functions in two dimensions are

$$\alpha = \alpha(l)\colon \alpha(1) = i, \alpha(2) = j$$
$$\beta = \beta(l)\colon \beta(1) = j, \beta(2) = i.$$

The number of space dimensions will always be clear from the context.

Only two difference operators are needed in this paper. They are the backward difference operator and the translation operator. The difference operator is limited to spatial differences. The backward difference operator is defined as

$$\Delta_\alpha U = U - U_{\alpha-1}, \qquad l = 1, 2, 3.$$

The translation operator is defined as

$$T_\alpha U = U_{\alpha+1}, \qquad l = 1, 2, 3.$$

It is necessary for indices of the lattice spacing, flux functions, and coefficients of the advection equations to be periodic with the appropriate spatial dimension N. For $N = 2$ or 3 we define

$$\Delta x_l = \Delta x_{l \bmod N}, \qquad F_l = F_{l \bmod N}, \qquad c_l = c_{l \bmod N}.$$

Finally an "inverse velocity" is defined:

$$\lambda_l = \Delta t / \Delta x_l.$$

DIFFERENCE SCHEMES

The starting point for this discussion of upwind schemes is the method of Courant, Issacson, and Rees [2]:

$$\frac{u^{n+1} - u}{\Delta t} + c\frac{\Delta_l u}{\Delta x} = 0, \qquad c > 0.$$

This method has many geometric and algebraic interpretations. The geometric picture in Fig. 1 is very appealing. By interpreting u as a piecewise constant mass density, the shaded areas in Fig. 1a represent the amount of mass $u\,\Delta x$. Figure 1b shows the mass distributions after translation a distance $c\,\Delta t$. After translation, the mass is summed within each cell and a new mass density u^{n+1} is found by dividing the mass by the width of the cell. The new mass density distribution is displayed in Fig. 1c.

Using this geometric approach most of the notable properties of this method can be seen from inspection. If the CFL (Courant, Friedrichs, Lewy) number, $c\,\Delta t/\Delta x$, does not exceed one, monotonicity of the solution can be observed from the fact that new mass densities are positively weighted sums of the old mass densities and from consistency the positive weights sum to unity. Monotonicity ensures stability in the discrete maximum norm. Since the scheme is both stable and consistent, it is convergent. Conservation is easily seen because mass leaving one cell goes into the next. The first-order rate of convergence is ascertained from the first-order truncation error of the scheme.

The method of Godunov is a generalization of the method of Courant, Issacson, and Rees. The difference equation for a hyperbolic system in one dimension is

$$\frac{U^{n+1} - U}{\Delta t} + \frac{\Delta_i F_1(U_{i+1/2}^{n+1/2})}{\Delta x} = 0,$$

where

$$U_{i+1/2}^{n+1/2} = R_1(U, U_{i+1}).$$

The function $R_1(U, U_{i+1})$ is a solution of a Riemann problem specified by the two states U and U_{i+1}. The

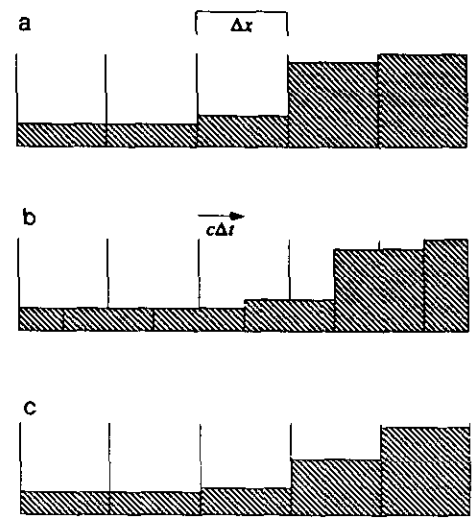

FIG. 1. Geometrical picture of a single time step of the method of Courant, Issacson, and Rees [2].

method of Courant, Issacson, and Rees can be derived from Godunov's method by substituting a linear flux function ($F(u) = cu$) and noting the solution of the Riemann problem is the upstream state. The upstream state for a linear flux function with positive coefficient ($c > 0$) is the left state. Fourier analysis of the linearized system yields CFL numbers constrained not to exceed unity. The CFL numbers would use the characteristic speeds of the gradient of the flux function instead of an advection velocity.

Godunov used this method for the solution of the Lagrangian equations of compressible fluid flow. Both of these methods, as mentioned in the Introduction, are first order. Experience has shown these methods to be nearly impractical for computational purposes. Unfortunately, the order of these upwind methods was not increased until much later. Instead, artificial viscosity methods of one type or another were used to stabilize oscillations in central difference methods around the discontinuities [4].

Van Leer, in his series of papers, arrived at a second-order scheme that is a generalization of the method of Courant, Issacson, and Rees. The idea behind his scheme is to use linear profiles, instead of piecewise constant profiles, for the mass densities discussed in the first-order linear scheme above. The initial profile which is piecewise constant in Fig. 2a. In Fig. 2b, a piecewise linear profile is derived using central differences. Figure 2c shows the slopes being limited so that no new maxima or minima are formed. As in the first-order advection scheme above, the profile is then translated (Fig. 2d) and the mass densities are summed (Fig. 2e) to obtain the piecewise constant profile at the next time step.

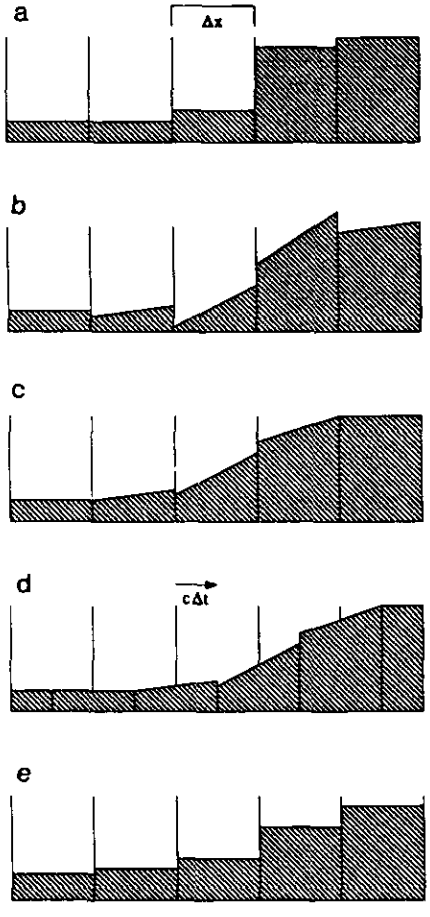

FIG. 2. Geometrical picture of a single time step of the method of van Leer.

An algebraic description of van Leer's algorithm is

$$\frac{u^{n+1}-u}{\Delta t}+c\frac{\Delta_i u^{n+1/2}_{i+1/2}}{\Delta x}=0,$$

where

$$u^{n+1/2}_{i+1/2}=u+\frac{1}{2}\left(1-\frac{c\,\Delta t}{\Delta x}\right)\overline{\Delta}_i u$$

and

$$\overline{\Delta}_i u=\begin{cases}\delta\ \mathrm{sgn}(u_{i+1}-u_{i-1}), & \text{if } \Delta_i u_{i+1}\,\Delta_i u>0\\ 0, & \text{otherwise.}\end{cases}$$

such that

$$\delta=\min(2\,|\Delta_i u_{i+1}|,\ 2\,|\Delta_i u|,\ \tfrac{1}{2}\,|u_{i+1}-u_{i-1}|).$$

A fourth-order estimate of the linear profiles can be used without affecting the monotonicity of the scheme [8]. A new expression for δ may be subsituted for the above expression:

$$\delta=\min(2\,|\Delta_i u_{i+1}|,\ 2\,|\Delta_i u|,\ \tfrac{2}{3}\,|((u_{i+1}-u_{i-1})-\tfrac{1}{4}(\overline{\Delta}_i u_{i+1}+\overline{\Delta}_i u_{i-1}))|).$$

