Lawrence Berkeley National Laboratory
Lawrence Berkeley National Laboratory

Title
A cartesian grid embedded boundary method for hyperbolic conservation laws

Permalink
https://escholarship.org/uc/item/6x21g73m

Authors
Colella, Phillip
Graves, Daniel T.
Keen, Benjamin J.
et al.

Publication Date
2004-10-03

Peer reviewed

A Cartesian Grid Embedded Boundary Method for Hyperbolic Conservation Laws

Phillip Colella, [1] Daniel T. Graves, [1] Benjamin J. Keen, [2]
David Modiano, [1]

Lawrence Berkeley National Laboratory, Berkeley, CA [1]

Mathematics Department, University of Michigan, Ann Arbor, MI [2]

Abstract

We present a second-order Godunov algorithm to solve time-dependent hyperbolic systems of conservation laws on irregular domains. Our approach is based on a formally consistent discretization of the conservation laws on a finite-volume grid obtained from intersecting the domain with a Cartesian grid. We address the small-cell stability problem associated with such methods by hybridizing our conservative discretization with a stable, nonconservative discretization at irregular control volumes, and redistributing the difference in the mass increments to nearby cells in a way that preserves stability and local conservation. The resulting method is second-order accurate in L^1 for smooth problems, and is robust in the presence of large-amplitude discontinuities intersecting the irregular boundary.

Key words:

1 Introduction

In this paper, we present an extension of the unsplit method for time-dependent hyperbolic conservation laws in [7,14] to the case of an embedded boundary representation of irregular geometries. Our approach is a generalization of the conservative method in [11], following the ideas in [5,2]. The present method uses the approach in [8] to construct a method that is formally consistent. Near the embedded boundary, the truncation error is $O(h)$, while in the interior, the truncation error is $O(h^2)$, where h is the mesh spacing. This leads to a method for which the solution error is $O(h^2)$ in L^1, and $O(h)$ in L^∞. This differs from the methods cited above, which are formally inconsistent, i.e. have $O(1)$ truncation errors near the embedded boundary. Our method uses a linear hybridization of explicit conservative and nonconservative updates combined with the use of the flux redistribution ideas in [5,2,11] to maintain local conservation. We present results for smooth solutions in two and three space dimensions that verify that the accuracy of the method is consistent with the truncation error and modified equation analysis. We also present results for strong shock problems that demonstrate the robustness and accuracy of the method.

2 Notation

Cartesian grids with embedded boundaries are useful to describe finite-volume representations of solutions to PDE in the presence of irregular boundaries. In figure 1, the grey area represents the region

excluded from the solution domain. The underlying description of space is given by rectangular control volumes on a Cartesian grid $\Upsilon_{\boldsymbol{i}} = [(\boldsymbol{i} - \frac{1}{2}\boldsymbol{v})h, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{v})h], \boldsymbol{i} \in \mathbb{Z}^D$, where D is the dimensionality of the problem, h is the mesh spacing, and $\boldsymbol{v}$ is the vector whose entries are all one. Given an irregular domain Ω, we obtain control volumes $V_{\boldsymbol{i}} = \Upsilon_{\boldsymbol{i}} \bigcap \Omega$ and faces $A_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}$ which are the intersection of the boundary of $\partial V_{\boldsymbol{i}}$ with the coordinate planes $\{\boldsymbol{x} : x_d = (i_d \pm \frac{1}{2})h\}$. We also define $A_{\boldsymbol{i}}^B$ to be the intersection of the boundary of the irregular domain with the Cartesian control volume: $A_{\boldsymbol{i}}^B = \partial\Omega \bigcap \Upsilon_{\boldsymbol{i}}$. For ease of exposition, we will assume here that there is only one control volume per Cartesian cell. The algorithm described here has been generalized to allow for boundaries whose width is less that the mesh spacing.

We recognize three kinds of cells and faces: *regular*, *irregular* and *covered*. Regular cells are cells within the solution domain away from the embedded boundary. Covered cells are completely covered by the boundary and are not part of the solution domain. Irregular cells are cells cut by the embedded boundary. State variables are defined at
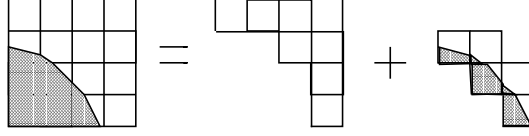
Fig. 1. Decomposition of the grid into regular, irregular, and covered cells. The gray regions are outside the solution domain.

the geometric centers of regular grid cells, even if a cell is irregular. The classification for faces is similar. In the most general case, a face is defined as a pair of control volumes in adjacent cells. In order to avoid the notational complications this would entail, we will assume that, if a face is covered, then at least one of the cells adjacent to the face is covered. This is condition is equivalent to assuming a minimum thickness to the covered regions.

To construct finite-difference methods using this description, we will need several quantities derived from these geometric objects.

- Volume fractions κ and area fraction α:

$$\kappa_{\boldsymbol{i}} = \frac{|V_{\boldsymbol{i}}|}{h^d} \quad , \quad \alpha_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_s} = \frac{A_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}_s}}{h^{(d-1)}}$$

- The centroids of the faces and of $A_{\boldsymbol{i}}^B$; and $\mathbf{n}$, the average of outward normal of $\partial\Omega$ over $A_{\boldsymbol{i}}^B$.

$$\boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{|A_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}|}\int_{A_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}} \boldsymbol{x}dA - (\boldsymbol{i}+\tfrac{1}{2}\boldsymbol{e}^d)h$$

$$\boldsymbol{x}_{\boldsymbol{i}}^B = \frac{1}{|A_{\boldsymbol{i}}^B|}\int_{A_{\boldsymbol{i}}^B} \boldsymbol{x}dA - \boldsymbol{i}h$$

$$\boldsymbol{n}_{\boldsymbol{i}} = \frac{1}{|A_{\boldsymbol{i}}^B|}\int_{A_{\boldsymbol{i}}^B} \boldsymbol{n}dA$$

We assume we can compute all derived quantities to $O(h^2)$. With just these geometric descriptors, we can define a conservative discretization of the divergence operator. Let $\vec{F} = (F^1...F^d)$ be a function of $\boldsymbol{x}$. Then

$$\nabla \cdot \vec{F} \approx \frac{1}{|V_{\boldsymbol{i}}|}\int_{V_{\boldsymbol{i}}} \vec{F}dV = \frac{1}{|V_{\boldsymbol{i}}|}\int_{\partial V_{\boldsymbol{i}}} \vec{F} \cdot \boldsymbol{n}dA$$

3

$$= \frac{1}{\kappa_i h} \left(\sum_{d=0}^{\boldsymbol{D}-1} \sum_{\pm=+,-} \pm \alpha_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} F^d(x_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) + \alpha_i^B \boldsymbol{n_i} \cdot \vec{F}(\boldsymbol{x_i^B}) \right) + O\left(\frac{h}{\kappa_i}\right) \qquad (1)$$

where (1) is obtained by replacing the normal components of the vector field $\vec{F}$ with the values at the centroids.

3 Stable Evolution of Hyperbolic Conservation Laws

We want to solve a hyperbolic system of conservation laws.

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} = 0 \tag{2}$$

$$U = U(\boldsymbol{x}, t), \ \boldsymbol{x} \in \Omega \subset \mathbb{R}^D \tag{3}$$

$$(F^1 ... F^D) = \vec{F} = \vec{F}(U) \tag{4}$$

$$U, F^d \in \mathbb{R}^m. \tag{5}$$

While the algorithm we describe here applies to general systems of hyperbolic conservation laws, we will show results for the case of polytropic gas dynamics in two and three dimensions.

$$U = (\rho, \rho\boldsymbol{u}, \rho E)^T \tag{6}$$

$$F^d(U) = (\rho u^d, \rho u^d \boldsymbol{u} + p\boldsymbol{e}^d, \rho u^d E + u^d p) \tag{7}$$

Here ρ is the fluid density, $(u^1 ... u^D) = \boldsymbol{u} \in \mathbb{R}^D$ is the velocity, $E \equiv \frac{p}{(\gamma-1)\rho} + |\boldsymbol{u}|^2$ the total energy per unit mass, and p the pressure and $\gamma > 1$ the ratio of specific heats.

