

Lawrence Berkeley National Laboratory

Lawrence Berkeley National Laboratory

Title

An unsplit, cell-centered Godunov method for ideal MHD

Permalink

<https://escholarship.org/uc/item/1cd3f1pr>

Authors

Crockett, Robert K.

Colella, Phillip

Fisher, Robert T.

et al.

Publication Date

2003-08-29

Peer reviewed

An Unsplit, Cell-Centered Godunov Method for Ideal MHD

Robert K. Crockett^{a,*} Phillip Colella^c Robert T. Fisher^d
Richard I. Klein^{b,d} Christopher F. McKee^{a,b}

^a*Physics Department, University of California, Berkeley, CA 94720, USA*

^b*Astronomy Department, University of California, Berkeley, CA 94720, USA*

^c*Lawrence Berkeley National Laboratory, Berkeley CA, 94720, USA*

^d*Lawrence Livermore National Laboratory, Livermore CA, 94551, USA*

Received 27 October 2003; accepted ***

Abstract

We present a second-order Godunov algorithm for multidimensional, ideal MHD. Our algorithm is based on the unsplit formulation of Colella (J. Comput. Phys. vol. 87, 1990), with all of the primary dependent variables centered at the same location. To properly represent the divergence-free condition of the magnetic fields, we apply a discrete projection to the intermediate values of the field at cell faces, and apply a filter to the primary dependent variables at the end of each time step. We test the method against a suite of linear and nonlinear tests to ascertain accuracy and stability of the scheme under a variety of conditions. The test suite includes rotated planar linear waves, MHD shock tube problems, low-beta flux tubes, and a magnetized rotor problem. For all of these cases, we observe that the algorithm is second-order accurate for smooth solutions, converges to the correct weak solution for problems involving shocks, and exhibits no evidence of instability or loss of accuracy due to the possible presence of non-solenoidal fields.

Key words: magnetohydrodynamics, numerical approximation, stability and convergence of difference schemes,

PACS:

* Corresponding author.

Email address: rcrockett@astron.berkeley.edu (Robert K. Crockett).

URL: astron.berkeley.edu/~mookie/ (Robert K. Crockett).

1 Introduction

In this paper we present a new Godunov method for the equations of multidimensional ideal magnetohydrodynamics (MHD). We give results from an implementation of the unsplit, second-order method of Colella [9] for these equations. The base scheme solves the ideal MHD equations using a second-order predictor-corrector formalism. To the base scheme we add three algorithmic components, whose effects upon accuracy and stability are measured. The first component is a MAC projection [5,18] step, which uses a Poisson solver to ensure that the cell-edge centered fields used to calculate fluxes are divergence-free to machine precision. This component, though commonplace in the context of incompressible Navier-Stokes simulation, is new to the MHD community. The second component is an approximate projection [1] that uses another solution of the Poisson equation to ensure that the cell-centered field is divergence-free to second-order. The last component is a filter [23] that also acts to suppress monopole sources in the cell-centered field. The combination of MAC and approximate projections along with filtering techniques to enforce the divergence-free constraint, $\nabla \cdot \vec{B} = 0$, is novel in the context of ideal MHD simulations.

The section that follows covers recent work and some of the schemes used for ideal MHD simulation. It also introduces methods for enforcing the divergence-free constraint, and introduces some issues surrounding multidimensional MHD. Section 3 introduces our basic algorithm and the extensions we have implemented. A suite of linear and nonlinear test problems will be used to determine which of our algorithmic extensions are best suited to each problem type. These tests and results are covered in Section 4. The overall purpose is to find one combination of these extensions which is well suited to all of the problems considered. This will be done through comparisons to published results and in some cases to a divergence-wave (or eight-wave) MHD algorithm we have implemented.

2 Background

The study of numerical algorithms for magnetohydrodynamics simulations remains an active one, with no one method having become the standard. Two generic algorithms are the most widely used at present: the Method of Characteristics/Constrained Transport (MOC/CT; common in the astrophysics community) [14,31] and shock-capturing (Godunov) methods [4,8,10,11,17,28,34,37]. Each has distinct benefits and drawbacks. Codes implementing the MOC/CT algorithm are relatively simple in design, and satisfy the divergence-free constraint to machine precision. However, the method of characteristics used by

the ZEUS scheme, as outlined in [31], is by construction second-order on Alfvén and advective waves, but does not address the two compressive waves of ideal MHD. Moreover, Falle [15] found that ZEUS exhibits spurious rarefaction shocks in certain 1-D MHD shock tubes for a non-isothermal equation of state. Codes implementing the shock-capturing algorithm on the other hand, while more complex, give highly accurate results even for strong shocks. They suffer from the drawback that the divergence-free constraint is only satisfied to truncation error, which can be large in the region of large jumps. In order to treat this difficulty, a variety of techniques have come into use. One such is the hybrid CT/shock-capturing scheme [4,12,22,29,34], for which the constraint is satisfied by design like in the MOC/CT case. The cost to the accuracy of the underlying shock-capturing scheme is unclear. Another approach, originally due to Brackbill and Barnes [7] and implemented by workers such as Ryu et al [28], uses a divergence cleaning step on the cell-centered fields to enforce the constraint. In a third approach, Powell and co-workers [17,25] use a divergence-wave reformulation of the ideal MHD equations originally due to Godunov [16]. Tóth [34] (hereafter T00) implements all three types of schemes, among others, using them as the basis for a comparison on a variety of 1-D and 2-D tests. More recently, Dedner et al [13] compare several hyperbolic schemes with additional waves and divergence-damping terms on the T00 tests.

The 2-D tests of T00 serve to underscore the importance of using multidimensional problems in evaluating different algorithms, since it was mainly in this context that differences between them became apparent. This is to be expected, since errors due to non-solenoidal fields will generally only show up for problems in two or more dimensions. Many shock-capturing MHD schemes use an operator-split (or dimensionally-split) formalism to treat multidimensional ideal MHD. This means that, for each spatial dimension of the scheme, the one dimensional MHD equations are applied once. Unsplit schemes, which instead use the full multidimensional version of the equations, have been implemented and shown to give results equivalent to those of split methods for hydrodynamical problems [9]. Unsplit shock-capturing schemes for multidimensional ideal MHD are relatively new, however. One of our main goals is to assess the efficacy of different approaches for enforcing the divergence-free constraint in one such unsplit scheme.

2.1 The Divergence-free Constraint

Some means must be employed to ensure that the field satisfies the divergence-free constraint, since this is only guaranteed to within truncation error in shock-capturing schemes. In the region of large jumps in the field, these truncation errors can lead to $\mathcal{O}(1)$ solution errors and incorrect dynamics. While large errors in any quantity can occur in shocks, in the case of non-solenoidal

fields they can lead to instability. A dramatic example of even small errors in the divergence-free condition leading to instability is given in Section 4.1. There, small amplitude, non-propagating waves become unstable unless non-solenoidal fields are suppressed. The possibility of incorrect field topologies, incorrect dynamics, and numerical instability motivate efforts to formulate and understand different means of suppressing non-solenoidal errors in the magnetic field.

The eight-wave MHD algorithm, as implemented by Powell et al [17,25] and others, addresses the problem by adding additional terms corresponding to monopoles to the ideal MHD equations. The resultant equations are symmetrizable, so that they are Galilean invariant and transport $\nabla \cdot \vec{B}$ [16]. The additional terms show up in two ways for shock-capturing schemes. Since they modify the 1-D MHD equations used for characteristic tracing to include an additional eighth wave that travels at the flow speed, monopoles will be advected along with the flow. Such monopoles could be carried out of the domain, or they might build up at a stagnation point. Secondly, the additional terms appear as source terms, making the system non-conservative if the divergence-free condition isn't already satisfied.

Dedner et al [13] test a scheme that extends the divergence-wave concept to damp and advect monopole sources, even at stagnation points in the flow. This is done through the magnetic analog of an artificial compressibility term, an approach that surfaced earlier in the context of the Maxwell equations [23]. To the extent that the computed auxiliary field remains continuous, the scheme will remain divergence free. Any monopole sources will be advected at the fastest speed allowed under the Courant condition, and damped as they are advected. This method is very useful on unstructured grids, where solving the Poisson equation in order to project out the solenoidal component of the field is difficult.

Divergence cleaning, or Hodge projection, in shock-capturing schemes can address the problem of non-solenoidal fields in two ways. The most widely discussed [2,7,28,34] involves projection of the cell-centered field onto the space of divergence-free fields. Projecting in this manner with a centered difference approximation to the divergence is consistent with the underlying cell-centered scheme. One is left with fields at the advanced time which are divergence-free to machine accuracy. Such a projection has been found to give correct field topologies in shock tube problems [28]. On the other hand, choosing to eliminate the divergence as measured in one metric does not guarantee that unphysical effects are not entering into the dynamics. An illustrative example from incompressible flow [21] where an analogous constraint on the fluid velocity, $\nabla \cdot \vec{v} = 0$, occurs shows that checkerboard modes in the velocity can cause instabilities when using a centered difference to approximate the divergence. Such modes must be damped by a suitably chosen filter in order to regain

stability.

Another option for cleaning of non-solenoidal fields is to project the fields at cell-edges, which are then used to calculate fluxes. In a MAC discretization [5,18], backward difference approximation to the divergence and a five-point finite difference Laplacian are used to enforce the divergence-free constraint to within machine precision. We use a multigrid solver to solve the associated Poisson equation. The MAC projection has the advantage that it does not affect the conservation properties of the scheme. However, the extent to which such a projection is required in magnetohydrodynamic problems has remained unclear to this point.

2.2 Multidimensional MHD

The MHD equations in two or more dimensions are decidedly more complex to solve than in one dimension. In MHD simulations with variations along the x-axis alone, there is no change in the field along the x-axis. The divergence-free constraint is therefore trivially satisfied. Obtaining the solution to the Riemann problem upon which Godunov methods are based is also relatively straightforward in 1-D.