The corresponding generalization of van Leer's scheme again uses the solution of a Riemann problem:

$$\frac{U^{n+1}-U}{\Delta t}+\frac{\Delta_i F_1(U^{n+1/2}_{i+1/2})}{\Delta x}=0 \tag{3a}$$

$$U^{n+1/2}_{i+1/2}=R_1({}_{(-)}U^{n+1/2}_{i+1/2},\ {}_{(+)}U^{n+1/2}_{i+1/2}) \tag{3b}$$

$${}_{(-)}U^{n+1/2}_{i+1/2}=U+\frac{1}{2}\left(I-\frac{\Delta t}{\Delta x}\frac{\partial F_1}{\partial U}\right)\overline{\Delta}_i U$$

$${}_{(+)}U^{n+1/2}_{i+1/2}=T_i\left(U-\frac{1}{2}\left(I+\frac{\Delta t}{\Delta x}\frac{\partial F_1}{\partial U}\right)\overline{\Delta}_i U\right). \tag{3c}$$

The addition of the central difference term does not affect the stability of the advection scheme or the linearization of (3a). For the hyperbolic systems described above, two- and three-dimensional schemes can be designed using splitting methods [16, 17]. However, the direction taken in

this paper is to develop unsplit methods. Colella, in [8], designed a 2D advection algorithm using a geometric approach that can be called a generalization of the method of Courant, Issacson, and Rees. In Fig. 3 the advection method is derived from how a rectangle is traced back along the characteristic direction of the equation.

By summing up the overlaps of the rectangles and rewriting the result in conservation form, the following-first order scheme is found:

$$u_{\alpha+1/2} = u - \frac{c_{l+1}\lambda_{l+1}}{2}\Delta_\beta u, \qquad c_l > 0, \quad l = 1,2 \tag{4a}$$

$$\frac{u^{n+1}-u}{\Delta t} + \sum_{l=1}^{2} \frac{c_l \Delta_\alpha u_{\alpha+1/2}}{\Delta x_l} = 0. \tag{4b}$$

The scheme is monotone since the solution at the advanced time level is a positive weighted sum of the current level and the sum of the weights is unity. The second part of the scheme is in conservation form and is called the corrector step. The relations described in the first part of the scheme are called the predictor step. From a truncation error analysis, it is necessary to center $u_{\alpha+1/2}$ both in time and space in order to obtain second-order accuracy. To accomplish this, the first part of the difference scheme is augmented by limited central difference approximations to the spatial derivatives. The predictor equation (4a) becomes

$$u_{\alpha+1/2} = u + \frac{1}{2}(1 - c_l\lambda_l)\overline{\Delta_\alpha} u - \frac{c_{l+1}\lambda_{l+1}}{2}\Delta_\beta u, \qquad l = 1, 2. \tag{4b'}$$

The difference scheme is dependent upon the signs of the coefficients of the advection equation. This dependence is exhibited primarily in the predictor equations where one-sided differences are used. The corrector equation, although symmetric in structure really is using upwind or one-sided differences as well. The monotonicity is destroyed by the addition of the central difference term, even using van Leer limiting. A stronger form of limiting is required to preserve monotonicity [18]. The appropriate generalization for systems is the scheme

$$\frac{U^{n+1}-U}{\Delta t} + \sum_{l=1}^{2} \frac{\Delta_\alpha F_l(U^{n+1/2}_{\alpha+1/2})}{\Delta x_l} = 0, \tag{5a}$$

where for $l = 1, 2$

$$U^{n+1/2}_{\alpha+1/2} = R_l({}_{(-)}U_{\alpha+1/2}, {}_{(+)}U_{\alpha+1/2}) \tag{5b}$$

$${}_{(-)}U_{\alpha+1/2} = U + \frac{1}{2}\left(I - \lambda_l \frac{\partial F_l}{\partial U}\right)\overline{\Delta_\alpha} U - \frac{1}{2}\lambda_{l+1}\Delta_\beta F_{l+1}(U_{\beta+1/2}) \tag{5c}$$

$${}_{(+)}U_{\alpha+1/2} = T_\alpha\left(U - \frac{1}{2}\left(I + \lambda_l \frac{\partial F_l}{\partial U}\right)\overline{\Delta_\alpha} U - \frac{1}{2}\lambda_{l+1}\Delta_\beta F_{l+1}(U_{\beta+1/2})\right) \tag{5d}$$

$$U_{\alpha+1/2} = R_l(U, U_{\alpha+1}). \tag{5e}$$

Using $F_l = c_l u$ and solving the appropriate 1D Riemann problems for the advection equation, the equations reduce to the difference equations for the advection equation. The directions of the one-sided differences correspond to the solutions of the 1D Riemann problems. This observation is what is actually used in generalizing the advection scheme to one for a hyperbolic system. By comparing the linear advection scheme (4) with the nonlinear scheme (5), one sees how fluxes can be generated which are influenced by cells diagonal to the central cell. The specific terms in (5c), (5d) that add influence from the diagonal cells are the last subexpressions. These terms are sometimes called the transverse fluxes.

The corresponding unsplit difference scheme for the 3D advection equation is derived by tracing a parallelepiped back along the characteristics. The volumes of intersection can be found, as in the 2D case, and a first-order scheme can be written,

$$\frac{u^{n+1}-u}{\Delta t} + \sum_{l=1}^{3} \frac{c_l \Delta_\alpha u_{\alpha+1/2}}{\Delta x_l} = 0 \tag{6a}$$

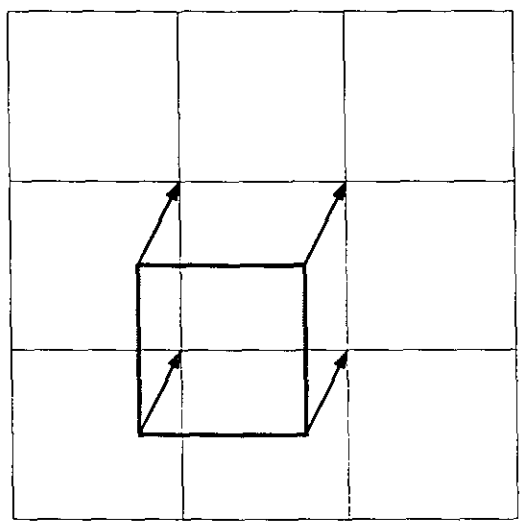

FIG. 3. The rectangle traced back using the characteristics of the advection equation.

where for $l = 1, 2, 3$ and $c_l > 0$

$$u_{\alpha+1/2} = u - \frac{c_{l+1}\lambda_{l+1}}{2}\Delta_\beta({}_\alpha u_{\beta+1/2}) - \frac{c_{l+2}\lambda_{l+2}}{2}\Delta_\gamma({}_\alpha u_{\gamma+1/2}) \tag{6b}$$

$${}_\alpha u_{\beta+1/2} = u - \frac{c_{l+2}\lambda_{l+2}}{3}\Delta_\gamma u \tag{6c}$$

$${}_\alpha u_{\gamma+1/2} = u - \frac{c_{l+1}\lambda_{l+1}}{3}\Delta_\beta u. \tag{6d}$$

There are three steps to this scheme; as data need to be propagated a larger number of steps in the computational lattice to affect the cell being updated. If the CFL conditions

$$c_l\lambda_l \leqslant 1, \qquad l = 1, 2, 3, \tag{7}$$

are met, the above scheme is monotone using the same kind of maximum principle argument as in the 2D case. The corresponding second-order correction of the predictor term (6b) is

$$u_{\alpha+1/2} = u + \frac{1}{2}(1 - c_l\lambda_l)\overline{\Delta}_\alpha u - \frac{c_{l+1}\lambda_{l+1}}{2}\Delta_\beta({}_\alpha u_{\beta+1/2}) - \frac{c_{l+2}\lambda_{l+2}}{2}\Delta_\gamma({}_\alpha u_{\gamma+1/2}), \quad l = 1, 2, 3. \tag{6b'}$$