We discretize the solution to (5) in space and time, approximating U by values at Cartesian cell centers: $U_{\boldsymbol{i}}^n \approx U(\boldsymbol{i}h, n\Delta\boldsymbol{t}), \boldsymbol{i}\epsilon\mathbb{Z}^D$. We can also use the quadrature rule (1) to construct the following conservative discretization of $\nabla \cdot F$.

$$(\nabla \cdot \vec{F})^C = \frac{1}{\kappa_{\boldsymbol{i}}h}\Big(\sum_{\pm=+,-} \sum_{d=1}^{D} \pm\alpha_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} F^d_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} + \alpha_{\boldsymbol{i}}^B \vec{F}_{\boldsymbol{i}}^B \cdot \boldsymbol{n_i} \Big) \tag{8}$$

Ideally we would like to use an explicit finite difference approximation to compute $F^s_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \approx F^d(\boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}), \vec{F}_{\boldsymbol{i}}^B \approx \vec{F}(\boldsymbol{x}_{\boldsymbol{i}}^B)$, and use (8) to compute the discrete evolution of U.

$$U_{\boldsymbol{i}}^{n+1} = U_{\boldsymbol{i}}^n - \Delta t (\nabla \cdot \vec{F})_{\boldsymbol{i}}^C \tag{9}$$

The update formula (9) satisfies the following discrete conservation identity.

$$\sum_{\boldsymbol{i}\in\Gamma} \kappa_{\boldsymbol{i}} U_{\boldsymbol{i}}^{n+1} = \sum_{\boldsymbol{i}\in\Gamma} \kappa_{\boldsymbol{i}} U_{\boldsymbol{i}}^n - \frac{\Delta t}{h} \sum_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d\in\partial\Gamma} \alpha_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \vec{F}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \cdot \boldsymbol{n}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \tag{10}$$

where Γ is any collection of control volumes, and $\partial\Gamma$ is the set of cell faces and boundary faces forming the boundary of Γ. The difficulty with this approach is that the CFL stability constraint on the time step is at best $\Delta t = O(h v_{\boldsymbol{i}}^{max}(\kappa_{\boldsymbol{i}})^{\frac{1}{D}})$, where
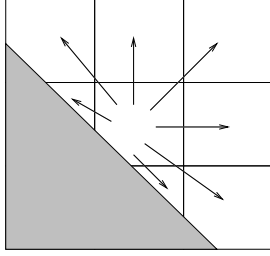
5

Fig. 2. The arrows indicate the control volumes to which δM would be redistributed from the central control volume.

$v_{\boldsymbol{i}}^{max}$ is the magnitude of the maximum wave speed for the $\boldsymbol{i}^{th}$ control volume. This is the well-known small-cell problem for embedded boundary methods. There have been a number of proposals to deal with this problem, including merging the small control volumes with nearby larger ones [13,6], and the development of specialized stencils that guarantee the required cancellations in (8) [4,3]. The approach we have taken to this problem has been to expand the range of influence of the small control volumes algebraically to obtain a stable method [5,2,11]. The starting point for this approach is to compute a stable, but nonconservative approximation to $\nabla \cdot \vec{F}$. One computes the a flux difference on the full Cartesian cell.

$$(\nabla \cdot \vec{F})_{\boldsymbol{i}}^{NC} = \frac{1}{h} \sum_{\pm=+,-} \sum_{d=1}^{D} \pm F_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} \tag{11}$$

where the fluxes in this expression are centered at $(\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d)h$. The initial update uses a linear hybridization of the two estimates of $\nabla \cdot \vec{F}$.

$$U_{\boldsymbol{i}}^{n+1} = U_{\boldsymbol{i}}^{n} - \Delta t(\eta_{\boldsymbol{i}}(\nabla \cdot \vec{F})_{\boldsymbol{i}}^{C} + (1 - \eta_{\boldsymbol{i}})(\nabla \cdot \vec{F})_{\boldsymbol{i}}^{NC}) \tag{12}$$

If we choose, for example, $\eta_{\boldsymbol{i}} = \kappa_{\boldsymbol{i}}$, then the small denominator in $(\nabla \cdot \vec{F})^{C}$ is cancelled, and we obtain a stable method. However, the method fails to conserve, in that it does not satisfy an identity of the form (10). This lack of conservation is measured by the difference between the mass increment $\kappa_{\boldsymbol{i}}(U_{\boldsymbol{i}}^{n+1} - U_{\boldsymbol{i}}^{n})$ given by (9) and that given by (12).

$$\delta M_{\boldsymbol{i}} = -\kappa_{\boldsymbol{i}}(1 - \eta_{\boldsymbol{i}})((\nabla \cdot \vec{F})_{\boldsymbol{i}}^{C} - (\nabla \cdot \vec{F})_{\boldsymbol{i}}^{NC})$$

To maintain overall conservation, we redistribute δM into nearby cells (figure 2):

$$U_{\boldsymbol{i}'}^{n+1} := U_{\boldsymbol{i}'}^{n+1} + w_{\boldsymbol{i},\boldsymbol{i}'}\delta M_{\boldsymbol{i}} , \ \boldsymbol{i}' \in N(\boldsymbol{i}). \tag{13}$$

$$w_{\boldsymbol{i},\boldsymbol{i}'} \geq 0, \quad \sum_{\boldsymbol{i}' \in N(\boldsymbol{i})} w_{\boldsymbol{i},\boldsymbol{i}'}\kappa_{\boldsymbol{i}'} = 1 \tag{14}$$

6

where $N(\boldsymbol{i})$ is some set of indices in the neighborhood of $\boldsymbol{i}$. The sum condition (14) makes the redistribution step conservative: a relationship of the form (10) is satisfied, with some additional boundary terms corresponding to redistribution into or out of the domain Γ. In addition, $w_{\boldsymbol{i},\boldsymbol{i}'}$ must be bounded independent of $(\kappa_{\boldsymbol{i}'})^{-1}$. One example of a redistribution strategy that meets our requirements is $w_{\boldsymbol{i},\boldsymbol{i}'} = (\sum_{\boldsymbol{i}' \in N(\boldsymbol{i})} \kappa_{\boldsymbol{i}'})^{-1}$, where $N(\boldsymbol{i})$ is a set of indices whose components differ from those of $\boldsymbol{i}$ by no more than one. For problems in gas dynamics involving strong shocks, the following mass-weighted redistribution has been observed to be more robust [11]:

$$w_{\boldsymbol{i},\boldsymbol{i}'} = \frac{\rho_{\boldsymbol{i}'}^{NC}}{\sum_{\boldsymbol{i}'' \in N(\boldsymbol{i})} \rho_{\boldsymbol{i}''}^{NC} \, \kappa_{\boldsymbol{i}''}} \tag{15}$$

where ρ^{NC} is a nonconservative estimate of the density at the new time.

$$\rho_{\boldsymbol{i}}^{NC} = \rho_{\boldsymbol{i}}^{n} - \Delta t (\nabla \cdot \vec{F})_{\boldsymbol{i}}^{NC} \tag{16}$$

Our procedure for calculating the fluxes used to compute $(\nabla \cdot \vec{F})^{C}$ assumes that we have a second-order accurate method for computing fluxes at the centers of cell faces.

$$F_{\boldsymbol{i} \pm \frac{1}{2} \boldsymbol{e}^{d}}^{n+\frac{1}{2}} = F^{d}((\boldsymbol{i} \pm \tfrac{1}{2}\boldsymbol{e}^{d})h, t^{n} + \frac{\Delta t}{2}) + O(h^{2}) \tag{17}$$

For all noncovered control volumes $\boldsymbol{i}$. We use these fluxes in (6) to compute $(\nabla \cdot \vec{F})^{NC}$. To compute $\nabla \cdot \vec{F}^{C}$, we interpolate the fluxes (17) to the face centroids, following [8]. In two dimensions, we use linear interpolation.

$$F_{\boldsymbol{i} + \frac{1}{2} \boldsymbol{e}^{d}} = \eta_{d'} F_{\boldsymbol{i} + \frac{1}{2} \boldsymbol{e}^{d}}^{n+\frac{1}{2}} + (1 - \eta_{d'}) F_{\boldsymbol{i} + \frac{1}{2} \boldsymbol{e}^{d} \pm \boldsymbol{e}^{d'}}^{n+\frac{1}{2}} \tag{18}$$

$$\eta_{d'} = 1 - \frac{1}{h} |\boldsymbol{x}_{\boldsymbol{i} + \frac{1}{2} \boldsymbol{e}^{d}} \cdot \boldsymbol{e}^{d'}|, \quad d' \neq d \tag{19}$$

$$\begin{aligned} \pm \boldsymbol{e}^{d'} &= \boldsymbol{e}^{d'} \text{ if } n_{\boldsymbol{i}}^{d'}, n_{\boldsymbol{i}+\boldsymbol{e}^{d}}^{d'} > 0 \\ &= -\boldsymbol{e}^{d'} \text{ if } n_{\boldsymbol{i}}^{d'}, n_{\boldsymbol{i}+\boldsymbol{e}^{d}}^{d'} < 0 \end{aligned} \tag{20}$$

In three dimensions, we use bilinear interpolation, following [15]:

7

$$F_{i+\frac{1}{2}e^d} = (1 - \eta_{d''})(1 - \eta_{d''})F_{i+\frac{1}{2}e^d}^{n+\frac{1}{2}} + (1 - \eta_{d''})\eta_{d'}F_{i+\frac{1}{2}e^d \pm e^d}^{n+\frac{1}{2}}$$

$$+ \eta_{d''}(1 - \eta_{d'})F_{i+\frac{1}{2}e^d \pm e^{d''}}^{n+\frac{1}{2}} + \eta_{d''}\eta_{d'}F_{i+\frac{1}{2}e^d \pm e^{d''} \pm e^{d'}}^{n+\frac{1}{2}}$$

(21)

where $d \neq d' \neq d''$, and $\pm e^{d'}$, $\pm e^{d''}$, $\eta_{d'}$, $\eta_{d''}$ are given by (19), (20).