In multiple dimensions, we are solving the full equations of ideal MHD, which in conservation form are

$$\partial_t \rho + \nabla \cdot (\rho \vec{u}) = 0 \quad (1)$$

$$\partial_t(\rho \vec{u}) + \nabla \cdot \left[\rho \vec{u} \vec{u} + \left(P + \frac{B^2}{8\pi} \right) \mathbf{I} - \frac{1}{4\pi} \vec{B} \vec{B} \right] = 0 \quad (2)$$

$$\partial_t(\vec{B}) + \nabla \cdot [\vec{u} \vec{B} - \vec{B} \vec{u}] = 0 \quad (3)$$

$$\partial_t(\rho E) + \nabla \cdot \left[\left(\rho E + P + \frac{1}{8\pi} B^2 \right) \vec{u} - \frac{1}{4\pi} (\vec{u} \cdot \vec{B}) \vec{B} \right] = 0, \quad (4)$$

subject to the constraint $\nabla \cdot \vec{B} = 0$. Here ρ is the mass density, $\rho \vec{u}$ the momentum density, \vec{B} the magnetic field, and $\rho E = \frac{1}{2} \rho |\vec{u}|^2 + \frac{1}{8\pi} |\vec{B}|^2 + \frac{1}{\gamma-1} P$ the total energy density. The ∂_t notation denotes derivatives with respect to time. In 2-D MHD, for example, the full x-momentum equation is

$$\partial_t(\rho u) + \partial_x \left(\rho u^2 + P + \frac{-B_x^2 + B_y^2}{8\pi} \right) + \partial_y \left(\rho uv - \frac{B_x B_y}{4\pi} \right) = 0, \quad (5)$$

For the sake of simplicity, most predictor-corrector schemes for solving the multidimensional MHD equations use a dimensional-splitting approach for predicting the states at cell-edges. In this approach, the 1-D MHD equations

are solved once for each direction. The 1-D equations applied to update the x-momentum in such a split predictor for 2-D MHD are:

$$\partial_t(\rho u) + \partial_x \left(\rho u^2 + P + \frac{B_y^2}{8\pi} \right) = 0 \quad (6)$$

$$\partial_t(\rho u) + \partial_y(\rho uv) - B_y \partial_y \left(\frac{B_x}{4\pi} \right) = 0. \quad (7)$$

Notice that terms involving derivatives of the normal field, like $-\frac{1}{4\pi} B_x \partial_x B_x$ in equation 5, are absent in equations 6 and 7 since the 1-D MHD equations exclude the possibility of jumps in the normal field. As a result, one would expect decreased accuracy on multidimensional problems from a code implementing such an algorithm. A related problem arises in algorithms utilizing Strang splitting, where due to the successive application of 1-D operators, a correct accounting for jumps in the normal field is unlikely [22]. This shortcoming is absent in an unsplit method since, properly implemented, it uses the full 2-D equation (5) in one step, as opposed to consecutive (in time) updates in different directions.

3 Equations and Algorithms

In general for hyperbolic conservation laws, the conserved variables U evolve according to $\partial_t U + \nabla \cdot \vec{F}(U) = 0$. Our code uses a second-order Godunov-type method for hyperbolic conservation laws [6]. Such volume-average schemes follow the flux of conserved quantities such as momentum into and out of each cell comprising the computational domain. Quantities are stored at cell-centers, their value at this point being the volume-average over the entire cell. During the course of a timestep, the flux of the conserved quantities at each edge of each cell is computed. Differencing these fluxes gives the update of the conserved quantity to the next time (see Figure 1).

More specifically, our code is based on the unsplit MHD algorithm of Colella [9]. It ostensibly tracks the evolution of the conserved density ρ , three components of momentum $\rho \vec{u}$, three components of magnetic field \vec{B} , and total energy density ρE , in two spatial dimensions. These variables would then be evolved according to the ideal MHD equations (1-4). However, for simplicity and accuracy a set of primitive variables W consisting of the density, velocity, field, and pressure are evolved in time:

$$W = [\rho \ u \ v \ w \ B_x \ B_y \ B_z \ P]. \quad (8)$$

We will switch freely between U and W here depending on which is most

convenient; the indices will indicate the centering of the variables. Thus i, j will denote cell-centered states, i, j, \pm forward- and backward-interpolations to cell-edges, and $i + \frac{1}{2}, j$ and $i, j + \frac{1}{2}$ (Riemann) states at the edges.

The fundamental aspects of our scheme are as follows. Given the state $W_{i,j}^n$ at time n and spatial coordinate (x_i, y_j) , we simultaneously interpolate in space and extrapolate in time to obtain the states $W_{i,j,\pm,x}^{n+\frac{1}{2}}$ on the two x-boundaries of each cell (i, j) . The same is done for the states at the y-boundaries, $W_{i,j,\pm,y}^{n+\frac{1}{2}}$. All this is done in the normal and transverse predictor steps of the algorithm. Next, the Riemann problem is solved using the $\hat{W}_{i,j,\pm,x}^{n+\frac{1}{2}}$ states. These Riemann states are used to calculate fluxes at each edge of a cell in the corrector step. The fluxes are then differenced to give the update to the next timestep, $W_{i,j}^{n+1}$. We go into more detail on these steps of our scheme in Sections 3.1 through 3.4.

The additional parts of our MHD scheme consist of additional terms in the normal predictor step to ensure correct multidimensional behavior, and several steps that address the divergence-free constraint. These are also discussed in Sections 3.1 through 3.4.

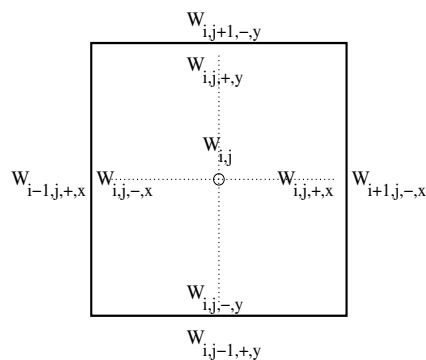


Fig. 1. Single cell of the computational domain, showing interpolated states $W_{i,j,\pm,x}^{n+\frac{1}{2}}$ and $W_{i,j,\pm,y}^{n+\frac{1}{2}}$, and cell-centered state $W_{i,j}$. The (Riemann) states at cell edges would lie between the interpolated states. So $W_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$, for instance, would lie between $W_{i,j,+x}^{n+\frac{1}{2}}$ and $W_{i+1,j,-x}^{n+\frac{1}{2}}$.

3.1 Normal Predictor

The predictor computes $W_{i,j,\pm,x}^{n+\frac{1}{2}}$ and $W_{i,j,\pm,y}^{n+\frac{1}{2}}$ using a Taylor series expansion:

$$W_{i,j,\pm,x}^{n+\frac{1}{2}} = W_{i,j}^n \pm \frac{\Delta x}{2} \partial_x W - \frac{\Delta t}{2} \mathbf{A}^x \partial_x W - \frac{\Delta t}{2} \frac{\partial U}{\partial W} \partial_y F^y. \quad (9)$$

Here the matrix \mathbf{A} is related to the flux F^x of of the conserved variable U by $\mathbf{A} = (\partial_U W)(\partial_W F^x)$. We give formulae for $W_{i,j,\pm,x}^{n+\frac{1}{2}}$, those for $W_{i,j,\pm,y}^{n+\frac{1}{2}}$ being similar. The first three terms on the right-hand side are computed in the normal predictor, and we label this intermediate result $\hat{W}_{i,j,\pm,x}^{n+\frac{1}{2}}$. We now separate out the evolution of the normal field $B_n = B_x$ through the following notation:

$$\hat{W} = \begin{bmatrix} \tilde{W} \\ B_x \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \tilde{\mathbf{A}} & a_B \\ 0 & 0 \end{bmatrix}. \quad (10)$$

The matrix $\tilde{\mathbf{A}}$ corresponds to the usual 1-D MHD equations, with its seven characteristics: forward- and backward-propagating fast, slow, and Alfvén, plus the advective wave. Second-order accuracy is achieved in part through the use of characteristic analysis to calculate second-order accurate derivatives $\partial_x W$ in the spatial interpolation. This characteristic interpolation is based on calculation of the eigenvalues λ_k and left- and right-eigenvectors l_k and r_k of the matrix $\tilde{\mathbf{A}}$, giving the following expression for the interpolation of the \tilde{W} variables to cell-edges:

$$\tilde{W}_{i,j,\pm,x}^{n+\frac{1}{2}} = \tilde{W}_{i,j}^n + \frac{1}{2} \sum_{k:\lambda_k \gtrless 0} \left(\pm 1 - \frac{\Delta t}{\Delta x} \lambda_k \right) \alpha_k r_k \quad (11)$$

$$\alpha_k = \begin{cases} \text{Min}(\alpha^0, \alpha^+, \alpha^-) & \text{if } \alpha^+ \alpha^- \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$\alpha^0 = \frac{1}{2} l_k \cdot (W_{i+1,j}^n - W_{i-1,j}^n) \quad (13)$$

$$\alpha^+ = 2 l_k \cdot (W_{i+1,j}^n - W_{i,j}^n) \quad (14)$$

$$\alpha^- = 2 l_k \cdot (W_{i,j}^n - W_{i-1,j}^n) \quad (15)$$

The α_k represent the strength of the k^{th} wave in the interpolant. The sum over $\lambda_k < 0$ would correspond to backward-propagating waves used in the interpolation to $\tilde{W}_{i,j,-,x}^{n+\frac{1}{2}}$, and similarly for $\lambda_k > 0$ and $\tilde{W}_{i,j,+,x}^{n+\frac{1}{2}}$.

A full accounting for all x-derivative terms in the 2-D MHD equations shows that a_B is given by

$$a_B = - \left[0, \frac{B_x}{4\pi\rho}, \frac{B_y}{4\pi\rho}, \frac{B_z}{4\pi\rho}, v, w, \frac{\vec{u} \cdot \vec{B}}{4\pi} \right]^T. \quad (16)$$

These terms are essential to the second-order accuracy of the scheme, in particular on multidimensional problems such as waves not propagating along cardinal axes. They are incorporated into our algorithm through a simple finite differencing of the normal derivative of the normal field, $\partial_n B_n = \partial_x B_x$. The terms are added to those already present due to the characteristics-based interpolation:

$$\tilde{W}_{i,j,\pm,x}^{n+\frac{1}{2}} := \tilde{W}_{i,j,\pm,x}^{n+\frac{1}{2}} - \frac{\Delta t}{2} a_B (\mathcal{D}_x^0 B_x^n)_{i,j}. \quad (17)$$

The $(\mathcal{D}_x^0 B_x^n)_{i,j} = ((B_x)_{i+1,j}^n - (B_x)_{i-1,j}^n) / (2\Delta x)$ correction term is the centered-difference approximation to $\partial_x B_x$. Note that this approximation to the derivative is not limited. The need for such correction terms in an unsplit scheme for multidimensional MHD was not addressed in C90. It was first noted by Stone [32] during an examination of the accuracy of an unsplit Godunov scheme on advected flux rings.