The appropriate generalization for systems is

$$\frac{U^{n+1} - U}{\Delta t} + \sum_{l=1}^{3}\frac{\Delta_\alpha F_l(U^{n+1/2}_{\alpha+1/2})}{\Delta x_l} = 0,$$

where $l = 1, 2, 3$

$$U^{n+1/2}_{\alpha+1/2} = R_l({}_{(-)}U_{\alpha+1/2}, {}_{(+)}U_{\alpha+1/2})$$

$${}_{(-)}U_{\alpha+1/2} = U + \frac{1}{2}\left(I - \lambda_l\frac{\partial F_l}{\partial U}\right)\overline{\Delta}_\alpha U - \frac{1}{2}\lambda_{l+1}\Delta_\beta F_{l+1}({}_\alpha U_{\beta+1/2}) - \frac{1}{2}\lambda_{l+2}\Delta_\gamma F_{l+2}({}_\alpha U_{\gamma+1/2})$$

$${}_{(+)}U_{\alpha+1/2} = T_\alpha\left(U - \frac{1}{2}\left(I + \lambda_l\frac{\partial F_l}{\partial U}\right)\overline{\Delta}_\alpha U - \frac{1}{2}\lambda_{l+1}\Delta_\beta F_{l+1}({}_\alpha U_{\beta+1/2}) - \frac{1}{2}\lambda_{l+2}\Delta_\gamma F_{l+2}({}_\alpha U_{\gamma+1/2})\right)$$

$${}_\alpha U_{\beta+1/2} = R_l({}_{(-)\alpha}U_{\beta+1/2}, {}_{(+)\alpha}U_{\beta+1/2})$$

$${}_\alpha U_{\gamma+1/2} = R_l({}_{(-)\alpha}U_{\gamma+1/2}, {}_{(+)\alpha}U_{\gamma+1/2})$$

$${}_{(-)\alpha}U_{\beta+1/2} = U - \frac{\lambda_{l+2}}{3}\Delta_\gamma F_{l+2}(U_{\gamma+1/2})$$

$${}_{(+)\alpha}U_{\beta+1/2} = T_\alpha({}_{(-)\alpha}U_{\beta+1/2})$$

$${}_{(-)\alpha}U_{\gamma+1/2} = U - \frac{\lambda_{l+1}}{3}\Delta_\beta F_{l+1}(U_{\beta+1/2})$$

$${}_{(+)\alpha}U_{\gamma+1/2} = T_\alpha({}_{(-)\alpha}U_{\gamma+1/2})$$

$$U_{\alpha+1/2} = R_l(U, U_{\alpha+1}).$$

As in the 2D case, the linear difference scheme was written in a form suggestive of the nonlinear case. Instead of having one-sided differences, solutions of Riemann problems are used to resolve which direction the various discrete solution increments are propagating. The transverse flux structure is even more complicated in three dimensions. The larger amount of data available to influence the central cell implies a larger number of Riemann solutions to resolve which contributions are valid. The number of Riemann problems to resolve is four solutions per face per time step or 12 solutions per cell per time step. It is tempting to call these schemes remapping algorithms because of their geometric origin. In fact they can be used for this purpose. However, remapping algorithms cannot be directly used to solve nonlinear systems because of their dependence on a single characteristic speed.

The discrete Fourier analysis of the linearization of the above system yields CFL numbers contrained not to exceed one. The CFL numbers would use the characteristic speeds of the gradient of the flux function instead of the three advection velocities. The following section describes the appropriate time step restrictions for the equations of gas dynamics.

IMPLEMENTATION

To illustrate the algorithm, it is implemented using the equations of inviscid compressible gas dynamics. The solution vector for this set of equations is

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$$

where ρ is the fluid density, (u, v, w) is the cartesian velocity of the fluid, and E is the total fluid energy per unit mass. The flux functions are:

$$F_1=\begin{pmatrix}\rho u\\ \rho uu+p\\ \rho uv\\ \rho uw\\ \rho uE+pu\end{pmatrix},\qquad F_2=\begin{pmatrix}\rho v\\ \rho vu\\ \rho vv+p\\ \rho vw\\ \rho vE+pv\end{pmatrix},$$

$$F_3=\begin{pmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho ww+p\\ \rho wE+pw\end{pmatrix}.$$

The variable p is the scalar pressure. The introduction of p leads to more unknowns than equations. An additional closure relation called the equation of state is introduced:

$$p=f(\rho, E-|u|^2/2).$$

For the examples in the next section, the ideal equation of state is used:

$$p=(\gamma-1)\rho(E-|u|^2/2).$$

The boundary conditions modeled are inflow, reflecting, and outflow conditions. Inflow and outflow conditions are analogous to Dirichlet and Neumann boundary conditions. How these conditions are differenced is outlined below. Neglecting the boundary conditions, for the moment, and assuming the availability of a Riemann solver, the difference scheme can be used as outlined in the previous section. However, some problem-dependent modifications are made that show discernible improvement. None of these changes are new and they are used in one and two dimensions. The modifications include a small amount of artificial viscosity, slope flattening, and characteristic projections. These items are described below.

The first modification to the basic scheme is the addition of a small amount of artificial viscosity ($\sim\frac{1}{10}$ the nominal value). Its purpose is to obtain additional stabilization of oscillations behind strong shocks. If a similar amount of viscosity is used with a standard scheme, it would be inconsequential. The structure of the dissipative term is a simple scalar viscosity based on the 1D viscosity of Lapidus [19]. Note also that the mass is diffused along with the total energy and momentum in contrast to what is done in the method of von Neumann and Richtmyer [4]. The corrector step of the scheme is modified as

$$\frac{U^{n+1}-U}{\Delta t}+\sum_{l=1}^{3}\frac{\Delta_\alpha F_l(U^{n+1/2}_{\alpha+1/2})}{\Delta x_l}=\sum_{l=1}^{3}\Delta_\alpha C_{\alpha+1/2}\,\Delta_\alpha U_{\alpha+1}.$$

The coefficient $C_{\alpha+1/2}$ is a face-centered quantity that is an average of coefficients that are generated at vertices:

$$C_{i+1/2}=-q\min\left(0,\frac{1}{4}\sum_{\substack{j'=-1,0\\k'=-1,0}}C_{i+1/2,j+1/2+j',k+1/2+k'}\right)$$

$$C_{j+1/2}=-q\min\left(0,\frac{1}{4}\sum_{\substack{i'=-1,0\\k'=-1,0}}C_{i+1/2+i',j+1/2,k+1/2+k'}\right)$$

$$C_{k+1/2}=-q\min\left(0,\frac{1}{4}\sum_{\substack{i'=-1,0\\j'=-1,0}}C_{i+1/2+i',j+1/2+j',k+1/2}\right).$$

The recommended value of the parameter $q=0.1$. Finally the vertex terms are a simple approximation to the divergence of the fluid:

$$C_{i+1/2,j+1/2,k+1/2}=\frac{1}{4}\left[\frac{\Delta_i}{\Delta x}(u+u_{i+1}+u_{k+1}+u_{i+1,k+1})\right.$$
$$+\frac{\Delta_j}{\Delta y}(v+v_{i+1}+v_{k+1}+v_{i+1,k+1})$$
$$\left.+\frac{\Delta_k}{\Delta y}(w+w_{i+1}+w_{j+1}+w_{i+1,j+1})\right].$$

Slope flattening is another dissipation mechanism for reducing oscillations behind a strong shock. Flattening locally reduces the difference scheme to first order around a shock. The method used here is the technique described in [8]. First define a linear function

$$\eta(z)=1-\frac{z-z_0}{z_1-z_0}.$$

Next limit the range of the function between 0 and 1:

$$\bar{\eta}(z)=\min(1,\max(0,\eta(z))).$$

Define the following cell centered variable:

$$\bar{\chi}^u=\begin{cases}\bar{\eta}\left(\dfrac{|p_{i+1}-p_{i-1}|}{|p_{i+2}-p_{i-2}|}\right),\\ \qquad \text{if}\quad u_{i-1}-u_{i+1}>0,\quad \dfrac{|p_{i+1}-p_{i-1}|}{\min(p_{i+1},p_{i-1})}>\delta\\ 1,\qquad \text{otherwise.}\end{cases}$$

Similarly variables $\bar{\chi}^v$ and $\bar{\chi}^w$ may then be defined by successively substituting j and k for i and v and w for u.