If we define $U_i^{n,exact} = U(ih, n\Delta t)$, then the truncation error is defined to be

$$\tau_i^{n+\frac{1}{2}} = \frac{(U_i^{n+1,exact} - U_i^{n,exact})}{\Delta t} - L(U^{n,exact})_i$$

where $\Delta t L(U)$ denotes the increment of the discrete solution by one time step outlined above, given data U at the beginning of the time step. This leads to a truncation error estimate

$$\tau_i^{n+\frac{1}{2}} = O(h) \text{ if } i \in N(i') \text{ for some irregular control volume } i'$$
$$= O(h^2) \text{ otherwise}$$

(22)

This truncation error estimate follows from (1) and the fact that $(\nabla \cdot \vec{F})^{NC}$ has a truncation error of $O(h)$, independent of κ. From this it follows that the truncation error of the hybrid method (12) satisfies (22).

The behavior of these methods can be understood from a modified equation analysis. We expect that the solution to the modified equation

$$\frac{\partial U^{mod}}{\partial t} + \nabla \cdot (\vec{F}(U^{mod})) = \tilde{\tau}$$

(23)

approximates the numerical solution to one order higher accuracy than the solution to the original conservation laws do, where $\tilde{\tau}$ is the piecewise constant interpolation of the grid function $\tau^{n+\frac{1}{2}}$ in space and time over each control volume. This suggests that the reduced order of accuracy in (22) will behave differently depending on whether or not the irregular boundary is characteristic. In the case that the boundary is noncharacteristic, a signal is exposed to the $O(h)$ forcing in (23) for a time that is $O(\Delta t) = O(h)$, leading to an integrated contribution to the solution error that is $O(h^2)$. In the case that the boundary is characteristic, a signal propagates along the boundary for an $O(1)$ length of time, leading to an integrated contribution to the solution error that is $O(h)$. However, that contribution is concentrated along the boundary, so that the solution error in the interior remains $O(h^2)$, leading to an overall solution error that is $O(h^2)$ in L^1, $O(h^{\frac{3}{2}})$ in L^2, and $O(h)$ in L^∞. For the

8

gas dynamics examples considered here, a solid wall boundary is characteristic for the particle paths, but mostly noncharacteristic for acoustic wave propagation, so we expect to see convergence rates intermediate between the two limiting cases.

Critical to the success of this approach is the calculation of $(\nabla \cdot \vec{F})^{NC}$. In control volumes with $\kappa_{\boldsymbol{i}} \ll 1$, $(\nabla \cdot \vec{F})^{NC}$ is almost entirely responsible for the update of $U_{\boldsymbol{i}}$. For that reason, $(\nabla \cdot \vec{F})^{NC}$ must be designed carefully, so that, for example, the solution on small control volumes comes into equilibrium with the larger ones around it.

4   Flux Calculation

Given $U_{\boldsymbol{i}}^n$, we need to compute an $O(h^2)$ estimate of the fluxes.

$$F_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}^{n+\frac{1}{2}} \approx F^d((\boldsymbol{i} + \tfrac{1}{2}\boldsymbol{e}^d)h, t^n + \tfrac{1}{2}\Delta t)$$

Specifically, we want to compute the fluxes at the center of the Cartesian grid faces corresponding to the faces of the embedded boundary geometry.

For many applications it is useful to perform the flux calculation using nonconservative variables $W = W(U)$. For the case of polytropic gas dynamics, the primitive variables are $W = (\rho, \boldsymbol{u}, p)^T$. The quasilinear equations for these variables are given as follows:

$$\frac{\partial W}{\partial t} + \sum_{d=0}^{D-1} A^d \frac{\partial W}{\partial x_d} = 0 \tag{24}$$

$$A^d = \nabla_U W \cdot \nabla_U F \cdot \nabla_W U \tag{25}$$

We use an upstream-centered Taylor expansion of the solution from the cell center at the initial time to the cell face at the half-time.

$$
\begin{aligned}
W_{\boldsymbol{i},\pm,d}^{n+\frac{1}{2}} &= W_{\boldsymbol{i}}^n \pm \frac{h}{2}\frac{\partial W}{\partial x} + \frac{\Delta t}{2}\frac{\partial W}{\partial t} \\
&= W_{\boldsymbol{i}}^n \pm \frac{h}{2}\frac{\partial W}{\partial x} - \frac{\Delta t}{2}\sum_{d=0}^{D-1} A^d \frac{\partial W}{\partial x_d}
\end{aligned}
\tag{26}
$$

9

$$W_{i+\frac{1}{2}e^d}^{n+\frac{1}{2}} = \mathcal{R}(W_{i,+,d}^{n+\frac{1}{2}}, W_{i+e^d,-,d}^{n+\frac{1}{2}}, d)$$

where $\mathcal{R}(W_L, W_R, d)$ is the solution to the Riemann problem with left and right states W_L, W_R for the 1D equations in the dth coordinate direction.

In computing $W_{i,\pm,e^d}^{n+\frac{1}{2}}$ we follow the approach in [7,14]. We use centered differences with limiting to approximate the derivatives in the d direction, and a lower-dimensional Godunov method to approximate the remaining tangential derivatives. In extending this to the case of embedded boundaries, it will be necessary to replace extrapolation step (26) on cells that are covered, but adjacent to valid control volumes.

The algorithm is given as follows.

(1)
$$W_{i,\pm,d} = W_i^n + \tfrac{1}{2}(\pm I - \tfrac{\Delta t}{h} A_i^d) P_\pm(\Delta_4^d W_i)$$
$$A_i^d = A^d(W_i)$$
$$P_\pm(\Delta_4^d W_i) = \sum_{\pm\lambda_k > 0} (l_k \cdot \Delta_4^d W_i) r_k$$

where λ_k are eigenvalues of A_i^d, and l_k and r_k are the corresponding left and right eigenvectors.
(2) Transverse predictor step
(i)
$$W_{i+\frac{1}{2}e^d}^{1D} = \mathcal{R}(W_{i,+,d}, W_{i+e^d,-,d}, d) \tag{27}$$
$$F_{i+\frac{1}{2}e^d}^{1D} = F^d(W_{i+\frac{1}{2}e^d}^{1D}) \tag{28}$$

(ii,2D):
$$W_{i,\pm,d}^{n+\frac{1}{2}} = W_{i,\pm,d} - \frac{\Delta t}{2h}\nabla_U W(F_{i+\frac{1}{2}e^d}^{1D} - F_{i-\frac{1}{2}e^d}^{1D}) \tag{29}$$
(ii,3D): For $d_1 \neq d_2 \neq d$

$$W_{i,\pm,d_1,d_2} = W_{i,\pm,d_1} - \frac{\Delta t}{3h}\nabla_U W(F_{i+\frac{1}{2}e^d}^{1D} - F_{i-\frac{1}{2}e^d}^{1D}) \tag{30}$$
$$W_{i+\frac{1}{2}e^d}^{\perp} = \mathcal{R}(W_{i,+,d_1,d_2}, W_{i+e^d,-,d_1,d_2}, d) \tag{31}$$
$$F_{i+\frac{1}{2}e^d}^{\perp} = F^d(W_{i+\frac{1}{2}e^d}^{\perp}) \tag{32}$$
$$W_{i,\pm,d}^{n+\frac{1}{2}} = W_{i,\pm,d} - \frac{\Delta t}{2h}\nabla_U W \sum_{d' \neq d}(F_{i+\frac{1}{2}e^{d'}}^{\perp} - F_{i-\frac{1}{2}e^{d'}}^{\perp}) \tag{33}$$

(iii)

$$W^{n+\frac{1}{2}}_{i+\frac{1}{2}e^d} = \mathcal{R}(W^{n+\frac{1}{2}}_{i,+,d}, W^{n+\frac{1}{2}}_{i+e^d,-,d}, d) \tag{34}$$

$$F^{n+\frac{1}{2}}_{i+\frac{1}{2}e^d} = F^d(W^{n+\frac{1}{2}}_{i+\frac{1}{2}e^d}) \tag{35}$$

5   Algorithm Details

5.1   *Slope Calculation*

The notation

$$CC = A \mid B \mid C$$

means that the 3-point formula A is used for CC if all cell-centered values it uses are available, the 2-point formula B is used if the cell to the right (i.e. the high side) of the current cell is covered, and the 2-point formula C is used if the cell to the left (i.e. the low side) current cell is covered.