3.2 Transverse Predictor

The last term in the evolution equation (9) is included via the transverse predictor. The basic idea is to approximate transverse derivatives (in this case in the y-direction) using a 1-D Godunov method. We take the states calculated in the normal predictor, $\hat{W}_{i,j,\pm,y}^{n+\frac{1}{2}}$, and first use them to solve the Riemann problem at each y-boundary in the domain. The resulting Riemann states $U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$ are subsequently used to calculate the fluxes needed for the last term in equation 9. In more formal terms,

$$U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \mathcal{R} \left(\hat{W}_{i,j,+,y}^{n+\frac{1}{2}}, \hat{W}_{i,j+1,-,y}^{n+\frac{1}{2}} \right) \quad (18)$$

$$F_{i,j+\frac{1}{2}}^{y,n+\frac{1}{2}} = F^y \left(U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \right). \quad (19)$$

Here $\mathcal{R}(\cdot, \cdot)$ denotes the Riemann problem solution using the two states on either side of an edge as input; see Section 3.4. It is then a straightforward

matter to use these fluxes to calculate a finite-difference approximation to $\partial_y F^y$, and thereby complete the calculation of the edge-centered states.

3.3 Corrector

The corrector first calculates fluxes at all cell-edges using another Riemann problem solve. At this stage, we have the first-order accurate approximation to the interpolated states $W_{i,j,\pm}^{n+\frac{1}{2}}$ from the predictor in hand. The Riemann solver takes these states and returns a single state for each cell-edge, $U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ and $U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$. For instance,

$$U_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \mathcal{R} \left(W_{i,j,+,x}^{n+\frac{1}{2}}, W_{i+1,j,-,x}^{n+\frac{1}{2}} \right). \quad (20)$$

The formula for $U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$ is similar. See Section 3.4 for more details on the Riemann problem solution.

The states at cell-edges are not guaranteed to be divergence-free. We modify the C90 algorithm to enforce the divergence-free condition for the Riemann problem states. These Riemann states have a non-solenoidal component that we treat through a MAC projection, earlier used in the context of incompressible fluid computations [5]. The edge-centered fields $(\vec{B}^*)_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ and $(\vec{B}^*)_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$ are used to calculate a cell-centered monopole charge density $q_M = \nabla \cdot \vec{B}^*$, and a Poisson solver is in turn used to find the scalar field ϕ implied by this monopole charge density distribution. The scalar field satisfies the following relations, in which \mathcal{D}_x^\pm correspond to the forward- and backward-difference approximations to the derivative $\frac{\partial}{\partial x}$, and similarly for \mathcal{D}_y^\pm :

$$(q_M)_{i,j} = \mathcal{D}_x^-(B_x^*)_{i+\frac{1}{2},j} + \mathcal{D}_y^-(B_y^*)_{i,j+\frac{1}{2}} \quad (21)$$

$$[\mathcal{D}_x^+ \mathcal{D}_x^- + \mathcal{D}_y^+ \mathcal{D}_y^-] \phi_{i,j} = (q_M)_{i,j}. \quad (22)$$

$$(23)$$

The correction to the field is calculated from ϕ as follows:

$$(B_x)_{i+\frac{1}{2},j} = (B_x^*)_{i+\frac{1}{2},j} - \mathcal{D}_x^+ \phi_{i,j} \quad (24)$$

$$(B_y)_{i+\frac{1}{2},j} = (B_y^*)_{i+\frac{1}{2},j} - \frac{1}{2} [\mathcal{D}_y^0 \phi_{i+1,j} + \mathcal{D}_y^0 \phi_{i,j}] \quad (25)$$

$$(B_x)_{i,j+\frac{1}{2}} = (B_x^*)_{i,j+\frac{1}{2}} - \frac{1}{2} [\mathcal{D}_x^0 \phi_{i,j+1} + \mathcal{D}_x^0 \phi_{i,j}] \quad (26)$$

$$(B_y)_{i,j+\frac{1}{2}} = (B_y^*)_{i,j+\frac{1}{2}} - \mathcal{D}_y^+ \phi_{i,j} \quad (27)$$

With this correction to the magnetic field of the Riemann states, the L^1 norm of the MAC monopole density is reduced from its initial value by a user-settable multiplicative factor, in our case 10^{-12} .

The algorithm now proceeds to calculate the fluxes associated with the Riemann states. These fluxes are then differenced to give the update to the next time $U_{i,j}^{n+1}$:

$$F_{i+\frac{1}{2},j}^{x,n+\frac{1}{2}} = F^x \left(U_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right) \quad (28)$$

$$U_{i,j}^{n+1} = U_{i,j}^n - \Delta t \mathcal{D}_x^- F_{i+\frac{1}{2},j}^{x,n+\frac{1}{2}} - \Delta t \mathcal{D}_y^- F_{i,j+\frac{1}{2}}^{y,n+\frac{1}{2}} \quad (29)$$

After the update to t^{n+1} , we are left with a cell-centered field $\vec{B}_{i,j}^{*,n+1}$ that is no longer divergence-free by a centered-difference divergence metric. To what extent, if any, this is a problem depends on the physical problem being considered, and will be addressed later.

For those cases where a reduction of the divergence is required, two algorithmic extensions have been implemented. The first follows from noting that it is desirable to have a diffusive term of the form (see [23,13]):

$$\frac{\partial(\nabla \cdot \vec{B})}{\partial t} = \eta \nabla^2 (\nabla \cdot \vec{B}) \quad (30)$$

act on the divergence of \vec{B} . This may be rewritten to eliminate a spatial derivative by pulling out a divergence operator, giving

$$\frac{\partial \vec{B}}{\partial t} = \eta \nabla (\nabla \cdot \vec{B}) \quad (31)$$

A simple, single-step filter may be derived [24] as a finite-difference approximation of equation 31:

$$B_x := B_x + \eta \Delta t (\mathcal{D}_x^+ \mathcal{D}_x^- B_x + \mathcal{D}_x^0 \mathcal{D}_y^0 B_y) \quad (32)$$

$$B_y := B_y + \eta \Delta t (\mathcal{D}_x^0 \mathcal{D}_y^0 B_x + \mathcal{D}_y^+ \mathcal{D}_y^- B_y) \quad (33)$$

In order to choose a value for η we use Fourier stability analysis, giving a stability condition for the scheme in equations 32 and 33 of $\Delta t \leq \frac{2(\Delta x)^2}{5\eta}$. Since Δt is set by the Courant condition, we are able to derive a condition on η , giving the maximum amount of diffusion of the monopole sources possible, given our timestep and grid spacing:

$$\eta = C \frac{(\Delta x)^2}{\Delta t} \quad (34)$$

with $C \leq \frac{2}{5}$. A stronger condition, $C \leq \frac{1}{5}$, will always damp monopole modes. This formulation will both decrease the cell-centered divergence and damp checkerboard modes. The filtering is always used when an approximate projection is performed.

The numerical effect of the filter can be modified through the parameter C . We found, through the nonlinear tests outlined in Section 4, that values in the range $10^{-2} - 10^{-1}$ worked best. We note here that we chose to apply the filter to the conserved variables U^{n+1} , so that the changed magnetic field causes no change in the total energy. Any addition (subtraction) of magnetic energy shows up as a decrease (increase) in the internal energy. If it is applied after the updated conserved variable state is converted back to primitive variables, then a change in total energy results and the internal energy is not affected.

The second means for treating non-solenoidal cell-centered fields is an approximate projection, in which the cell-centered divergence is used to calculate a monopole charge density $q_M = \mathcal{D}_x^0 B_x^* + \mathcal{D}_y^0 B_y^*$. Note that the multigrid Poisson solver uses the finite-difference operator $\mathcal{D}^+ \mathcal{D}^-$, not $\mathcal{D}^0 \mathcal{D}^0$, for the Laplacian ∇^2 . The resulting solution is not exact, but instead second-order accurate [1]. Exact Poisson solvers are another option. However, the extension of exact solvers to adaptive meshed is extremely complicated [19]. Furthermore, when using a centered-difference approximation to the divergence, Poisson solvers in general suffer from growth in checkerboard modes [21] in the field that, though divergence-free, are unphysical. An extensive analysis of this phenomenon is given in [26].

The Poisson solve yields the scalar field ϕ satisfying equation 22. We difference ϕ as follows to give the corrected field:

$$B_x^{n+1} = B_x^{*,n+1} - \mathcal{D}_x^0 \phi_{i,j} \quad (35)$$

$$B_y^{n+1} = B_y^{*,n+1} - \mathcal{D}_y^0 \phi_{i,j} \quad (36)$$

We find in our tests that an approximate projection does not have a significant impact on the quality of our results when used in conjunction with a MAC projection. Given the significant computational expense of using both, we advise against such a scheme.

3.4 Riemann Solver

We solve the Riemann problem for ideal MHD using a linearized solver [33]. It employs characteristic analysis, like that of the normal predictor of Section 3.1, to solve for the state at cell edges. The eigenvalues and eigenvectors required for the characteristic analysis are sensitive to the input state; see Roe and Balsara [27]. Additionally, eigenvalue (ie. wavespeed) degeneracies can cause problems when the flow speed is close to zero. Errors due to degeneracies in the wavespeeds can build up in this case, causing numerical instability. We add a simple degeneracy fix to address this issue. (A similar issue occurs in the normal predictor, where problems can arise in the upwinding procedure used to preserve monotonicity. The same fix is used to address the problem there, as well.) Eigenvalues are modified when they are the same to within a chosen tolerance. So if the Alfvén speed $c_{A,x}$ is less than a small parameter, ϵ_P , times the fast speed c_F , then we make the forward/backward Alfvén and forward/backward slow characteristic speeds, $\lambda_{\pm A}$ and $\lambda_{\pm S}$, equal to the advective speed λ_0 : $\lambda_{+A} = \lambda_{-A} = \lambda_{+S} = \lambda_{-S} = \lambda_0$. Similarly, if the slow speed c_S is less than ϵ_P times c_F , then we take the forward/backward slow characteristic speeds equal to the advective one: $\lambda_{+S} = \lambda_{-S} = \lambda_0$. This modification occurs only when the advective speed is small: $\lambda_0 < 10^{-6}c_F$.