Corresponding to $\tilde{\chi}^u$ is

$$\chi^u = \min(\tilde{\chi}^u_{i-s}, \tilde{\chi}^u), \qquad \text{where} \quad s = \operatorname{sign}(p_{i+1} - p_{i-1}).$$

Similarly, variables χ^v and χ^w may be defined by, again, successively substituting j and k for i and v and w for u.

Finally let

$$\chi = \min(\chi^u, \chi^v, \chi^w).$$

Then the slopes are modified as

$$\overline{\Delta}_\alpha \Rightarrow \chi \overline{\Delta}_\alpha, \qquad l = 1, 2, 3.$$

The recommended values of the parameters are $\delta = 0.33$, $z_0 = 0.75$, and $z_1 = 0.85$.

The last modification to the difference scheme is a twofold change in the central difference part of the predictor step:

$$({}_-)U_{\alpha+1/2} = U + \frac{1}{2}\left(I - \lambda_l \frac{\partial F_l}{\partial U}\right)\overline{\Delta}_\alpha U - \cdots, \qquad l = 1, 2, 3$$

$$({}_+)U_{\alpha+1/2} = T_\alpha\left(U - \frac{1}{2}\left(I + \lambda_l \frac{\partial F_l}{\partial U}\right)\right)\overline{\Delta}_\alpha U - \cdots, \qquad l = 1, 2, 3.$$

First the increment of the left and right states from the central difference terms is calculated using the primitive flow variables. Second, the increments are further filtered so that jumps propagating away from a given cell edge do not contribute to the increment at the cell edge. This is redundant for linear systems but makes a noticeable difference for this nonlinear system. The modifications of the left state calculation is written out. The corresponding modification to the right state is similar and consequently omitted.

It is found that using the primitive variables in the central difference increment calculation leads to less error in the pressure gradients than using conserved variables. The primitive variable formulation is of the form

$$V = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix}$$

such that

$$\frac{\partial V}{\partial t} + \sum_{l=1}^{3} A_l \frac{\partial V}{\partial x_l} = 0$$

and

$$A_1 = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 1/\rho \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & 0 & u \end{pmatrix}, \qquad A_2 = \begin{pmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 1/\rho \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho c^2 & 0 & v \end{pmatrix},$$

$$A_3 = \begin{pmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 1/\rho \\ 0 & 0 & 0 & \rho c^2 & w \end{pmatrix},$$

where

$$c^2 = \frac{\partial p}{\partial \rho} - \frac{p}{\rho^2}\frac{\partial p}{\partial e}$$

and

$$e = E - \frac{|u|^2}{2}.$$

The mapping for physical values of gas dynamics is one-to-one and onto, between the conserved and primitive variables for a reasonable equation of state. To calculate the increment from the central difference part of the scheme, transform that part to primitive variables:

$$({}_-)V_{\alpha+1/2} = V + \tfrac{1}{2}(I - \lambda_l A_l)\,\overline{\Delta}_\alpha V.$$

Once having obtained $({}_-)V_{\alpha+1/2}$, then transform back to conservation variables. The above fragment is further modified by a characteristic projection technique due to Harten [20]. First an arbitrary reference state may be added as only the jumps in the solution are used:

$$({}_-)V_{\alpha+1/2} - V_{\text{ref}} = V - V_{\text{ref}} + \tfrac{1}{2}(I - \lambda_l A_l)\,\overline{\Delta}_\alpha V. \qquad (8)$$

Let

$$e^l_1, e^l_2, \ldots, e^l_5$$

be the five eigenvalues of A_l and

$$r^l_1, r^l_2, \ldots, r^l_5; \qquad l^l_1, l^l_2, \ldots, l^l_5$$

be the associated right and left eigenvectors. The eigenvectors are assumed to be scaled such that they are

biorthonormal. For $l = 1, 2, 3$ define a projection operator on a vector w,

$$P^l w = \tfrac{1}{2} \sum_{m=1}^{5} (l^l_m \cdot w)(\operatorname{sign}(e^l_m) + 1)\, r^l_m.$$

P^l is then applied to the right-hand side of (8):

$$\begin{aligned} &P^l(V - V^l_{\text{ref}}) + P^l(\tfrac{1}{2}(I - \lambda_l A_l)\, \overline{\Delta}_\alpha V) \\ &\quad = P^l(V - V^l_{\text{ref}}) + \tfrac{1}{4} \sum_{m=1}^{5} (\operatorname{sign}(e^l_m) + 1) \\ &\qquad \times (1 - \lambda_l e^l_m)(l^l_m \cdot \overline{\Delta}_\alpha V)\, r^l_m. \end{aligned}$$

A reference state is chosen to simplify the projection as much as possible:

$$V^l_{\text{ref}} = V + \tfrac{1}{2}(1 - \lambda_l e^l_{\max})\, \overline{\Delta}_\alpha V.$$

Here $e^l_{\max}$ is the maximum value of all the eigenvalues. Using this particular reference state yields the following increment in primitive variables:

$$\begin{aligned} {}_{(-)}V_{\alpha+1/2} = V^l_{\text{ref}} + \frac{\lambda_l}{4} \sum_{m=1}^{5} &(e^l_{\max} - e^l_m) \\ &\times (\operatorname{sign}(e^l_m) + 1)(l^l_m \cdot \overline{\Delta}_\alpha V)\, r^l_m. \end{aligned}$$

For the right state, the projection operator is

$$\tilde{P}^l w = \tfrac{1}{2} \sum_{m=1}^{5} (l^l_m \cdot w)(1 - \operatorname{sign}(e^l_m))\, r^l_m.$$

Similarly, the right state reference value is

$$\tilde{V}^l_{\text{ref}} = V - \tfrac{1}{2}(1 + \lambda_l e^l_{\min})\, \overline{\Delta}_\alpha V$$

and

$$\begin{aligned} {}_{(+)}V_{\alpha+1/2} = T_\alpha \Bigg(\tilde{V}^l_{\text{ref}} - \frac{\lambda_l}{4} \sum_{m=1}^{5} &(e^l_m - e^l_{\min}) \\ &\times (1 - \operatorname{sign}(e^l_m))(l^l_m \cdot \overline{\Delta}_\alpha V)\, r^l_m \Bigg). \end{aligned}$$

Here $e^l_{\min}$ is the minimum value of all the eigenvalues.

The time step restrictions used in this implementation, following from the discrete Fourier analysis mentioned above, are

$$\max_{1 \leqslant l \leqslant 3,\, 1 \leqslant m \leqslant 5} (|e^l_m|\, \lambda_l) \leqslant 1.$$

Because of the nonlinearity of the equations, a CFL number less than one is used. All the test problems described below use a CFL number of 0.8.

An operational view in formulating boundary conditions is used. An operational point of view means that the difference scheme is divided into small parts and simple boundary conditions are applied to each part. The advantage to the approach is simplicity. By partitioning the difference scheme into small enough pieces, one-dimensional boundary conditions can be applied. In addition, the stencil (support) of these boundary conditions is smaller. The disadvantage is that boundary conditions need to be applied a larger number of times than alternative global approaches.

The types of boundary conditions outlined in this paper are reflecting, inflow, and outflow conditions. Inflow conditions are much like Dirichlet conditions and are generally set by making boundary or so-called ghost cells equal to the inflow values. Reflecting boundary conditions are physically akin to having a wall or solid impediment block the flow. This condition is more complicated than inflow but it can be handled by appealing to symmetry and the physical boundary conditions of the compressible equations at boundaries (called slip boundaries). The last boundary condition, outflow, tends to be the most problematical. If the outflow is supersonic or nearly constant, a Neumann-like boundary condition is used. That is a finite difference approximation of the normal derivative of the flow variables set to zero. For subsonic flows with some variation more complicated absorbing boundary conditions must be used. Only simple outflow conditions are discussed. All three types of boundary conditions share the property that cells just outside the computation region must be set to implement these conditions.