To compute the limited differences in the first step on the algorithm, we use the fourth-order slope calculation [7] combined with characteristic limiting.

$$\Delta^d_4 W_{\boldsymbol{i}} = \zeta_{\boldsymbol{i}} \, \tilde{\Delta}^d_4 W_{\boldsymbol{i}}$$

$$\tilde{\Delta}^d_4 W_{\boldsymbol{i}} = \Delta^{vL}(\Delta^B W_{\boldsymbol{i}}, \Delta^L W_{\boldsymbol{i}}, \Delta^R W_{\boldsymbol{i}}) \mid \Delta^d_2 W_{\boldsymbol{i}} \mid \Delta^d_2 W_{\boldsymbol{i}}$$

$$\Delta^d_2 W_{\boldsymbol{i}} = \Delta^{vL}(\Delta^C W_{\boldsymbol{i}}, \Delta^L W_{\boldsymbol{i}}, \Delta^R W_{\boldsymbol{i}}) \mid \Delta^{VLL} W_{\boldsymbol{i}} \mid \Delta^{VLR} W_{\boldsymbol{i}}$$

$$\Delta^B W_{\boldsymbol{i}} = \tfrac{2}{3}((W - \tfrac{1}{4}\Delta^d_2 W)_{\boldsymbol{i}+e^d} - (W + \tfrac{1}{4}\Delta^d_2 W)_{\boldsymbol{i}-e^d})$$

$$\Delta^C W_{\boldsymbol{i}} = \tfrac{1}{2}(W^n_{\boldsymbol{i}+e^d} - W^n_{\boldsymbol{i}-e^d})$$

$$\Delta^L W_{\boldsymbol{i}} = W^n_{\boldsymbol{i}} - W^n_{\boldsymbol{i}-e^d}$$

$$\Delta^R W_{\boldsymbol{i}} = W^n_{\boldsymbol{i}+e^d} - W^n_{\boldsymbol{i}}$$

$$\Delta^{3L} W_{\boldsymbol{i}} = \tfrac{1}{2}(3W^n_{\boldsymbol{i}} - 4W^n_{\boldsymbol{i}-e^d} + W^n_{\boldsymbol{i}-2e^d}$$

$$\Delta^{3R} W_{\boldsymbol{i}} = \tfrac{1}{2}(-3W^n_{\boldsymbol{i}} + 4W^n_{\boldsymbol{i}+e^d} - W^n_{\boldsymbol{i}+2e^d})$$

11

$$\Delta^{VLL}W_{\pmb{i}} = \min(\Delta^{3L}W_{\pmb{i}}, \Delta^{L}W_{\pmb{i}}) \quad \text{if } \Delta^{3L}W_{\pmb{i}} \cdot \Delta^{L}W_{\pmb{i}} > 0$$

$$\Delta^{VLL}W_{\pmb{i}} = 0 \qquad\qquad\qquad\quad \text{otherwise}$$

$$\Delta^{VLR}W_{\pmb{i}} = \min(\Delta^{3R}W_{\pmb{i}}, \Delta^{R}W_{\pmb{i}}) \quad \text{if } \Delta^{3R}W_{\pmb{i}} \cdot \Delta^{R}W_{\pmb{i}} > 0$$

$$\Delta^{VLR}W_{\pmb{i}} = 0 \qquad\qquad\qquad\quad \text{otherwise}$$

There are two versions of the van Leer limiter $\Delta^{vL}(\delta W_C, \delta W_L, \delta W_R)$ that are commonly used. One is to apply a limiter to the differences in characteristic variables. We compute expansions of one-sided and centered differences in characteristic variables and apply van Leer limiter.

$$\Delta^{vL} = \sum_k \alpha^k r^k$$

$$\alpha^k = \min(2\,|\,\alpha_L^k\,|,\, 2\,|\,\alpha_R^k\,|,\, |\,\alpha_C^k\,|) \text{ if } \alpha_L^k \cdot \alpha_R^k > 0$$
$$= 0 \text{ otherwise}$$

$$\alpha_L^k = l^k \cdot \delta W_L \ , \ \alpha_R^k = l^k \cdot \delta W_R \ , \ \alpha_C^k = l^k \cdot \delta W$$

Here $l^k = l^k(W_{\pmb{i}}^n), r^k = r^k(W_{\pmb{i}}^n)$. For a variety of problems, including our gas dynamics example, it suffices to apply the van Leer limiter componentwise to the differences. Formally, this can be obtained from the more general case above by taking the matrices of left and right eigenvectors to be the identity.

As discussed in [16], for Godunov methods it is necessary to introduce additional dissipation at strong compressive discontinuities in continum mechanics problems. Following the approach in [7], we do this using two mechanisms. One, discussed in Section 4.3, is to add a small amount of antificial viscosity to the fluxes. The other is to introduce additional slope limiting in places where the steepness of the discontinuity exceeds some threshhold. We compute a flattening coefficient $\zeta_{\pmb{i}}$, a multiplicative factor by which we reduce the slopes. We assume that there is a quantity corresponding to the pressure in gas dynamics (denoted here as p) which can act as a steepness indicator, and a quantity corresponding to the bulk modulus (denoted here as K, given as γp in a gas), that can be used to non-dimensionalize differences in p.

$$\zeta_{\pmb{i}} = \min_{0 \le d < \pmb{D}}(\zeta_{\pmb{i}}^d) \quad \text{if } \sum_{d=0}^{\pmb{D}-1} \Delta_1^d u_{\pmb{i}}^d < 0$$
$$\zeta_{\pmb{i}} = 1 \qquad\qquad \text{otherwise}$$

$$\zeta_{\boldsymbol{i}}^d = min_3(\widetilde{\zeta}^d, d)_{\boldsymbol{i}}$$

$$\widetilde{\zeta}_{\boldsymbol{i}}^d = \eta(\Delta_1^d p_{\boldsymbol{i}}, \Delta_2^d p_{\boldsymbol{i}}, min_3(\mathrm{K}, d)_{\boldsymbol{i}})$$

$$\Delta_1^d p_{\boldsymbol{i}} = \Delta^C p_{\boldsymbol{i}} \mid \Delta^L p_{\boldsymbol{i}} \mid \Delta^R p_{\boldsymbol{i}}$$

$$\Delta_2^d p_{\boldsymbol{i}} = \Delta_1^d p_{\boldsymbol{i}+\boldsymbol{e}^d} + \Delta_1^d p_{\boldsymbol{i}-\boldsymbol{e}^d} \mid 2\Delta_1^d p_{\boldsymbol{i}} \mid 2\Delta_1^d p_{\boldsymbol{i}}$$

$$min_3(q, d)_{\boldsymbol{i}} = \min(q_{\boldsymbol{i}+\boldsymbol{e}^d}, q_{\boldsymbol{i}}, q_{\boldsymbol{i}-\boldsymbol{e}^d}) \mid \min(q_{\boldsymbol{i}}, q_{\boldsymbol{i}-\boldsymbol{e}^d}) \mid \min(q_{\boldsymbol{i}}, q_{\boldsymbol{i}+\boldsymbol{e}^d})$$

$$\zeta(\delta p_1, \delta p_2, p_0) = 0 \qquad \text{if } \frac{|\delta p_1|}{p_0} > d \text{ and } \frac{|\delta p_1|}{|\delta p_2|} > r_1$$

$$\zeta(\delta p_1, \delta p_2, p_0) = 1 - R \quad \text{if } \frac{|\delta p_1|}{p_0} > d \text{ and } r_1 \geq \frac{|\delta p_1|}{|\delta p_2|} > r_0$$

$$\zeta(\delta p_1, \delta p_2, p_0) = 1 \qquad \text{otherwise}$$

where

$$R = \frac{\frac{|\delta p_1|}{|\delta p_2|} - r_0}{r_1 - r_0}$$

and the values we use for the parameters above are $r_0 = 0.75, r_1 = 0.85, d = 0.33$.

5.2 Extrapolation to Covered Faces

A covered face is a face whose aperature vanishes. To compute the stable, nonconservative divergence of the flux (see (11)), we need a second-order flux at covered faces. The flux is obtained by solving a Riemann problem at the face. For the side of the face next to the control volume, we use the extrapolated state from the control volume. For the other side of the covered face, we must extrapolate from neighboring values at the same orientation of the face.

Specifically, assume that all $\boldsymbol{i}$ is not covered, but $\boldsymbol{i} \mp \boldsymbol{e}^d$ is covered, so that the face connecting the two is covered. Then we want to compute $W_{\boldsymbol{i} \mp \boldsymbol{e}^d, \pm, d}$, given a collection of values $\{W_{\boldsymbol{i}', \pm, d}\}$ that are assumed to be defined if $\alpha_{\boldsymbol{i}' \pm \frac{1}{2} \boldsymbol{e}^d} \neq 0$.

5.2.1 Two-dimensional Extrapolation

In two dimensions, extrapolation to covered faces is done as illustrated in figure 3. First we define the control volumes involved.

$$\boldsymbol{i}^u = \boldsymbol{i} + s^{d'} \boldsymbol{e}^{d'} - s^d \boldsymbol{e}^d$$

$$\boldsymbol{i}^s = \quad \boldsymbol{i} + s^d \boldsymbol{e}^d$$

$$\boldsymbol{i}^c = \quad \boldsymbol{i} + s^{d'} \boldsymbol{e}^{d'}$$

13

where $d' \neq d$ and $s^d = \mathrm{sign}(n^d)$.