The first condition handles degeneracy between the Alfvén and advective waves. This occurs when the field in the direction of interpolation is small. In this case the Alfvén and slow characteristic speeds are set to the advective speed. The second condition handles degeneracy between the slow and advective wave, and in that case only the slow characteristic speed is set to the advective speed. Based on experiments with degenerate and nearly-degenerate input states, we choose $\epsilon_P = 0.01$ so that the errors introduced are minimized while still preventing spurious wave strengths.

For strongly nonlinear problems, it sometimes happens that the CFL condition is not sufficient to keep a scheme stable. If accelerations are large, velocities can grow such that in one timestep the pressure or density becomes negative. In order to dynamically adjust to such situations, we implement a scheme that checks for negative pressures or densities in the cell-centered states, $W_{i,j}^{n+1}$, after the corrector step. If one is encountered, that timestep is restarted with the CFL number lowered by a factor of two from its nominal value, down to a minimum of one-eighth of the initial CFL number. Once the code has proceeded for several timesteps without again encountering negative values, the CFL number is raised by a factor of two, eventually reaching its initial value. In practice, negative values are encountered only rarely, so that the average CFL number over the course of a run is close to the nominal value. Note that even if the CFL ramps quickly back up to the nominal value, the timestep is never allowed to increase by more than 10% in one iteration. In

this way, the effect of an increase in CFL number is in fact spread over several iterations.

4 Numerical Tests

In this section, we compare the behavior of the code on a variety of linear and nonlinear problems. Both seven- and eight-wave MHD codes were used. The code implementing the eight-wave MHD algorithm also uses a predictor-corrector formalism. The characteristic analysis performed in the predictor steps uses an eighth wave carrying changes in the normal field at the advection velocity, in addition to the seven waves of ideal MHD. As a result, in equation 10 of Section 3.1, the (2, 2) entry of \mathbf{A} is equal to the advective speed u and not zero. The terms needed for second-order accuracy in ideal MHD (equation 16) are accounted for in this case, so that $a_B = 0$. The eight-wave algorithm also adds a source term,

$$S = - \left[0, \frac{B_x}{4\pi}, \frac{B_y}{4\pi}, \frac{B_z}{4\pi}, u, v, w, \frac{\vec{u} \cdot \vec{B}}{4\pi} \right]^T (\nabla \cdot \vec{B}), \quad (37)$$

to the right-hand side of the MHD equations 1-4. The source term is calculated in the transverse predictor and corrector steps. It is added to the updated states along with the differenced fluxes. We note, however, that performing a MAC projection guarantees that $\nabla \cdot \vec{B} = 0$ for the edge-centered states, and as a result the source term calculated in the corrector is numerically zero in this case. This implies that only with a MAC projection is the scheme conservative and satisfies the jump relations. The source term in the transverse predictor, while generally small, is not numerically zero.

For both MHD implementations, different variations on the base algorithm were tested. In what follows, codes with conservative filtering are labeled with 'CF'. An 'AP' denotes codes with an approximate projection; it is always accompanied by conservative filtering, in order to suppress checkerboard modes. Those codes utilizing a MAC projection are labeled with 'MAC'. So, for example, a MAC projection code with conservative filter is labeled 'MAC+CF'.

4.1 Simple Linearized MHD Waves

We tested the performance of the code on all four varieties (advective, fast, slow, and Alfvén) of linearized MHD waves. These tests comprised waves propagating along x- and y-axes (wavenumbers $\vec{n} = (1, 0)$ and $\vec{n} = (0, 1)$), along

with waves at slopes of 1:1 ($\vec{n} = (1, 1)$) and 2:1 ($\vec{n} = (2, 1)$). The simulation domain had length $L = 1$ in both dimensions, and the boundaries in all cases were periodic. The Alfvén waves used $\rho_0 = 1$, $\vec{u}_0 = 0$, $\vec{B}_0 = B_0 \hat{b} = \sqrt{4\pi} \hat{b}$ with unit vector $\hat{b} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, and $P_0 = 1$. The perturbation δW is

$$\delta W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -c_A \\ 0 \\ 0 \\ B_0 \\ 0 \end{bmatrix} \delta_{\text{pert}} \sin(\vec{k} \cdot \vec{x}), \quad \text{with } \vec{k} = 2\pi \vec{n}, \quad (38)$$

where \vec{n} is the aforementioned vector of integers chosen so as to be consistent with the periodic boundaries. The Alfvén speed $c_A = B_0/\sqrt{4\pi\rho_0} = 1$.

Fast and slow wave expressions are somewhat more complicated. In this case, \hat{b} lies at 45 degrees from the unit wavevector \hat{k} , all other aspects of the unperturbed state remaining unchanged from the Alfvén case. The perturbation is

$$\delta W = \begin{bmatrix} \rho_0 \\ \frac{\sqrt{2}c_{F/S}^2 \hat{b}_y - a^2 \hat{n}_y}{c_{F/S}} \\ \frac{a^2 \hat{n}_x - \sqrt{2}c_{F/S}^2 \hat{b}_x}{c_{F/S}} \\ 0 \\ -\sqrt{2}B_0 \frac{c_{F/S}^2 - a^2}{c_A^2} \hat{n}_y \\ \sqrt{2}B_0 \frac{c_{F/S}^2 - a^2}{c_A^2} \hat{n}_x \\ \rho_0 a^2 \end{bmatrix} \delta_{\text{pert}} \sin(\vec{k} \cdot \vec{x}) \quad (39)$$

$$a^2 = \frac{\gamma P_0}{\rho_0} \quad (40)$$

$$c_{F/S}^2 = \frac{1}{2} \left(a^2 + c_A^2 \pm \sqrt{c_A^4 + a^4} \right) \quad \text{corresponding to fast/slow speeds} \quad (41)$$

$$\hat{n} = (\hat{n}_x, \hat{n}_y) = \frac{\vec{n}}{|\vec{n}|} \quad (42)$$

All tests gave results consistent with second order accuracy, although waves at 2:1 showed slightly smaller (≥ 1.8 , versus 2.0) rates of convergence in some components. The modifications suggested by Stone were essential in obtaining second-order convergence for the 1:1 and 2:1 slope tests, increasing the convergence rates from first- to second-order for waves propagating along the diagonal. An example of the convergence rates for small amplitude fast waves is given in Table 1.

Component	Pre-Correction	Post-Correction
ρ	0.977	2.03
v_x	1.40	2.03
v_y	1.15	2.02
B_x	1.60	2.01
B_y	1.57	2.02
P	0.977	2.03

Table 1

Convergence rates by component for fast waves ($\delta_{\text{pert}} = 10^{-5}$) propagating at 45 degrees, for seven-wave MHD code before and after corrections suggested by Stone. [32]

A more stringent test of the code is to advect the linearized waves so that their profile remains stationary. In these tests, the background state has a non-zero velocity equal to the wavespeed:

$$\text{Alfvén: } \vec{u}_0 = -c_A \hat{n} \tag{43}$$

$$\text{Fast/Slow: } \vec{u}_0 = -c_{F/S} \hat{n} \tag{44}$$

An analysis of the eigenstructure of MHD shows (Appendix A) that such waves should cause trouble for the seven-wave MHD codes that do not suppress non-solenoidal fields. Errors generated in this case are not advected away, causing difficulty in the case that they are not diffused or otherwise dealt with. We find that the seven-wave and eight-wave MHD codes without a MAC projection go unstable for low amplitude ($\delta_{\text{pert}} = 10^{-4}$) fast waves with \hat{n} at 2:1 slope. The instability starts as a high frequency oscillation in the field parallel to \vec{n} . It grows with time and spreads to the other components, causing a low order of convergence even at early times. Once the oscillations reach a certain level, they drive the code unstable. The oscillations are almost completely absent, and the code stable, using a MAC projection. Adding filtering of the field further reduces the errors. A code including filtering but no MAC projection, and one with approximate projection and filtering both stabilize the scheme. We conclude that either projection or filtering is essential to stability for this class of problems.

4.2 Decay of Linearized MHD Waves

The equations of ideal MHD neglect the effects of viscosity and electrical resistivity. Numerical dissipation, however, can affect the solution in ways that mirror these physical effects. Following Ryu, Jones, and Frank [28] (hereafter RJF95), we measure the decay of Alfvén, fast, and slow waves, and use the implied physical resistivity as a measure of the numerical resistivity of our ideal MHD scheme.

We use exactly the same set-up as RJF95, with Alfvén, fast, and slow waves propagating at a 1:1 slope and wavelength of $\sqrt{2}$ times the length of one side of the computational domain. The decay of these waves was measured by fitting a decaying exponential to a measure of the wave strength,

$$\varepsilon = \sum_{i,j} l_k \cdot (\delta W)_{i,j}, \quad (45)$$

where $(\delta W)_{i,j} = W_{i,j} - W_0$ is the perturbation in the primitive variables. l_k is the left-eigenvector associated with the mode in question, evaluated at W_0 , the unperturbed state. Figure 2 shows the dependence of the decay rate on resolution for runs with number of cells per dimension $N = 16, 32, 64, 128, 256,$ and 512 . Comparison with Figure 4 of RJF95 reveals a much smaller decay rate for our unsplit method. Furthermore, we find that the decay rate varies according to the power law $\Gamma \propto N^{-3}$, not N^{-2} . The former is consistent with second-order accuracy, since the truncation errors have the following form:

$$\tau = C_2(\Delta x)^2 \frac{\partial^3 U}{\partial x^3} + C_3(\Delta x)^3 \frac{\partial^4 U}{\partial x^4} + \mathcal{O}(\Delta x^4). \quad (46)$$

We see here that the second-order error term is proportional to $\frac{\partial^3 U}{\partial x^3}$, which is dispersive in nature. The third-order term is the first error term that is dissipative. As a result, we expect the resistivity to decrease as Δx^3 , or N^{-3} . We posit that the smaller decay rates $L\Gamma/c$ and larger exponent describing the rate of decrease of Γ with increasing resolution can be attributed to one or both of two effects. The first is the unsplit methodology used in the scheme presented here. The second is the multidimensional corrections described in Section 3.1, which played a pivotal role in obtaining second-order accuracy on linearized MHD waves much like those used in this test.