The implication of the above discussion is that boundary conditions must be set for the Riemann solver steps, for the predictor step where slopes are determined, the slope flattening algorithm, and the artificial viscosity.

The Riemann solver step boundary conditions are easiest to implement as they are quite local. Assume that a boundary is at the index $i_b + \frac{1}{2}$ and that the computational region is locally to the right of $i_b + \frac{1}{2}$ ($i \geqslant i_b + \frac{1}{2}$). Then for the approximate solution of the Riemann problem

$$U_{i_b+1/2} = R(U_{i_b+1}, U_{i_b}),$$

a value for U_{i_b} must be supplied. The value for U_{i_b} for inflow boundary conditions is

$$U_{i_b} = U_{\text{inflow}}.$$

The outflow boundary condition is Neumann-like:

$$U_{i_b} = U_{i_b+1}.$$

The reflecting boundary condition has the form

$$U_{i_b}=\begin{pmatrix}\rho_{i_b+1}\\ -\rho_{i_b+1}u_{i_b+1}\\ \rho_{i_b+1}v_{i_b+1}\\ \rho_{i_b+1}w_{i_b+1}\\ \rho_{i_b+1}E_{i_b+1}\end{pmatrix}.$$

Boundary conditions in the y and z direction are easily deduced from the above relations.

The slopes in the predictor step reach over two cells for the fourth-order approximation and one cell for the second-order approximation. Assuming, once again, that the boundary is at the index $i_b+\frac{1}{2}$ and that the computational region is locally to the right of this boundary, then the values of U must be supplied at the indices i_b and i_b-1. For inflow with fourth-order slopes this implies that

$$V_{i_b}=V_{i_b-1}=V_{\text{inflow}}.$$

Outflow conditions are of the form:

$$V_{i_b}=V_{i_b-1}=V_{i_b+1}.$$

Second-order slopes only require the value at V_{i_b}. Reflecting boundary conditions require more detail. It is observed that normal velocities at cell edges of reflecting boundaries should be as close as possible to zero in order not to cause spurious waves. Tangential velocities, density, and pressure (remember, primitive variables are being used) are reflected in the usual manner. To derive values for boundary ghost cells for the slopes, two conditions are imposed:

$$0=u_{i_b+1}-\tfrac{1}{2}\overline{\Delta}_i V_{i_b+1}$$

and

$$u_{i_b-1}=-u_{i_b+2}.$$

The fourth-order unlimited slope formula can be used to determine the values of

$$u_{i_b}=-3u_{i_b+1}-u_{i_b+2}-\tfrac{1}{8}(u_{i_b+3}+u_{i_b+2}).$$

To summarize, the values of the boundary ghost cells are

$$V_{i_b}=\begin{pmatrix}\rho_{i_b+1}\\ -3u_{i_b+1}-u_{i_b+2}-\frac{1}{8}(u_{i_b+3}+u_{i_b+2})\\ v_{i_b+1}\\ w_{i_b+1}\\ p_{i_b+1}\end{pmatrix},$$

$$V_{i_b-1}=\begin{pmatrix}\rho_{i_b+2}\\ -u_{i_b+2}\\ v_{i_b+2}\\ w_{i_b+2}\\ p_{i_b+2}\end{pmatrix}.$$

Second-order slopes only require an obvious modification to V_{i_b} and there are no V_{i_b-1}.

The slope flattening algorithm does not require a very exact treatment of the boundary conditions. Simple reflection is done instead of forcing the normal velocity to zero at the boundary edge. The flattening does reach much further out than any other part of the algorithm. Both inflow and outflow boundary conditions require four ghost cells. For inflow at $i_{b+1/2}$

$$p_{i_b-3}=p_{i_b-2}=p_{i_b-1}=p_{i_b}=p_{\text{inflow}}$$

and

$$u_{i_b-2}=u_{i_b-1}=u_{i_b}=u_{\text{inflow}}.$$

For outflow conditions

$$p_{i_b-3}=p_{i_b-2}=p_{i_b-1}=p_{i_b}=p_{i_b+1}$$

and

$$u_{i_b-2}=u_{i_b-1}=u_{i_b}=u_{i_b+1}.$$

For reflecting conditions

$$p_{i_b-3}=p_{i_b+4},\qquad p_{i_b-2}=p_{i_b+3},$$
$$p_{i_b-1}=p_{i_b+2},\qquad p_{i_b}=p_{i_b+1},$$

and

$$u_{i_b-2}=-u_{i_b+3},\qquad u_{i_b-1}=-u_{i_b+2},\qquad u_{i_b}=-u_{i_b+1}.$$

The artificial viscosity calculation uses the same boundary data as the slope flattening algorithm. What distinguishes the artificial viscosity from the other terms requiring boundary data is the apparent mixing of directions. The Riemann solver, flattening, and slope calculations are essentially one-dimensional. By breaking the artificial viscosity into one-dimensional sweeps this same kind of application of boundary conditions can be preserved. This is especially important at corners in a nonrectangular computational mesh.

TEST PROBLEMS

To convey a qualitative idea of the performance of this scheme, several calculations are done. These calculations are the 1D interacting blast problem described in [7], as well as the 2D Mach reflection problem described in the same reference. Benchmark calculations in three dimensions are still relatively rare. The astrophysical calculations of Arnold [21, 22] are used as comparisons for the three-dimensional case. All of these problems can be run in various orientations in order to test for the preservation of symmetry by the finite difference scheme. Some of the problems have been run using different orientations to verify correct coding, but no further discussion is included in this paper about symmetry preservation other than that the obvious symmetries are preserved. The Riemann solver used in these calculations is an implementation of the algorithm described in [12].

The 1D blast problem is calculated on the unit interval using a ratio of specific heats, γ, of 1.4. The initial density everywhere within the interval is 1.0 and the velocity is everywhere zero. The problem is driven by the variation in the pressure. In the interval (0, 0.1) the pressure is 1000. In the interval (0.9, 1.0) the pressure is 100. The remaining region in the problem has a pressure of 0.01. The boundary conditions are reflecting on both ends of the interval. As mentioned above, the CFL number used in this test problem and the other two problems is 0.8. The results displayed in [7] are at resolutions of 200 and 1200 cells. The results in [7] are also overlayed with a solution from a highly refined run that was labeled the "converged result."

Figures 4 and 5 show density plots of the finish time (0.038) using the unsplit algorithm in 1D. The results are very similar to the MUSCL code results in the review paper. This is not surprising since the algorithms have roughly the same spatial and temporal accuracy and use similar upstream centered differencing. At the resolution of 1200 points, the unsplit solution is converged to the larger features of the solution. Only the left-facing contact discontinuity is noticeably different from the PPM solution. The difference observed is some smearing. Consideration was given to putting in contact-steepening in the unsplit algorithm, but on the advice of Colella [23], it was left out. The reason for this omission is that unphysical instabilities are observed at contact surfaces in higher dimensional flows and are traceable to the contact-steepening algorithm.

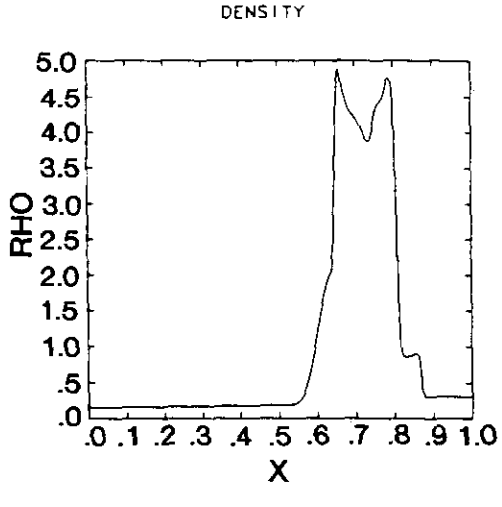

FIG. 4. Blast problem using 200 cells.