Define $W^{u,s,c}$, extrapolations to the edges near the control volumes near $\boldsymbol{i}$.

$$W^u = W_{\boldsymbol{i}^u, \mp, d}$$
$$W^s = W_{\boldsymbol{i}^s, \mp, d} - s^d \Delta^d W$$
$$W^c = W_{\boldsymbol{i}^c, \mp, d}$$

To extrapolate to the covered faces, we use a linear combination of the values defined above to compute the value along a ray normal to the boundary and passing through the center of the covered face. We then extrapolate that value to the covered face using the second-order slopes combined with characteristic limiting described in section 5.1. In the case where one of the values being used to interpolate corresponds to a value on the cell adjacent to the covered face in question, (the case illustrated in figure 3) we use a value extrapolated from $\boldsymbol{i}^s$ (the cell adjacent in the d direction) rather than $\boldsymbol{i}$. This choice satisfies the design criterion that the action of the nonconservative evolution should, over time, tend to make the solution at $\boldsymbol{i}$ tend toward the value of a locally constant solution in the surrounding cells. This was the design criterion for computing covered faces in [11]; the procedure given here has the same goal, but using an approach that produces second-order accurate fluxes. For example, in the case of a linear equation and the normal pointing in the $\boldsymbol{e}^1$ direction, extrapolation from a locally constant state to the right of $\boldsymbol{i}$ in figure 3 leads to the solution in $\boldsymbol{i}$ to eventually take on that constant value. If one used the value at the face extrapolated from $\boldsymbol{i}$, the solution would tend to the locally constant value be true only if the advection velocity were negative; otherwise, the value at $\boldsymbol{i}$ would remain unchanged.

If $|n_d| < |n_{d'}|$:

$$W_{\boldsymbol{i} \mp \boldsymbol{e}^d, \pm, d} = \frac{|n_d|}{|n_{d'}|} W^c + (1 - \frac{|n_d|}{|n_{d'}|}) W^u - (\frac{|n^d|}{|n^{d'}|} s^d \Delta^d W + s^{d'} \Delta^{d'} W)$$

$$\Delta^{d''} W = \frac{|n_d|}{|n_{d'}|} \Delta_2^{d''} W_{\boldsymbol{i}^c}^n + (1 - \frac{|n_d|}{|n_{d'}|}) \Delta_2^{d''} W_{\boldsymbol{i}^u}^n, \ d'' = 1, 2 \tag{36}$$

If $|n_d| \geq |n_{d'}|$:

$$W_{\boldsymbol{i} \mp \boldsymbol{e}^d, \pm, d} = \frac{|n_{d'}|}{|n_d|} W^c + (1 - \frac{|n_{d'}|}{|n_d|}) W^s - (\frac{|n^{d'}|}{|n^d|} s^{d'} \Delta^{d'} W + s^d \Delta^d W)$$

$$\Delta^{d''} W = \frac{|n_{d'}|}{|n_d|} \Delta_2^{d''} W_{\boldsymbol{i}^c}^n + (1 - \frac{|n_{d'}|}{|n_d|}) \Delta_2^{d''} W_{\boldsymbol{i}^s}^n, \ d'' = 1, 2 \tag{37}$$
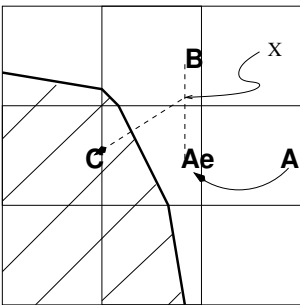
Fig. 3. Illustration of extrapolation to covered faces in two dimensions The covered face is at C. We extrapolate from A to Ae and interpolate between Ae and B to the point X where the boundary normal intersects the line. We then extrapolate back along the normal to get to the covered face.

We found that the use of the linear interpolation algorithms (36), (37) to compute the slopes used in extrapolating to the covered faces led to a more robust and accurate algorithm than other simpler choices that we considered. The intent is to use slopes computed at the same cell centers as the values used in the original linear interpolation in figure 3, and in the same proportions. By using that choice, it appears that no further limiting of those slopes is required.

If one or both of the faces from which we are extrapolating are covered we drop order. If only one of the faces is covered we set the extrapolated value to be the value on the other face. If both faces are covered, we set the extrapolated value to $W_{\boldsymbol{i}}^n$.

5.2.2   Extrapolation to Covered Face in Three Dimensions

We define the direction of the face normal to be d and d_1, d_2 to be the directions tangential to the face. The procedure extrapolation procedure is given as follows.

- Define the associated control volumes.
- Form a 2×2 grid of values along a plane h away from the covered face and bilinearly interpolate to the point where the normal intersects the plane.
- Use the slopes of the solution to extrapolate along the normal to obtain a second-order approximation of the solution at the covered face.

Which plane is selected is determined by the direction of the normal. See figure 4 for an illustration.

If $|n^d| \geq |n^{d_1}|, |n^{d_2}|$, we define a bilinear function $\mathcal{B}$ that interpolates the 2×2 grid of values.

$$\mathcal{B}(Q, \Delta) = A + B\xi + C\eta + D\xi\eta - \zeta\Delta_{\boldsymbol{i}^{00}} \tag{38}$$

$$A = Q_{\boldsymbol{i}^{00}}$$
$$B = s^{d_1}(Q_{\boldsymbol{i}^{01}} - Q_{\boldsymbol{i}^{00}})$$
$$C = s^{d_2}(Q_{\boldsymbol{i}^{10}} - Q_{\boldsymbol{i}^{00}})$$
$$D = s^{d_1}s^{d_2}(Q_{\boldsymbol{i}^{11}} - Q_{\boldsymbol{i}^{00}}) - (Q_{\boldsymbol{i}^{10}} - Q_{\boldsymbol{i}^{00}}) - (Q_{\boldsymbol{i}^{01}} - Q_{\boldsymbol{i}^{00}})$$

$$\xi = \frac{|n_{d_1}|}{|n_d|}, \quad \eta = \frac{|n_{d_2}|}{|n_d|}, \quad \zeta = -1 + s^{d_1}\xi + s^{d_2}\eta + (s^{d_1}s^{d_2} - 2s^d)\xi\eta$$

$$s^{d_i} = sign(n_{d_i})$$
$$\boldsymbol{i}^{00} = \boldsymbol{i} + s^d \boldsymbol{e}^d$$
$$\boldsymbol{i}^{10} = \boldsymbol{i} + s^{d_1} \boldsymbol{e}^{d_1}$$
$$\boldsymbol{i}^{01} = \boldsymbol{i} + s^{d_2} \boldsymbol{e}^{d_2}$$
$$\boldsymbol{i}^{11} = \boldsymbol{i} + s^{d_1} \boldsymbol{e}^{d_1} + s^{d_2} \boldsymbol{e}^{d_2}$$

$\mathcal{B}$ interpolates the values in the (d_1, d_2) plane in figure 4, with $Q_{\boldsymbol{i}^{00}}$ the value at A, $Q_{\boldsymbol{i}^{00}} - s^d \Delta_{\boldsymbol{i}^{00}}$ the value at point B, and the remaining values filling in the bilinear stencil. Using this function, we can define the extrapolated value on the covered face.

$$\begin{aligned} W_{\boldsymbol{i} \mp \boldsymbol{e}^d, \pm, d} =& \mathcal{B}(W_{\cdot, \pm, d}, \Delta_2^d W^n) - \mathcal{B}(\Delta_2^d W^n, \Delta \equiv 0) \\ & - s^{d_1}\frac{|n^{d_1}|}{|n^d|}\mathcal{B}(\Delta_2^{d_1}W^n, \Delta \equiv 0) - s^{d_2}\frac{|n^{d_2}|}{|n^d|}\mathcal{B}(\Delta^{d_2}W, \Delta \equiv 0) \qquad (39) \\ & d_1 \neq d_2 \neq d \end{aligned}$$

We use the bilinear stencil to interpolate values of both the solution and of the slopes, except that we use piecewise-constant extrapolation to extrapolate the value of the slopes from A to B.