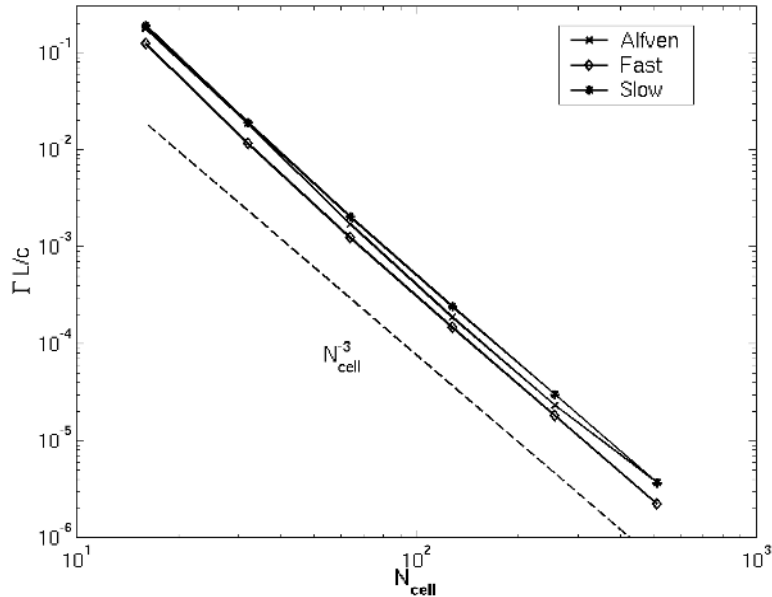


Fig. 2. Normalized damping rate versus resolution for the decay of Alfvén, fast, and slow modes propagating at a 1:1 slope. The calculations were done at resolutions of 16×16 , 32×32 , etc. up to 512×512 .

4.3 MHD Shock Tube

The second test is the MHD shock tube problem from Ryu, Jones, and Frank [28]. The solution consists of two fast shocks, one slow shock, one slow rarefaction, and a contact discontinuity. We have run the problem in two orientations: (1) with the shock velocity aligned with the x-axis of the computational domain (referred to below as 1-D), and (2) with this velocity inclined at a 2:1 slope. The latter configuration follows the 2-D shock tube test case from T00. It was chosen in order to test the multidimensional behavior of the code, meanwhile ensuring there were no serendipitous cancellations of errors, as might be the case in a 45° inclined case. In runs of a 1-D tube at R_{512} (ie. 512 cells per linear dimension) on a domain of size $L = 1$ to a time of $t = 0.08$, we are able to reproduce the results given in Table VI of Dai & Woodward [10] (hereafter DW94) with errors of $\leq 0.12\%$. These results are independent of the type of filter used and whether an approximate projection was performed. Both the seven- and eight-wave codes give exactly the same result. When combined with an observed first-order convergence rate, they give us great confidence in

the performance of both codes on 1-D shock tube problems.

The inclined version of this shock tube problem was run on grids of size $2N \times N$, with $N = 64, 128, \text{ and } 256$. We compared these results with coarsened versions of a R_{4096} 1-D shock tube results by first taking cuts of the data that included all cells lying along a line at 2:1 slope ($\alpha = 26.57^\circ$). Then, all velocities and fields in the cut were rotated by $-\alpha$. A check using the initial conditions showed that such a cut of the inclined initial conditions matched perfectly with the 1-D shock tube ICs outside of the jump region. (In the jump region, the inclined ICs were slightly different due to the volume averaging performed in producing them. This leads the initial jump to be spread over two zones instead of one.) Note that we chose the grid spacing for the inclined runs to be a factor of $\sqrt{5}$ smaller than the 1-D grid spacing, so that the shock covers the same distance in physical space in a given time. The domain was therefore of length $L = 1$ in the direction of shock propagation, and the final time $t = 0.08$, in both the aligned and inclined shock runs.

Figure 3 shows results for the inclined shock tube overlaid on the (coarsened) aligned result. They compare a R_{256} 1-D run with a $N = 256$ 2-D run that has equivalent effective grid spacing. The ideal MHD codes shown give fractional errors with an L^1 norm of 3% or less, averaged over all components. Errors in the normal component of the field are relatively large, however. As illustrated here and by Figure 11 in T00, shock-capturing codes generally produce a jump in the normal component of the magnetic field in non-grid-aligned shock tube problems such as this. More generally, such codes exhibit non-monotonic behavior inside the shock structure, so that Riemann invariants are not exactly preserved there [36]. We find that the observed jumps are indeed limited to the region of the shocks and initial jump, and have a very similar magnitude to that shown in the T00 results. Most importantly, despite errors in B_n inside the jump region, the jump relations are still satisfied outside of it.

Figures 4 and 5 show plots of the L^1 norm of the error versus resolution for the 2-D shock tube runs. We first note that neither the seven- nor eight-wave base code converges at first order. This is most evident in the errors in the transverse velocity and transverse field. The same plots indicate filtering alone is not sufficient to produce reasonable convergence rates. A MAC projection step added to either scheme produces much improved results (label 'MAC' in the figures), and the further addition of a filtering step gives first-order convergence in all components (label 'MAC+CF'). The addition of an approximate projection (label 'MAC+CF+AP') produces decreased errors in the normal field, but no improvement in the convergence rate or in the errors in other components.

While the results for seven- and eight-wave codes with a MAC projection step are very similar, such is not the case when the MAC projection is replaced

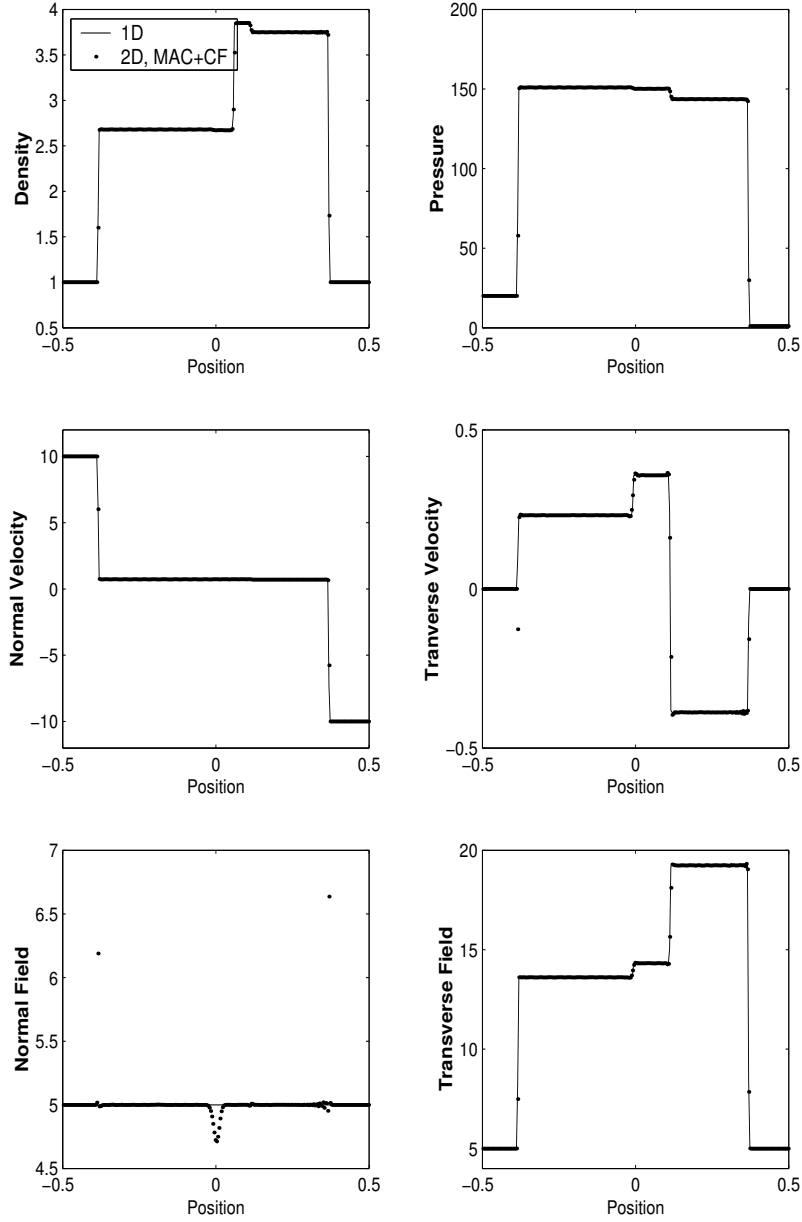


Fig. 3. MHD shock tube comparison by component. The 1-D shock tube (line) is used as a basis for comparison with 2-D shock tube runs with same effective resolution. The jumps in normal field B_n are expected, and the size of the jumps is similar to that published in Tóth 2000.

with an approximate projection (label 'CF+AP'). The seven-wave code gives results identical to those obtained using MAC and approximate projection plus a filter (MAC+CF+AP). In fact, the errors in normal component of the field are smaller in both cases. However, the errors for the approximate projection do not fall off at first order with the eight-wave code. This indicates that a MAC projection is necessary to preserve the accuracy of the eight-wave code.