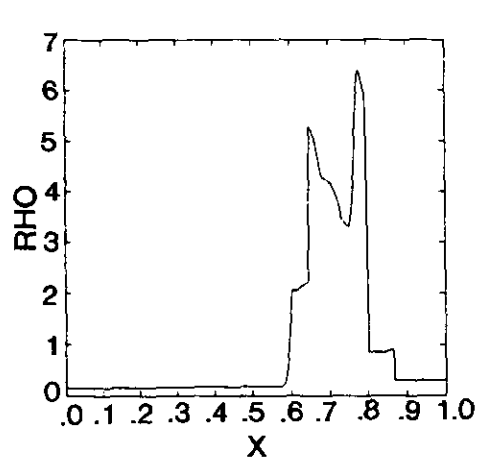

FIG. 5. Blast problem using 1200 cells.

It seems almost paradoxical that the right-facing contact-discontinuity is not also smeared, as it is a stronger wave than the left-facing contact. The reason for the steeper right contact is that it has a smaller lifetime than the left. The rate of diffusion of the linearly degenerate wave is proportional to the number of time steps through which it is propagated.

The 2D Mach reflection is a more complicated test problem. This simulation is performed in a rectangular region of length 4.0 and height 1.0. The boundary conditions are set to simulate a Mach 10 shock in a shock tube hitting a ramp with an angle of 30° from the horizontal. To fit this problem into a rectangular box, the Mach 10 shock is initialized with an angle of 60° from the x axis. The shock touches the x axis at the point $\frac{1}{6}$. The ratio of specific heats is again 1.4. The preshocked values of density, velocity, and pressure are 1.4, 0.0, and 1.0, respectively. Using the equations in Sections 67 and 68 of Courant and Friedrichs [24], the one-dimensional postshock values of a Mach 10 shock are 8.0, 8.25, and 116.5 for the density, velocity, and pressure, respectively. The one-dimensional velocity must be broken up into components parallel and perpendicular to the x axis.

The bottom of the ramp is simulated with a reflecting boundary along the x axis in the z direction. However, the boundary only extends over the interval $(\frac{1}{6}, 4)$. On the interval $(0, \frac{1}{6})$ an inflow condition with the postshock values of

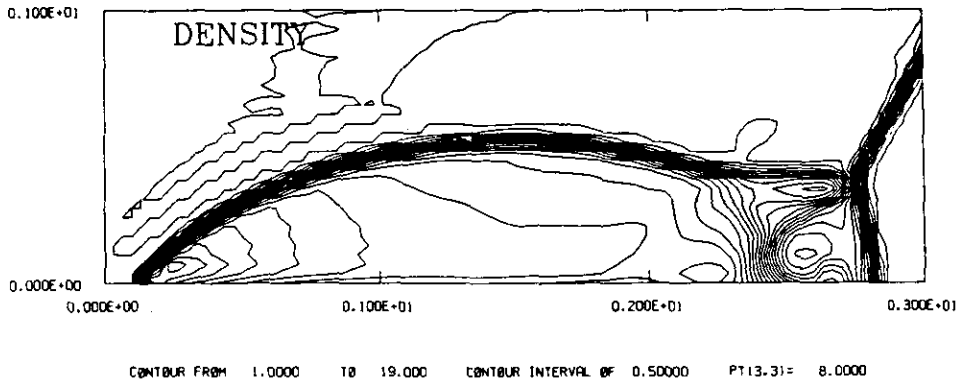

FIG. 6. Double Mach reflection problem on a 30×120 grid.

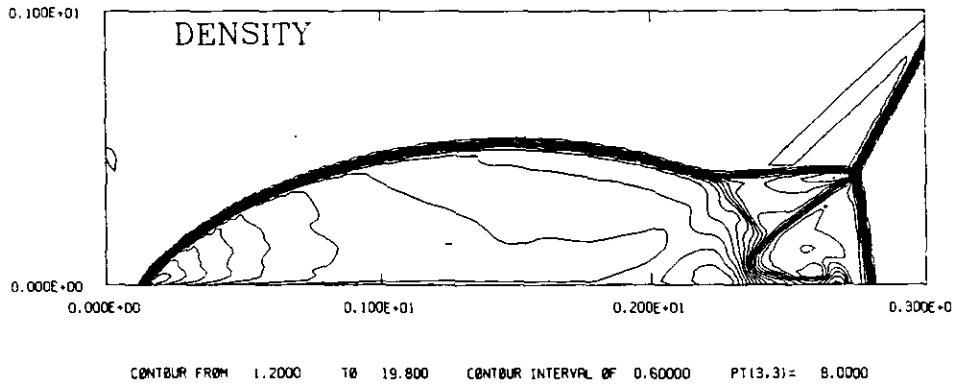

FIG. 7. Double Mach reflection problem on a 60×240 grid.

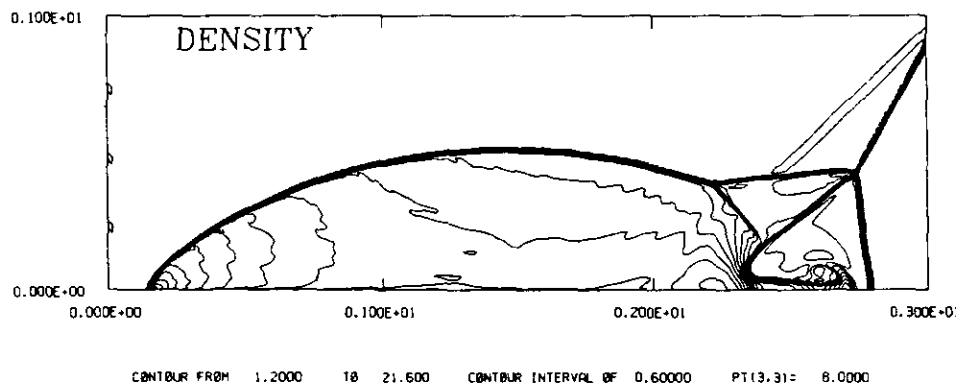

FIG. 8. Double Mach reflection problem on a 120×480 grid.

the shock is maintained. To complete the description of the boundary conditions at the bottom a reflecting boundary in the x direction is set at $\frac{1}{6}$. These conditions act as a "knife edge" that attaches the flow to this point. The boundary conditions at the right edge of the box are set to the preshocked values of the fluid while the boundary conditions on the left side of the box are set to the shocked values of the fluid. The top boundary condition is time dependent. The movement of the shock is tracked by the analytic values derived above and the preshock and postshock values are inserted before and after the shock, respectively. Zones through which the shock passes are set to a linear combination of the postshock and preshock values according to the volume fractions.

Figures 6, 7, and 8 show density fields for calculations done on 30×120, 60×240, and 120×480 meshes, respectively. The density results tend to be closer to the PPM results in the review paper [7] than to the MUSCL results. The instabilities pointed out in the MUSCL scheme are not present in these runs. It is much harder to obtain a quantitative measure of accuracy in these tests since a converged result is not available. It is clear that the algorithm does not suffer from the problems encountered using the Godunov, MacCormack, BBC, or FCT algorithms. The Godunov, MacCormack, and BBC methods all suffered from too much numerical dissipation. The FCT algorithm generated very noisy solutions with some structures (the unstable jet) diverging from what appears in the other solutions. The results can also be compared to those found in [8] as the 3D algorithm is identical to Colella's algorithm in 2D.