The case where one of the tangential directions corresponds to the largest component of the normal is similar. Assuming $|n^{d_1}| > |n^d|, |n^{d_2}|$, we define

$$\mathcal{B}(Q) = A + B\zeta + C\eta + D\xi\eta \qquad (40)$$

with

16

$$A = Q_{\boldsymbol{i}^{00}} \tag{41}$$

$$B = s^d(Q_{\boldsymbol{i}^{01}} - Q_{\boldsymbol{i}^{00}}) \tag{42}$$

$$C = s^{d_2}(Q_{\boldsymbol{i}^{10}} - Q_{\boldsymbol{i}^{00}}) \tag{43}$$

$$D = s^d s^{d_2}(Q_{\boldsymbol{i}^{11}} - Q_{\boldsymbol{i}^{00}}) - (Q_{\boldsymbol{i}^{01}} - Q_{\boldsymbol{i}^{00}}) - (Q_{\boldsymbol{i}^{10}} - Q_{\boldsymbol{i}^{00}}) \tag{44}$$

$$\zeta = \frac{|n^d|}{|n^{d_1}|}, \ \eta = \frac{|n^{d_2}|}{|n^{d_1}|} \tag{45}$$

$$\boldsymbol{i}^{00} = \boldsymbol{i} + s^{d_1}\boldsymbol{e}^{d_1} - s^d\boldsymbol{e}^d$$

$$\boldsymbol{i}^{10} = \boldsymbol{i} + s^{d_1}\boldsymbol{e}^{d_1} - s^d\boldsymbol{e}^d + s^{d_2}\boldsymbol{e}^{d_2}$$

$$\boldsymbol{i}^{01} = \boldsymbol{i} + s^{d_1}\boldsymbol{e}^{d_1}$$

$$\boldsymbol{i}^{11} = \boldsymbol{i} + s^{d_1}\boldsymbol{e}^{d_1} - s^{d_2}\boldsymbol{e}^{d_2}$$

Then

$$
\begin{aligned}
W_{\boldsymbol{i}\mp\boldsymbol{e}^d,\pm,d} =& \mathcal{B}(W_{\cdot,\pm,d}) - \mathcal{B}(\Delta_2^{d_1}W) \\
& - s^d\frac{|n^d|}{|n^{d_1}|}\mathcal{B}(\Delta_2^d W) - s^{d_2}\frac{|n^{d_2}|}{|n^{d_1}|}\mathcal{B}(\Delta_2^{d_2}W)
\end{aligned} \tag{46}
$$

If any of the values required to perform the interpolation are unavailable, e.g. because the cells are covered, we drop order by using a weighted sum of the available values:

$$W_{\boldsymbol{i}\mp\boldsymbol{e}^d,\pm,d} = \frac{\sum_{\boldsymbol{i}'} W_{\boldsymbol{i}',\pm,d}\kappa_{\boldsymbol{i}'}}{\sum_{\boldsymbol{i}'}\kappa_{\boldsymbol{i}'}} \tag{47}$$

where the sums are over $\boldsymbol{i}' \in \{\boldsymbol{i}^{00}, \boldsymbol{i}^{01}, \boldsymbol{i}^{10}, \boldsymbol{i}^{11}\}$, provided that at least one of the $\boldsymbol{i}'$ is not covered. If all of the faces used for interpolation are covered, we set the extraoplated value to be $W_{\boldsymbol{i}}^n$.
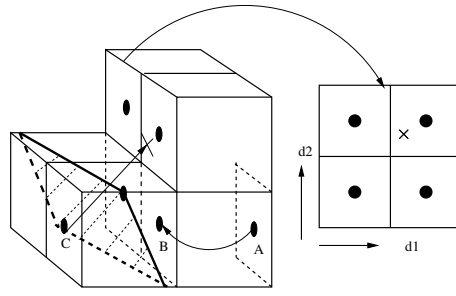
Fig. 4. Illustration of extrapolation to covered faces in three dimensions. The covered face is at C. We extrapolate from A to B to form a plane of values in $d1 - d2$. We interpolate within that plane to the point X where the boundary normal intersects the plane. We then extrapolate back along the normal to get to the covered face.

5.3 Artifical Viscosity

For compressive discontinuities in continuum mechanics problems, we add a small artificial viscosity. This takes the form of an increment to the flux by an undivided difference of U^n.

$$F^{n+\frac{1}{2}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} := F^{n+\frac{1}{2}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - K_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}(U^n_{\boldsymbol{i}+\boldsymbol{e}^d} - U^n_{\boldsymbol{i}})$$

$$K_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = K_0 \max(-(D\boldsymbol{u}^n)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, 0)$$

$$(D\boldsymbol{u})_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = (u^d_{\boldsymbol{i}+\boldsymbol{e}^d} - u^d_{\boldsymbol{i}}) + \sum_{d' \neq d} \frac{1}{2}(\Delta^{d'} u^{d'}_{\boldsymbol{i}+\boldsymbol{e}^{d'}} + \Delta^{d'} u^{d'}_{\boldsymbol{i}})$$

$$\Delta^{d'} u^{d'}_{\boldsymbol{i}} = \tfrac{1}{2}(u^{d'}_{\boldsymbol{i}+\boldsymbol{e}^{d'}} - u^{d'}_{\boldsymbol{i}-\boldsymbol{e}^{d'}})|u^{d'}_{\boldsymbol{i}} - u^{d'}_{\boldsymbol{i}-\boldsymbol{e}^{d'}}|u^{d'}_{\boldsymbol{i}+\boldsymbol{e}^{d'}} - u^{d'}_{\boldsymbol{i}} \qquad (48)$$

We modify the covered face with the same divergence used in the adjacent uncovered face. If $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ is covered, but $\boldsymbol{i} - \frac{1}{2}\boldsymbol{e}^d$ is not, we set

$$F^{n+\frac{1}{2}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} := F^{n+\frac{1}{2}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - K_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}(U^n_{\boldsymbol{i}} - U^n_{\boldsymbol{i}-\boldsymbol{e}^d})$$

This has the effect of negating the effect of artificial viscosity on the non-conservative divergence of the flux at irregular cells.

18

5.4 Computing fluxes at the irregular boundary

The flux at the embedded boundary is centered at the centroid of the boundary $\boldsymbol{x}^B$. We extrapolate the primitive solution in space from the cell center. We then transform to the conservative solution and extrapolate in time using the stable, non-conservative estimate of the flux divergence.

$$W_{\boldsymbol{i},B} = W_{\boldsymbol{i}}^n + \sum_{d=0}^{D-1} (x_d^B \Delta_4^d W_{\boldsymbol{i}}^n) \tag{49}$$

$$W_{\boldsymbol{i},B}^{n+\frac{1}{2}} = W_{\boldsymbol{i},B} - \frac{\Delta t}{2} \nabla_U W (D \cdot \vec{F})^{NC} \tag{50}$$

$$F_{\boldsymbol{i},B}^{n+\frac{1}{2}} = R_B(W_{\boldsymbol{i},B}^{n+\frac{1}{2}}, \boldsymbol{n_i}) \tag{51}$$

Here R_B denotes the solution to the boundary Riemann problem, which takes a value of W at the boundary and returns a flux that satisfies the boundary conditions. For continuum mechanics problems in which we are using an artificial viscosity, we calculate an approximation of the divergence of the velocity at the irregular cell $(D\boldsymbol{u})_{\boldsymbol{i}}$ and use it to modify the flux to be consistent with artificial viscosity. The d-direction momentum flux at the irregular boundary is given by $p^B n^d$ where p^B is the pressure to emerge from the Riemann solution in equation (51). For artificial viscosity, we modify this flux as follows:

$$(D\boldsymbol{u})_{\boldsymbol{i}} = \sum_{d'=0}^{D-1} \Delta^{d'} u_{\boldsymbol{i}}^{d'}$$

$$p_{\boldsymbol{i}}^B := p_{\boldsymbol{i}}^B - 2K_0 \max(-(D\boldsymbol{u}^n)_{\boldsymbol{i}}, 0) \boldsymbol{u}_{\boldsymbol{i}}^n \cdot \boldsymbol{n_i}$$

where $\Delta^{d'} u^{d'}$ is defined as in (48).

6 Results

6.1 Convergence Tests

Our test problem is a simple wave propagating in a straight circular channel (a straight-walled channel in two dimensions). The flow field is a stagnant fluid with a small perturbation in a single Riemann invariant. We specify an initial profile for density at time $t = 0$,

$$\rho_0(x) = \rho_{\text{ref}} \left(1 + \alpha f(\bar{x})\right) \tag{52}$$

where

$$f(\bar{x}) = \begin{cases} (\bar{x}^2 - 1)^4 & \text{if } 0 \le \bar{x} \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

with the dimensionless coordinate

$$\bar{x} = \vec{x} \cdot \hat{n} / w. \tag{54}$$

The parameters are α, the amplitude of the wave; w, the width of the wave; and $\hat{n}$, the direction of propagation of the wave. The initial pressure is found from the isentropic relation

$$\ln p_0(x) - \gamma \ln \rho_0(x) = \ln p_{\text{ref}} - \gamma \ln \rho_{\text{ref}}. \tag{55}$$

The initial fluid velocity is found by characteristic analysis. The value of the Riemann invariant

$$J_+ = u + \frac{2c}{\gamma - 1} \tag{56}$$

is taken from the profile $u = u_0(x)$, $c = c_0(x)$, while the Riemann invariant

$$J_- = u - \frac{2c}{\gamma - 1} \tag{57}$$

is taken from the reference ambient conditions $u = 0$, $c = c_{\text{ref}}$. Equating

$$u_0(x) = \frac{1}{2} \left(J_+ + J_-\right) \tag{58}$$

yields

$$u_0(x) = \frac{2}{\gamma - 1} \left(c_0(x) - c_{\text{ref}}\right) \tag{59}$$

The exact solution $u(x, t)$ is obtained by using the profile $u_0(x_+)$, $c_0(x_+)$ in (56), where $x_+(x, t) = x - (u + c)t$, and iterating to convergence of x_+.