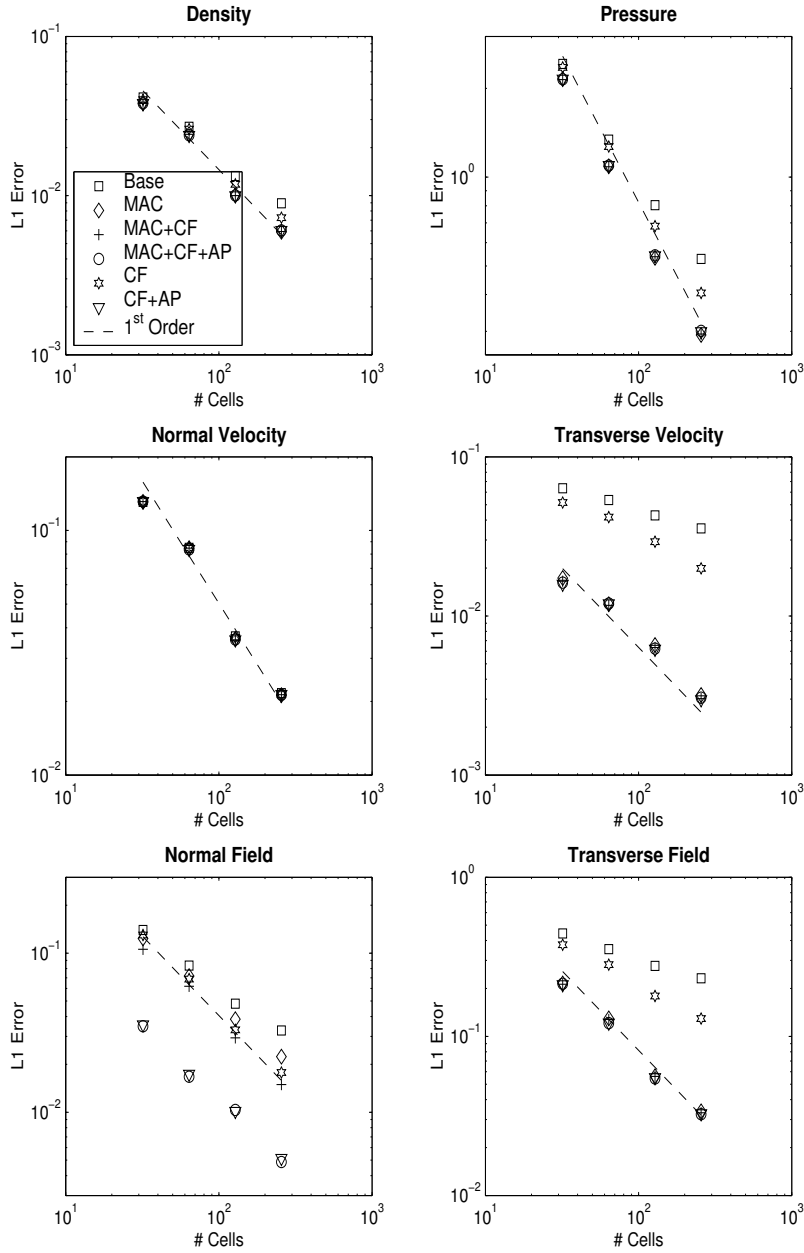


Fig. 4. L^1 norm of error versus resolution (number of cells per dimension) for 2-D MHD shock tube run with variants of the seven-wave code. The dashed line is a fiducial showing first-order fall-off of errors. Note that some manner of projection step is required for first-order convergence.

Table 2 gives the fractional errors, again relative to a high resolution 1-D shock tube run, in each component and averaged over all components. They are not much different in magnitude, though somewhat smaller, than those given in T00, Table III. Taken together, Table 2 and Figure 4 illustrate that where the fractional error is not a good discriminator between codes, the convergence rates of these codes can be a very good guide.

Code	$\delta\rho$	δp	δv_n	δv_t	δB_n	δB_t	$\bar{\delta}$
Base	0.00329	0.00427	0.00534	0.144	0.00653	0.0178	0.0302
CF	0.00265	0.00320	0.00520	0.0826	0.00355	0.00986	0.0178
CF+AP	0.00218	0.00229	0.00514	0.0106	0.00103	0.00235	0.00394
MAC	0.00212	0.00222	0.00505	0.0118	0.00447	0.00245	0.00469
MAC+CF	0.00213	0.00224	0.00508	0.0114	0.00299	0.00240	0.00437
MAC+CF+AP	0.00219	0.00228	0.00510	0.0112	0.000975	0.00233	0.00401

Table 2

Relative error, by component and averaged, for the 2-D MHD shock tube problem. Results are given for codes solving the ideal MHD equations. The base code includes only the hyperbolic solver, while the others combine this with a conservative filter step (CF), an approximate projection step (AP), and/or a MAC projection step (MAC). The errors were calculated using the same procedure as in Tóth (2000), Table III.

4.4 Magnetized Flux Tube

This problem involves a high-field, low gas pressure region bounded on both sides by a high-gas pressure, zero-field region. The physical domain is of length $L_x = L_y = 1$ on both sides. The base state has the entire 2-D domain in pressure balance. The boundaries between magnetized and unmagnetized regions are discontinuous and lie along $x = \pm 0.2$. In both regions, $\rho = 1$, $\vec{u} = 0$ and $B_x = B_z = 0$ initially. In the magnetized region, $x \in [-0.2, 0.2]$, $B_y = \sqrt{80\pi}$ and $P = 1$. Outside of this, $B_y = 0$ and $P = 11$. To the background state is added a sinusoidal perturbation upon the x-velocity whose amplitude is $\delta_{\text{pert}} = 0.01$ times the Alfvén speed, and covers the entire domain:

$$\delta u = \delta_{\text{pert}} c_A \sin(2\pi y) \quad \text{and} \quad \delta \vec{B} = \delta \rho = \delta P = \delta v = \delta w = 0 \quad (47)$$

The Alfvén speed in this case is $c_A = B_0/\sqrt{4\pi\rho} = \sqrt{20}$. The strong discontinuity in the field is expected to cause problems for algorithms that do not suppress non-solenoidal fields. In particular, at the stagnation points where the perturbation velocity is zero, truncation errors leading to non-solenoidal

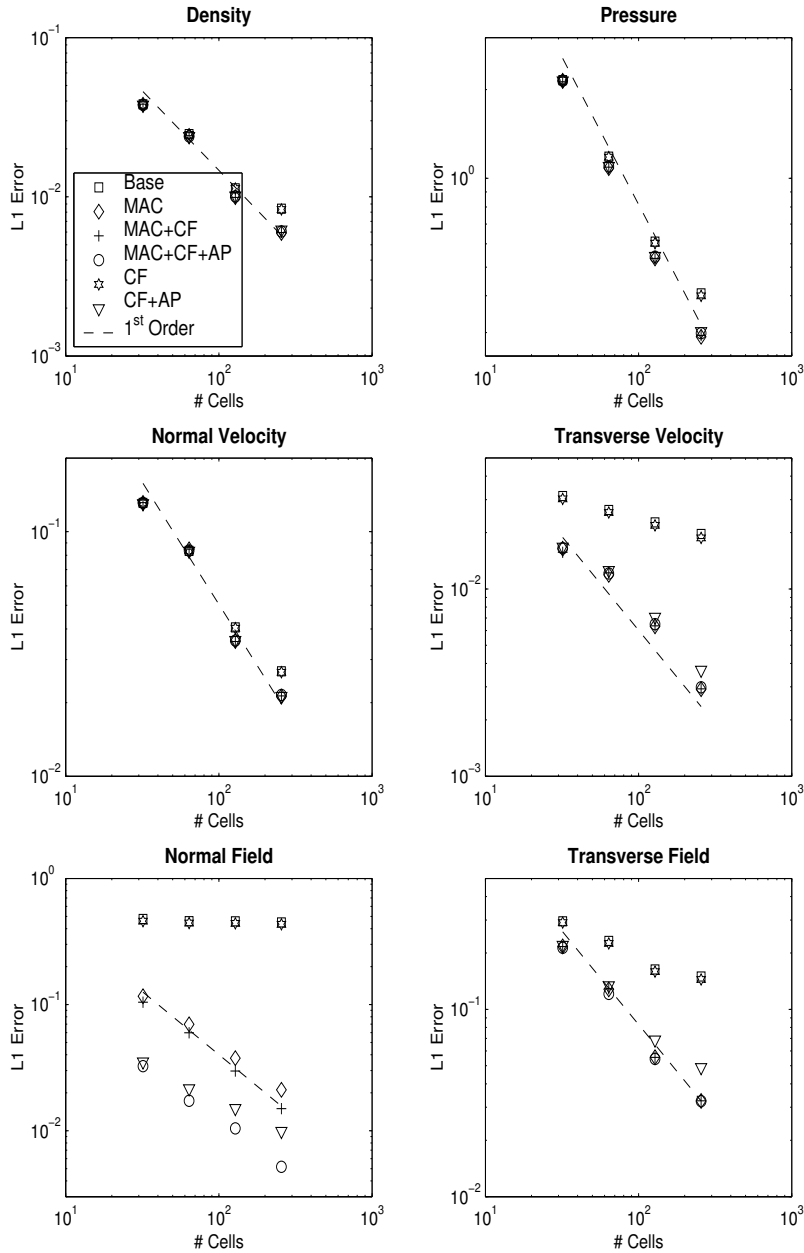


Fig. 5. L^1 norm of error versus resolution (number of cells per dimension) for 2-D MHD shock tube run with codes solving the divergence-wave MHD equations. The dashed line is a fiducial showing first-order fall-off of errors.

fields can build up and cause numerical schemes to go unstable. This is in fact what we find for base schemes run without a projection step. Non-solenoidal fields build up at these stagnation points, causing the production of spurious velocities and triggering numerical instabilities.

Figure 6 shows the initial conditions for the flux tube problem. The perturbation plucks the field. After an initial transient at start-up, the perturbation develops into a standing wave in the magnetized region. The unmagnetized region sloshes back and forth from one side of the flux tube to the other, owing to the periodic boundary conditions. We expect the initial velocity perturbation to cause standing Alfvénic waves in the magnetized region and, because of the slightly different oscillation frequencies of the two regions, compressive waves at the boundaries of the tube. The change in the internal energy tracks the compressive waves. The problem was run to a time $t = 6.0$, corresponding to about 60 Alfvén crossings of the short dimension of the tube. This was enough time for the problem to reach a steady state and subsequently evolve for many periods of the oscillation.

In this case we tested both seven- and eight-wave codes, but only with either a MAC projection and filter, or an approximate projection and filter. This is justified by the sub-par performance of the other variants on the inclined shock tube problem. These four versions of the code performed almost identically on this test, as evidenced by Figure 7. There, we plot global quantities such as kinetic and magnetic energy and L^1 norm of the monopole density. Note that the cell-centered divergence of the field is smaller for the approximate projection, though the dynamics remain the same.

4.5 *Inclined Flux Tube*

A second version of the flux tube problem, in which it is inclined at an angle of 45 degrees with respect to the original, is better at differentiating between the algorithms. It constitutes a strong test of the robustness and stability of the codes.