The three-dimensional problem calculated in this paper is a Mach 6 jet described in Arnold's thesis [21] and published in [22]. The computation region is a parallelepiped with length (x direction) 16.0, and height (z direction) and depth (y direction) 8.0. Initially the region, called the Intergalactic Medium (IGM), is at rest with a unit density and sound speed. At the $y-z$ plane with x coordinate equal to zero a jet of material is injected into the IGM with a density of 0.1 and a local Mach number of 6.0. The beam is in pressure equilibrium with the IGM. The local Mach number is the ratio of the velocity of the beam to the sound speed of the beam. To find the velocity of the beam, the pressure equilibrium is used:

$$u_{\text{beam}} = Mc_{\text{ambient}} \sqrt{\rho_{\text{ambient}}/\rho_{\text{beam}}}.$$

The Mach number is M, sound speed is c, and density is ρ. The subscript "ambient" refers to the material in the IGM while subscript "beam" refers to the inflow material. The size of the opening of the jet into the IGM is a unit square in the positive quadrant of the $y-z$ plane with one corner placed at the origin. Where the beam is not being injected through the $y-z$ plane outflow boundary conditions are imposed. Only the positive octant of the flow is calculated in this example. As a result, symmetry planes are imposed on the $x-y$ and $x-z$ planes passing through the origin. The other boundary planes in the problem can be handled with either outflow or symmetry conditions as signals do not reach them within the running time of the simulation.

Although many features of the jet are stable, the overall problem is unstable enough to inhibit comparisons of convergence or other detailed numerical properties. Figures 9a, 9b, 9c correspond to Figs. 1a, 1b, 1c in [22]. Qualitatively the pictures are similar, but even some of the large features

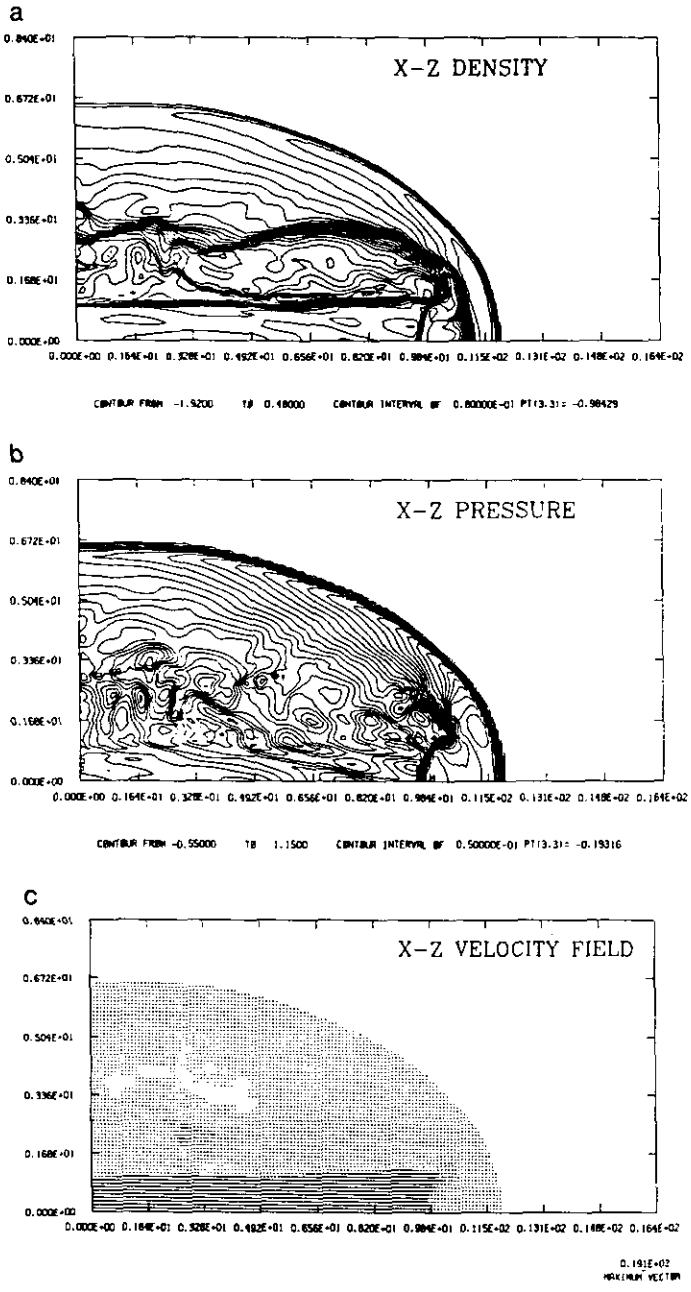

FIG. 9. Axial cross sections at time 2.4 for density, pressure, and the velocity field.

are different. The location of the front of the bow shock is further ahead in Arnold's results. The contact discontinuity is more widely separated from the beam shock in Fig. 9a than in the corresponding results in [22]. The shock structure within the beam is similar to the results in [22].

The cross sections in Figs. 10a–10f correspond to Figs. 2a–2f in [22]. Arnold and Arnett do not describe the location of these cross sections, but in Arnold's thesis they are documented as 8.0 beam radii (Figs. 2a, 2d), 5.33 beam radii (Figs. 2b, 2e), and 2.67 beam radii (Figs. 2c, 2f). A beam radius is normalized to unity. The largest difference between the unsplit method of this paper and the split method of Arnold is the symmetry of the results. The operator split method causes asymmetries in space that

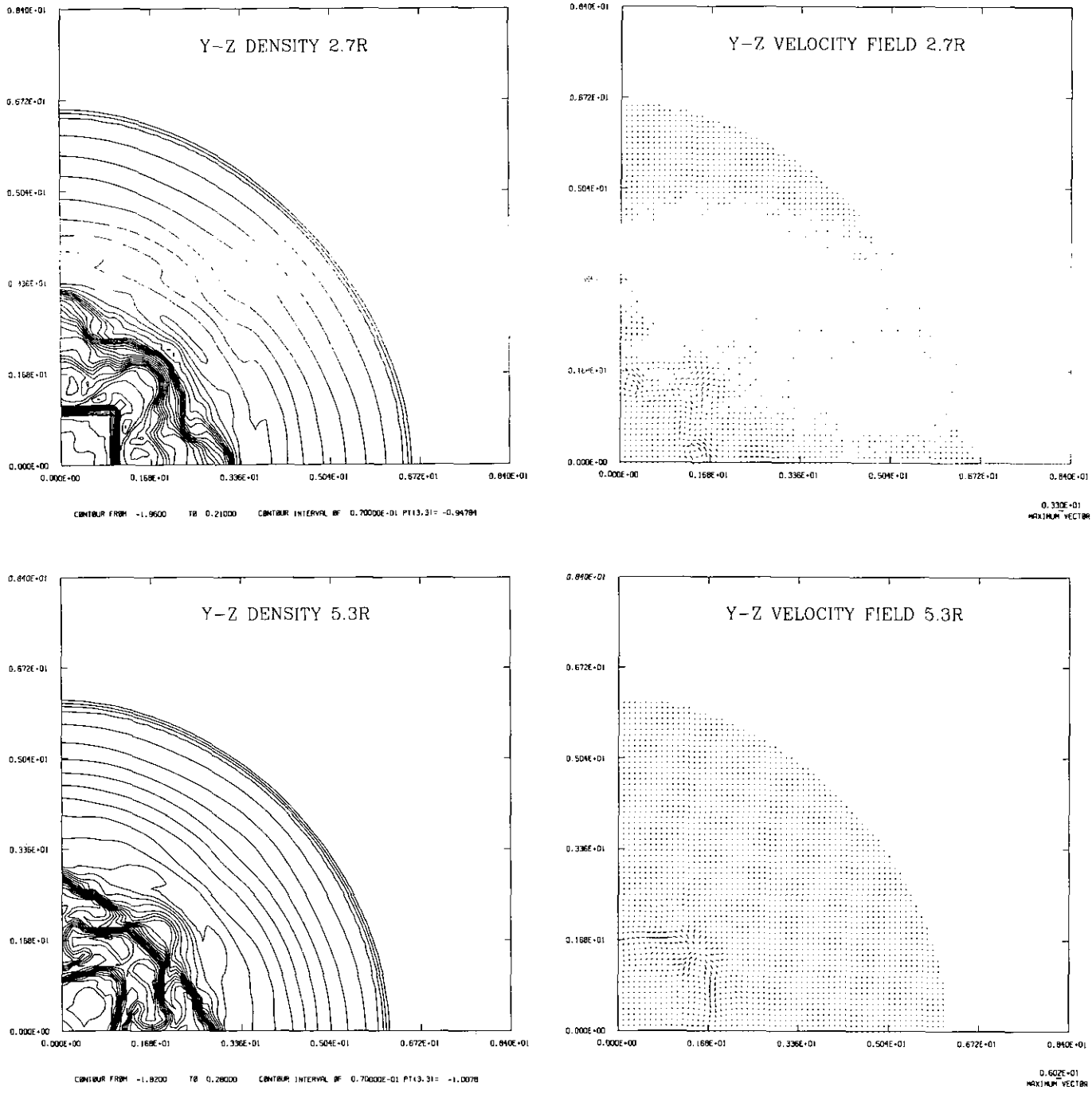

FIG. 10. Axial cross sections at time 2.4 for density, pressure, and the velocity field.