In the two-dimensional problem, the sides of the channel are at a 30 degree angle with the x-axis, and the width of the channel is .125. In three dimensions, axis of the

20

cylinder is $(4, 1, 1)$ and its radius is 0.0625. The parameters of the pulse are $\alpha = 10^{-4}$, $w = 0.2$, and its direction is along the axis of the cylinder. The finer grid is $256 \times 64 \times 64$ and the coarser is $128 \times 32 \times 32$ in three dimensions. For the two-dimensional problem, the grids are 256×256 and 128×128.

The truncation error at a given grid resolution, τ^h, is approximated by

$$\tau^h = \frac{U^h(\Delta t) - U^e(\Delta t)}{\Delta t} \tag{60}$$

where $U^h(t)$ is the numerical solution, and $U^e(t)$ is the exact solution evaluated at grid points. The solution error at a given grid resolution, ϵ^h, is approximated by

$$\epsilon = U^h(t) - U^e(t) \tag{61}$$

Where t is some fixed time interval independent of the mesh spacing. The order of convergence p is estimated by

$$p = \frac{\log(\frac{|\epsilon^{2h}|}{|\epsilon^h|})}{\log(2)} \tag{62}$$

Finally, for the purpose of the convergence study, we have turned off the van Leer limiters, using instead the linear difference formulas for computing slopes. this allows us to determine the extent to which the modified equation analysis is valid, without the contaminating effects of limiters acting at extrema in the interior of the domain. Similarly, we have also set the artificial viscosity coefficient to zero.

The results of these convergence tests are shown in tables 1 through 12. In both two and three dimensions, we obtain the expected truncation results in all norms. The truncation error is first order in max norm, and because of the estimate (22) first order at irregular cells. Since there are $O(n^{D-1})$, irregular cells, it follows that the truncation error should be second order in L^1 and $O(h^{1.5})$ in L^2.

The solution error is second-order in L^1, as expected, and exhibits behavior intermediate between first-order and second-order accuracy in other norms. This is consistent with the modified equation picture of the effect of the truncation error on the solution error: the boundary is characteristic for the particle paths, but mostly non-characteristic for acoustic waves, leading both first- and second-order contributions to the solution error.

6.2 Strong Shock Calculations

We have implemented an adaptive mesh refinement (AMR) version of the embedded boundary solver described above using the same fully conservative method described

Variable	Coarse Error	Fine Error	Order
mass-density	1.239713e-05	6.472469e-06	9.376179e-01
x-momentum	2.180364e-05	1.240106e-05	8.141054e-01
y-momentum	5.344646e-05	2.449178e-05	1.125797e+00
energy-density	4.339256e-05	2.265505e-05	9.376146e-01

Table 1

Truncation error convergence rates using L^∞ norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$

Variable	Coarse Error	Fine Error	Order
mass-density	2.725599e-07	6.900854e-08	1.981727e+00
x-momentum	2.441675e-07	6.253509e-08	1.965133e+00
y-momentum	4.452873e-07	1.134483e-07	1.972702e+00
energy-density	9.539919e-07	2.415380e-07	1.981727e+00

Table 2

Truncation error convergence rates using L^1 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$

Variable	Coarse Error	Fine Error	Order
mass-density	6.701257e-07	2.113327e-07	1.664916e+00
x-momentum	9.718477e-07	3.349814e-07	1.536649e+00
y-momentum	2.309499e-06	8.063959e-07	1.518019e+00
energy-density	2.345529e-06	7.396932e-07	1.664915e+00

Table 3

Truncation error convergence rates using L^2 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$

Variable	Coarse Error	Fine Error	Order
mass-density	2.198289e-05	1.165817e-05	9.150395e-01
x-momentum	2.441534e-05	1.257685e-05	9.570173e-01
y-momentum	3.657839e-05	1.977152e-05	8.875675e-01
z-momentum	3.657839e-05	1.977152e-05	8.875675e-01
energy-density	7.694624e-05	4.080685e-05	9.150394e-01

Table 4

Truncation error convergence rates using L^∞ norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

Variable	Coarse Error	Fine Error	Order
mass-density	5.532517e-07	1.402915e-07	1.979508e+00
x-momentum	4.926733e-07	1.200369e-07	2.037153e+00
y-momentum	3.884727e-07	9.850097e-08	1.979604e+00
z-momentum	3.884727e-07	9.850097e-08	1.979604e+00
energy-density	1.936452e-06	4.910386e-07	1.979508e+00

Table 5

Truncation error convergence rates using L^1 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

Variable	Coarse Error	Fine Error	Order
mass-density	2.010660e-06	6.881709e-07	1.546830e+00
x-momentum	2.254747e-06	7.602553e-07	1.568410e+00
y-momentum	1.854337e-06	6.572475e-07	1.496395e+00
z-momentum	1.854337e-06	6.572475e-07	1.496395e+00
energy-density	7.037591e-06	2.408695e-06	1.546830e+00

Table 6

Truncation error convergence rates using L^2 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

Variable	Coarse Error	Fine Error	Order
mass-density	1.158701e-07	2.706018e-08	2.098265e+00
x-momentum	2.214051e-07	7.641742e-08	1.534715e+00
y-momentum	1.583313e-07	4.180777e-08	1.921104e+00
energy-density	4.055756e-07	9.471992e-08	2.098231e+00

Table 7

Solution error convergence rates using L^∞ norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$

Variable	Coarse Error	Fine Error	Order
mass-density	1.843621e-09	4.201618e-10	2.133525e+00
x-momentum	2.429062e-09	5.872275e-10	2.048408e+00
y-momentum	1.452380e-09	3.212463e-10	2.176667e+00
energy-density	6.452813e-09	1.470597e-09	2.133527e+00

Table 8

Solution error convergence rates using L^1 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$
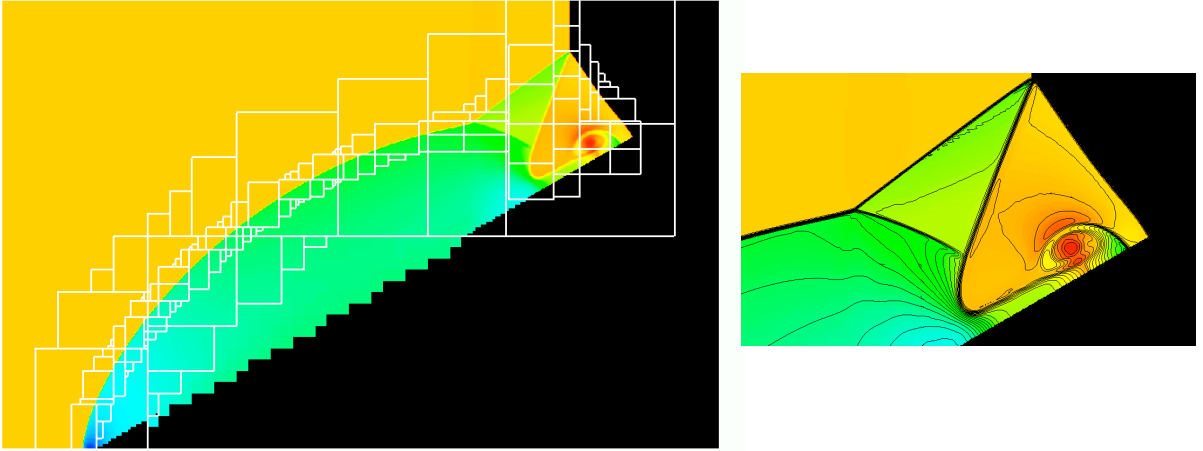
Fig. 5. Embedded boundary calculation of shock reflection problem in [16]: density field. The shock Mach number is 10, the ramp angle is 30 degrees, and $\gamma = 1.4$. Left: complete reflection region, showing the AMR grid hierarchy. Right: blowup of double Mach region, with contours. The problem domain is a $1 \times .625$ rectangle, the effective grid resolution at the finest level is 1024×640, and the output time is .064.

in [11]. We use this implementation to compute solutions to time-dependent compressible flow with strong shocks.

In figure 5, we show results for the double Mach reflection test problem in [16] using an embedded boundary representation of the ramp. In figure 6 we show the same problem computed using the method in [7] on the grid-aligned configuration in [16]. There are two differences between the grid-aligned and embedded boundary solutions. One is that the grid-aligned case displays signs of incipient Kelvin-Helmholtz instability in the main contact discontinuity, while the embedded boundary solution at the same resolution does not. Such a result is not surprising, since the contact discontinuity in the grid-aligned solution is moving much more slowly in the vertical direction than the embedded boundary solution, and consequently undergoes less dissipation. The second difference is the signal originating in the grid-aligned case from the point where the incident shock meets the top boundary. This signal is a numerical artifact stemming from the mismatch between the discrete traveling wave profile of the computed shock and the analytic solution imposed as a boundary condition. The embedded boundary calculation uses a solid wall boundary condition at the top boundary, which does not lead to such an artifact. Otherwise, the solutions are very similar. In particular, the leading edge of the jet in the double Mach region intersects the boundary at right angles in both cases, as it should. This particular feature of the solution is very sensitive to the details of the solid wall boundary condition, even in the grid-aligned case, and is a discriminating test of the accuracy of the embedded boundary representation of the boundary conditions.