The size of the domain was $L_x = L_y = \sqrt{2}$, and both magnetized and unmagnetized regions have the same physical extent as in the aligned case. In both regions, we have $\rho = 1$, $\vec{u} = 0$ and $B_z = 0$ initially. In the magnetized region, $B_x = -\sqrt{40\pi}$, $B_y = \sqrt{40\pi}$, and $P = 1$. In the unmagnetized region, $P = 11$ and $B_x = B_y = 0$. The perturbation is again applied to the entire domain, and has strength $\delta_{\text{pert}} = 0.01$:

$$\delta u = \delta_{\text{pert}} \frac{c_A}{\sqrt{2}} \sin \left(2\pi \frac{-x + y}{\sqrt{2}} \right) \quad (48)$$

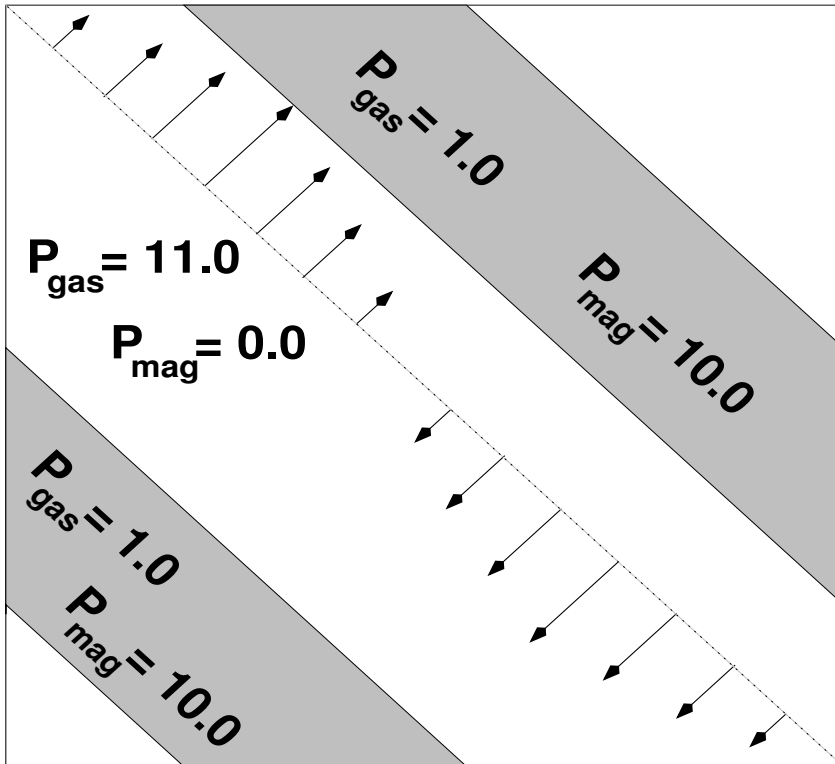
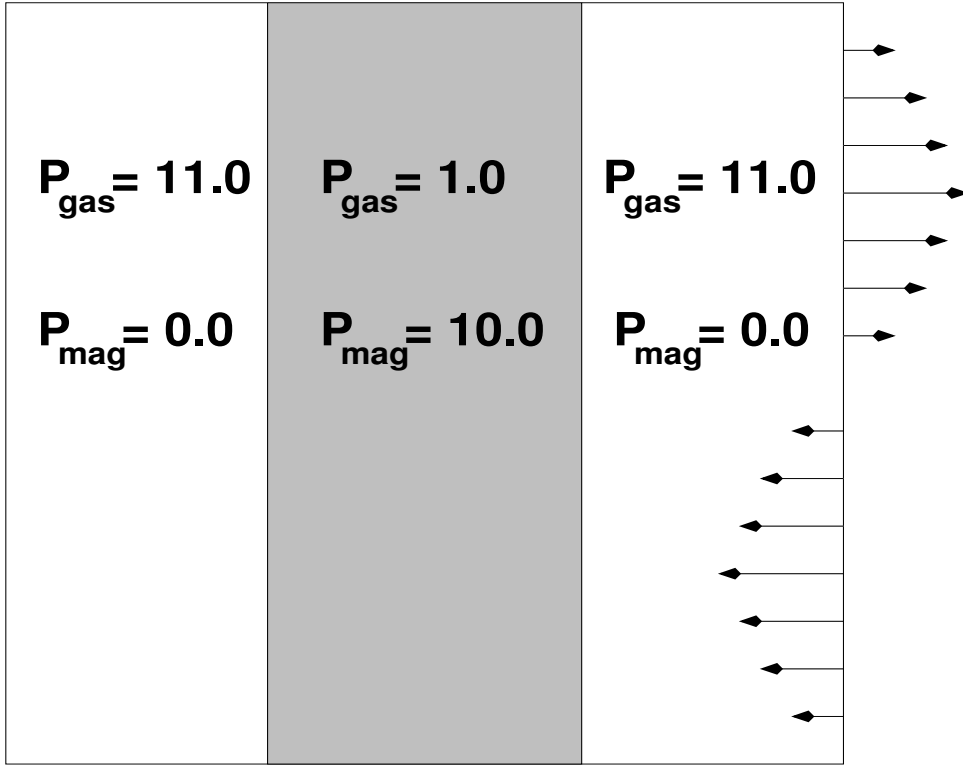


Fig. 6. Grid-aligned (top) and inclined (bottom) flux tube initial conditions. Vectors indicate the perturbation velocity field; it is constant along lines parallel to the vectors. There are two regions: one magnetized, the other not. The entire domain is initially in total (magnetic plus thermal) pressure balance. The velocity perturbation at 1% of the Alfvén speed is applied to the entire domain, causing the tube and surrounding unmagnetized medium to oscillate.

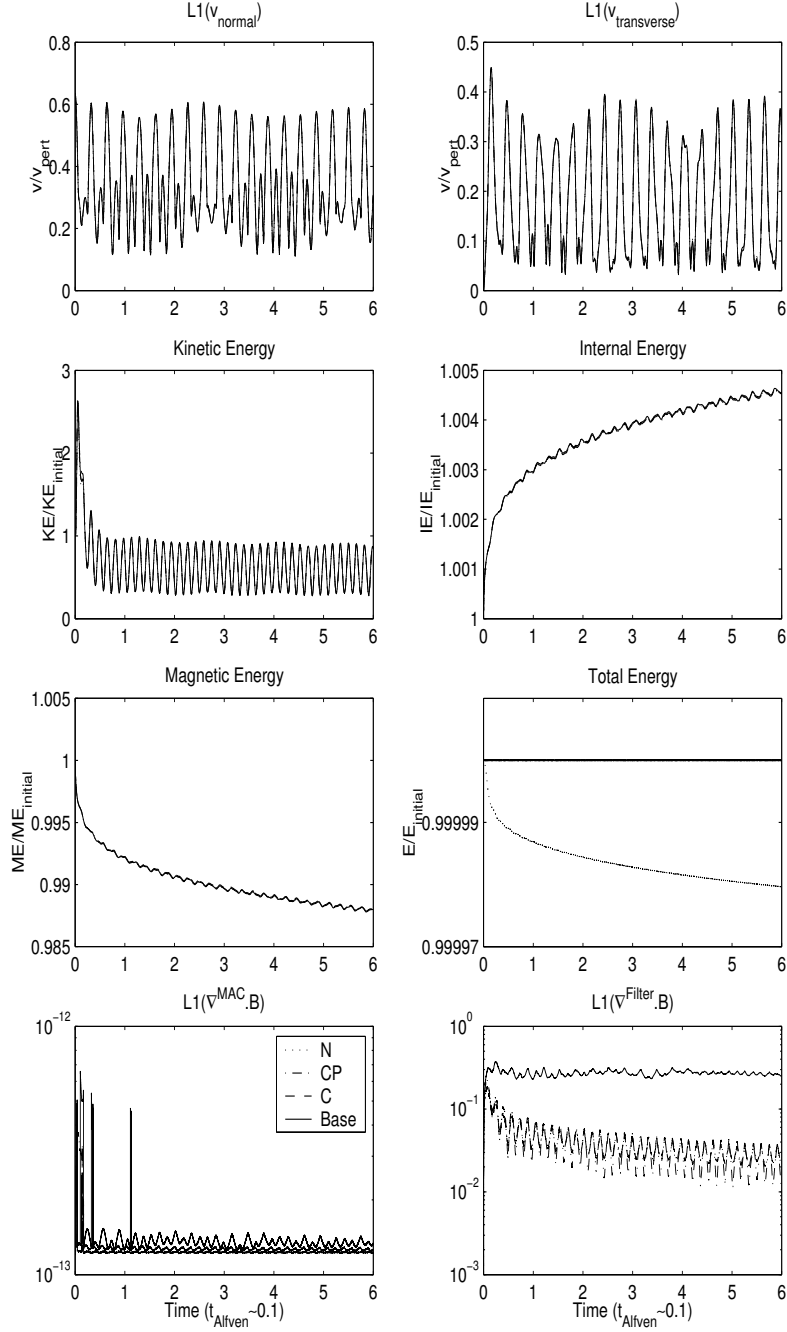


Fig. 7. Grid-aligned flux tube results for codes solving the ideal MHD equations. We use L^1 norms of velocities and monopole charge densities, plus grid-summed energies, to follow the evolution. The velocities are normalized to the initial perturbation velocity, $0.01c_A \approx 0.045$. Energies are normalized to their initial values. The two measures of monopole charge density use different discretizations: ∇^{MAC} corresponds to the MAC divergence of the states at cell-edges, while ∇^0 corresponds to a cell-centered measure of the divergence.

$$\delta v = \delta_{\text{pert}} \frac{c_A}{\sqrt{2}} \sin\left(2\pi \frac{-x+y}{\sqrt{2}}\right) \quad (49)$$

$$\delta \vec{B} = \delta \rho = \delta P = \delta w = 0 \quad (50)$$

One effect of rotating the flux tube is the addition of a numerical perturbation, or gridding effect. The volume-averaging required for producing the initial conditions creates a region of intermediate pressure and field between the fully magnetized and unmagnetized regions. The total pressure in this intermediate region was kept the same as elsewhere in the domain. The physical extent, and presumably physical effects, of this transition region become smaller as the resolution is increased. The second effect of our rotation is that the physical domain was somewhat larger. As mentioned above, this was done in order to keep the physical width of the magnetized and unmagnetized regions the same as for the aligned flux tube. The result is a factor of $\sqrt{2}$ mismatch between the grid spacing in the aligned and inclined flux tube runs.

We expect many qualitative similarities between the aligned and inclined flux tube results. The characteristics of the start-up transients are somewhat different, due in part to the gridding effects. The slightly different grid spacing also means that slightly higher resolution is required to obtain comparable results. Indeed, these tests were run at resolutions ranging by factors of two from R_{128} to R_{1024} , a factor of two higher than for the aligned case.