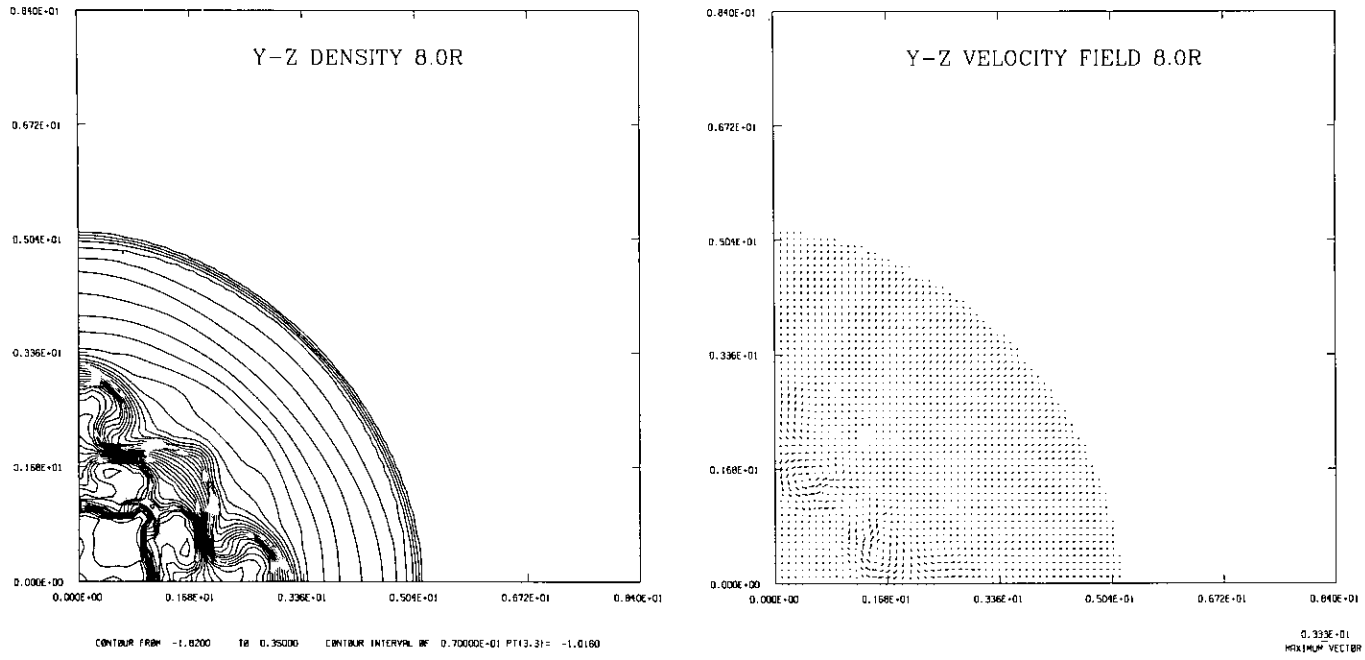

FIGURE 10—*Continued*

cause different instabilities to appear. The obvious effect of this is the shape of the beam. The unsplit method appears to have a much more collimated beam than the split method. The cross sections in Figs. 10a, b, c are much more square than in the corresponding cross sections in Figs. 2a, b, c of [22]. A possible cause for this difference is a larger number of unstable modes being excited by the split method leading to greater beam degradation.

CONCLUSIONS

A three-dimensional unsplit difference scheme is outlined in this paper and tested on several problems. The results of the test problems show the algorithm to be accurate enough to be competitive with other methods. The unsplit scheme also has the advantage of preserving symmetries destroyed by splitting as exhibited by the astrophysical jet example. The disadvantage of the scheme is the number of required Riemann solves per computational cell face. Every face of a computational cell requires four solves which means 12 Riemann solves per cell per time step. However, one cannot conclude that the unsplit method is four times slower than an operator-split method without taking into account their relative accuracies as well as the fraction of time taken in the Riemann solver and the rest of the algorithm. Further work must be done in making these kinds of comparisons. Certainly the method can be made less costly by making the Riemann solves less costly. Work is currently being done in comparing the relative performance of various approximate Riemann solvers mentioned in the text.

ACKNOWLEDGMENTS

This work is funded by U. S. Department of Energy through Los Alamos National Laboratory. The author acknowledges helpful discussions with John Bell and Phil Colella. The author also thanks the tireless and very patient reviewers—one of them being John Bolstad.

REFERENCES

1. P. Lax, A. Harten, and B. van Leer, *SIAM Rev.* **25** (1), 35 (1983).
2. R. Courant, E. Issacson, and M. Rees, *Commun. Pure Appl. Math.* **5**, 243 (1952).
3. S. K. Godunov, *Math. Sb.* **47**, 271 (1959).
4. R. Richtmyer and K. Morton, *Difference Methods for Initial-Value Problems* (Wiley Interscience, New York, 1967), p. 311.
5. B. van Leer, *J. Comput. Phys.* **32**, 101 (1979).
6. P. Colella and P. Woodward, *J. Comput. Phys.* **54**, 174 (1984).
7. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
8. P. Colella, *J. Comput. Phys.* **87**, 171 (1990).
9. B. van Leer, in *Computing Methods in Applied Sciences and Engineering VI*, edited by R. Glowinski and J.-L. Lions (North-Holland, Amsterdam, 1984). p. 493.
10. P. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, CBMS Regional Conference Series in Applied Mathematics, Vol. 11, (SIAM, Philadelphia, 1972).
11. P. Roe, *J. Comput. Phys.* **43**, 357 (1981).

12. P. Colella and H. Glaz, *J. Comput. Phys.* **59**, 264 (1985).
13. S. Osher and F. Solomon, *Math. Comput.* **38**, 339 (1982).
14. J. Bell, P. Colella, and J. Trangenstein, *J. Comput. Phys.* **82**, 362 (1989).
15. J. Dukowicz, *J. Comput. Phys.* **61**, 119 (1985).
16. W. Strang, *SIAM J. Numer. Anal.* **5**, 506 (1968).
17. A. Gourlay and J. Morris, *J. Comput. Phys.* **5**, 229 (1970).
18. J. Saltzman, LAUR-87-2479, Los Alamos National Laboratory, Los Alamos, NM, 1987 (unpublished).
19. A. Lapidus, *J. Comput. Phys.* **2**, 154 (1967).
20. A. Harten, NASA Armes Report NCA2-ORS25-201, NASA Ames Laboratory, Moffett Field, CA (unpublished).
21. C. Arnold, Doctoral thesis, Department of Astronomy, University of Michigan, 1985.
22. C. Arnold and W. Arnett, *Astrophys. J.* **305**, L57 (1986).
23. P. Colella, personal communication, 1989.
24. R. Courant and K. Friedrichs, *Supersonic Flow and Shock Waves* (Springer-Verlag, New York, 1976).