24

Variable	Coarse Error	Fine Error	Order
mass-density	5.276566e-09	1.109110e-09	2.250197e+00
x-momentum	7.478798e-09	1.873967e-09	1.996711e+00
y-momentum	6.904018e-09	1.577492e-09	2.129804e+00
energy-density	1.846867e-08	3.882026e-09	2.250198e+00

Table 9

Solution error convergence rates using L^2 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 2$

Variable	Coarse Error	Fine Error	Order
mass-density	3.459602e-07	1.173065e-07	1.560324e+00
x-momentum	5.518585e-07	2.573757e-07	1.100422e+00
y-momentum	3.790191e-07	1.299267e-07	1.544573e+00
z-momentum	3.790191e-07	1.299267e-07	1.544573e+00
energy-density	1.210945e-06	4.106031e-07	1.560317e+00

Table 10

Solution error convergence rates using L^∞ norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

Variable	Coarse Error	Fine Error	Order
mass-density	1.549612e-08	3.785211e-09	2.033461e+00
x-momentum	1.920087e-08	4.955733e-09	1.954001e+00
y-momentum	9.875944e-09	2.831778e-09	1.802210e+00
z-momentum	9.875944e-09	2.831778e-09	1.802210e+00
energy-density	5.423856e-08	1.324875e-08	2.033462e+00

Table 11

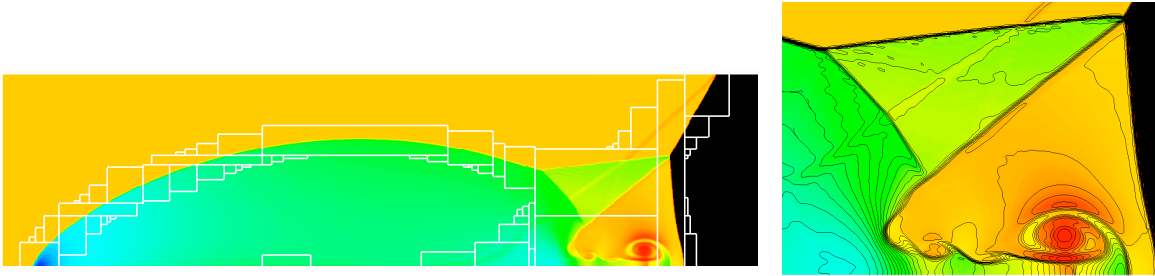Solution error convergence rates using L^1 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

Fig. 6. Grid-aligned shock reflection calculation. The problem domain is a $1 \times .25$ rectangle, and the effective resolution at the finest level of 1024×256. Otherwise the same as Figure 5.
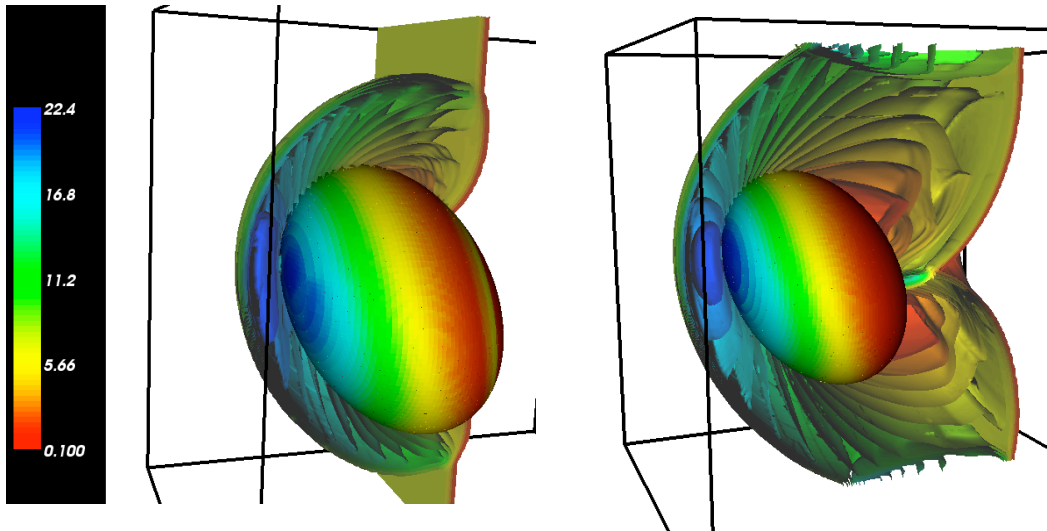
Fig. 7. Propagation of a Mach 10 shock over an ellipsoidal body – density isosurfaces, and density on the ellipsoid. Left: early time; right: late time. The color map of the density values is given on the left of the figure.

Finally, we show a computation in three dimensions of a planar shock reflecting off an ellipsoidal object (figure 7). This calculation demonstrates the ability of the three-dimensional algorithm to compute with strong discontinuities intersecting the irregular boundary.

7 Conclusions

We presented here a new Cartesian grid embedded boundary algorithm for systems of conservation laws, generalizing the unsplit second-order Godunov method described in [7,14]. It is formally consistent, with a truncation error that vanishes as the mesh spacing goes to zero, leading to a method that is at least second-order accurate in L^1, and first-order accurate in L^∞. On standard strong-shock test problems, it is robust, leading to results that are nearly indistinguishable from corresponding grid-aligned calculations. Using the ideas in [11], we have developed an adaptive mesh refinement version of this algorithm. With appropriate software support, the present algorithm

can be used to compute solutions to a broad range of problems in complex geometries.

There are several ways in which the algorithmic ideas described here can be generalized. One is to revisit the hyperbolic free-boundary algorithms in [5,2,12,10]. Such an approach could potentially lead to more accurate free-boundary methods: for the case of non-characteristic boundaries, such as shocks and flame fronts, we would expect uniformly second-order accurate methods in the absence of secondary captured discontinuities. A second application would be to combine this algorithm with the finite-volume algorithms for elliptic and parabolic problems in [8,9,15] to solve incompressible or low-Mach number flow problems with irregular fixed or free boundaries, following the ideas in [1].

Variable	Coarse Error	Fine Error	Order
mass-density	3.538533e-08	8.544086e-09	2.050153e+00
x-momentum	6.075919e-08	1.965511e-08	1.628199e+00
y-momentum	2.507996e-08	7.694576e-09	1.704621e+00
z-momentum	2.507996e-08	7.694576e-09	1.704621e+00
energy-density	1.238544e-07	2.990567e-08	2.050154e+00

Table 12
Solution error convergence rates using L^2 norm. $h_f = \frac{1}{256}$ and $h_c = 2h_f$, $D = 3$

References

[1] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.

[2] J.B. Bell, P. Colella, and M.L. Welcome. Conservative front-tracking for inviscid compressible flow. In *AIAA 10th Computational Fluid Dynamics Conference. Honolulu*, pages 814–822, 1991.

[3] M. Berger, C. Helzel, and R. LeVeque. H-box methods for the approximation of hyperbolic conservation laws on irregular grids. *SIAM Journal of Numerical Analysis*, 41:893–918, 2003.

[4] M. J. Berger and R. J. Leveque. Stable boundary conditions for cartesian grid calculations. Technical Report 90-37, ICASE, May 1990.

[5] I. L. Chern and P. Colella. A conservative front-tracking method for hyperbolic conservation laws. Technical Report UCRL-97200, Lawrence Livermore National Laboratory, 1987.

[6] W. J. Coirier and K. G. Powell. An assessment of cartesian-mesh approaches for the euler equations. *Journal of Computational Physics*, 117:121–131, 1995.

[7] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[8] H. S. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 147(2):60–85, December 1998.

[9] P. McCorquodale, P. Colella, and H. Johansen. A Cartesian grid embedded boundary method for the heat equation on irregular domains. *J. Comput. Phys.*, 173(2):620–635, November 2001.

[10] G.H. Miller and P. Colella. A conservative three-dimensional eulerian method for coupled solid-fluid shock capturing. *Journal of Computational Physics*, 183(1):26–82, Nov. 2002.

[11] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *J. Comput. Phys.*, 120(2):278–304, September 1995.

[12] J. E. Pilliod. *A Second-Order Unsplit Method for Modeling Flames in Two-Dimensional Compressible Flow*. Ph. D. Thesis, Department of Mathematics, University of California, Davis, September 1996.

[13] J. J. Quirk. An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies. *Computers and Fluids*, 23:125–142, 1994.

[14] J. S. Saltzman. An unsplit 3d upwind method for hyperbolic conservation laws. *J. Comput. Phys.*, 115:153–168, 1994.

[15] P. O. Schwartz, M. Barad, P. Colella, and T. J. Ligocki. A Cartesian grid embedded boundary method for Poisson's equation and the heat equation in three dimensions. Submitted for publication, 2004.

[16] P.R. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–174, 1984.