We again chose to run only those codes that showed close to first-order convergence for the inclined shock tube on this problem: seven- and eight-wave codes with either MAC projection and conservative filter, or approximate projection and conservative filter. There was very little difference between the two seven-wave codes, however. The difference between seven- and eight-wave codes was more noticeable. Both eight-wave codes both experienced large increases in kinetic energy, indicating that the magnetized tube was no longer in equilibrium. The size of this increase was largest ($\sim 10^3 \text{KE}_{\text{initial}}$) for the eight-wave code with approximate projection, which experienced large deviations from energy conservation. This behavior improved dramatically at higher resolutions, with the eight-wave code with MAC projection evidencing smaller jumps in kinetic energy.

Figure 9 shows a resolution study of the incline flux tube, with global quantities plotted out to $t = 1$. (This is about 5 Alfvén crossing times; a shorter time was chosen in order to show more detail. All codes that were stable to this time were also stable out to $t = 6$.) The plots include kinetic, internal, and magnetic energies, all scaled to their initial values. We also show the L^1 norm of cell-centered measure of the divergence, $\nabla^0 \cdot \vec{B}$, along with the magnetic energy change due to the MAC projection and filtering steps. First, note that the internal and magnetic energies appear to be converging, while convergence of the kinetic energy is not so clear. This result can be attributed to the large

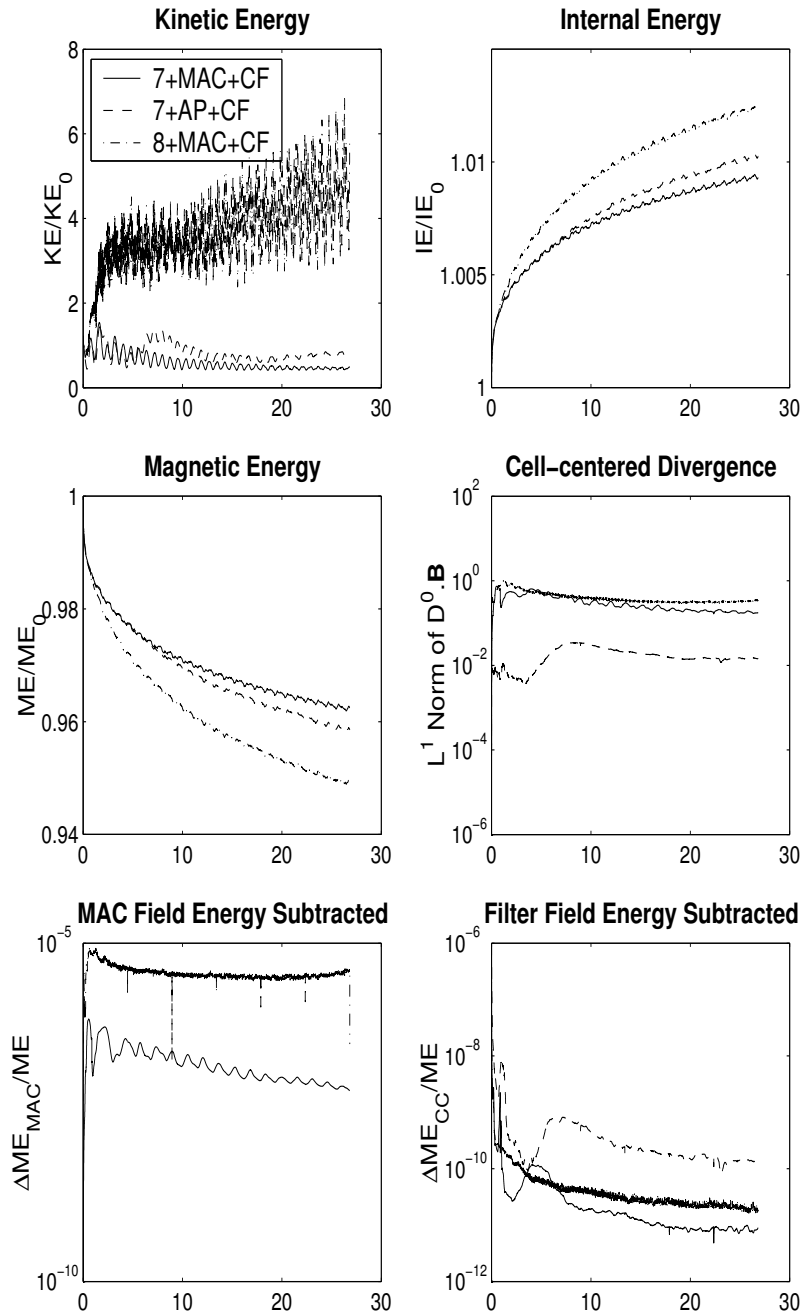


Fig. 8. Inclined flux tube results for two seven-wave (“7+MAC+CF” and “7+AP+CF”) and one eight-wave (“8+MAC+CF”) code. Note the relative size of increases in the kinetic energy of the codes, with smaller kinetic energies indicating stable oscillations.

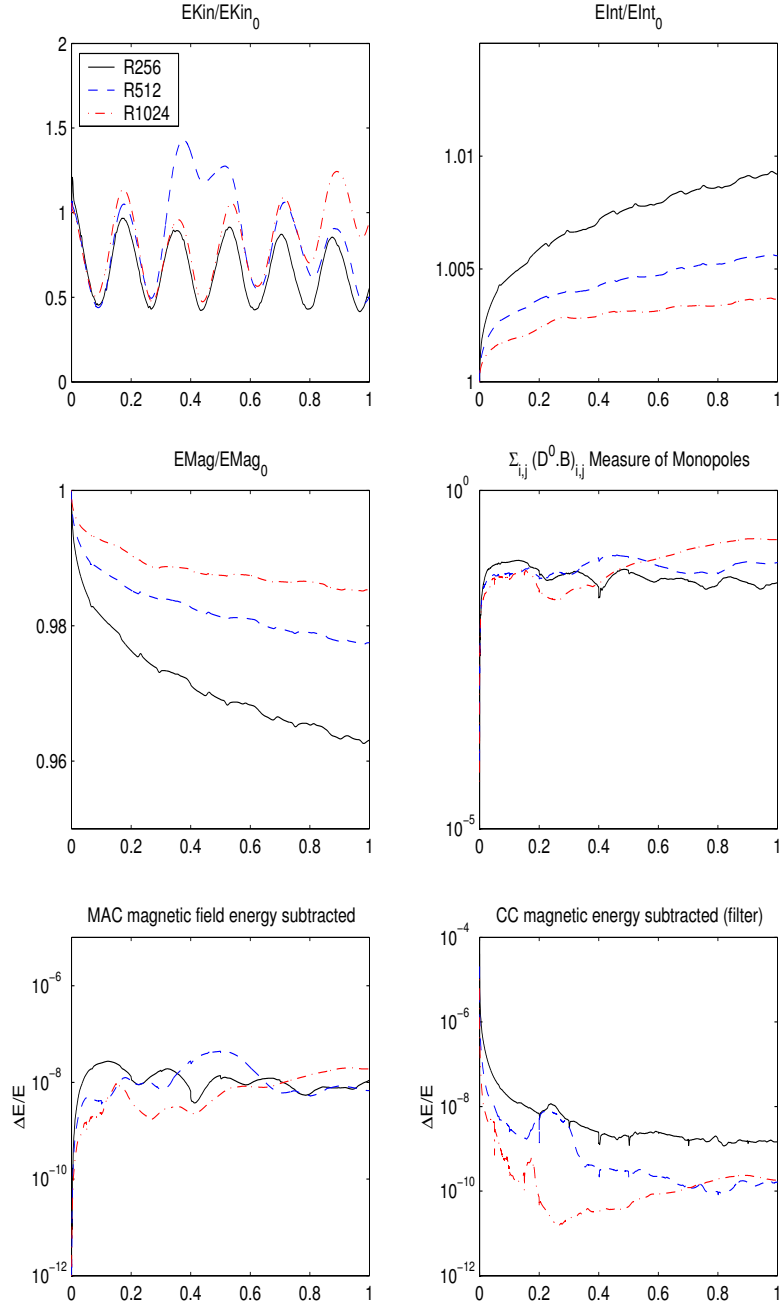


Fig. 9. Inclined flux tube results for 256, 512, and 1024 cells in each direction. These results used the 7-wave MHD code with conservative filtering and a coefficient of 0.1. The quantities shown are the same as in Figure 7.

difference in the size of these quantities; the initial kinetic energy is $\sim 10^5$ times smaller than the initial internal energy, and $\sim 10^4$ times smaller than the initial magnetic energy. Thus, small errors in either of these quantities can appear as sizeable changes in the kinetic energy. We note in particular that increases in the kinetic energy seem to correlate with increases in the fraction of the magnetic energy subtracted in either the filter or the MAC projection.

The expected qualitative similarities between aligned and incline flux tube results are present, such as oscillations in kinetic and magnetic energy of similar magnitudes. The size of the oscillations in the L^1 norm of v_t and v_n were also found to be very similar. However, substantial differences are also present. Comparing Figures 8 and 7, we note that the transient rise in kinetic energy early in the simulation begins later for the inclined tube than for the 1-D tube. In fact, the rise begins later and has smaller amplitude as resolution is increased, a fact that we attribute in part to the decrease in physical extent of the transition region.

As a final test of the nonlinear behavior of the code, we reproduce first MHD rotor problem outlined in T00, earlier performed by Balsara and Spicer [4]. This problem constitutes a central, high-density ($\rho = 10.0$) region surrounded by a low-density medium. The central region is given a constant angular velocity. The entire domain is threaded by an initially constant field, $\vec{B} = 5.0\hat{e}_x$. The rotation winds the magnetic field, sending Alfvén waves propagating into the surrounding medium.

This test was run using both seven-wave codes (MAC+CF and AP+CF) and one eight-wave variant (MAC+CF). We see no negative pressures in any of the code variants, and obtain first-order convergence (errors decreasing as the grid spacing to the 0.95 power, as averaged over all variables) using the seven-wave codes, and the eight-wave code with a MAC projection. A reproduction of Figure 18 from T00 is given in Figure 10. Note the steeper gradient in Mach number; we do not see the peaks mentioned in T00 and attributed to pressure undershoots.

5 Conclusions

We have presented an unsplit method for ideal MHD which, when combined with projection and filtering steps, shows no effects of non-solenoidal fields, while retaining the co-location of all physical quantities at cell-centers. The latter point is important because such a uniform centering makes it easier to extend the scheme to adaptive meshes, and has the advantage of using a well-understood Godunov method for time integration. These are in contrast to the Constrained Transport (CT) approach, where staggered grids add additional

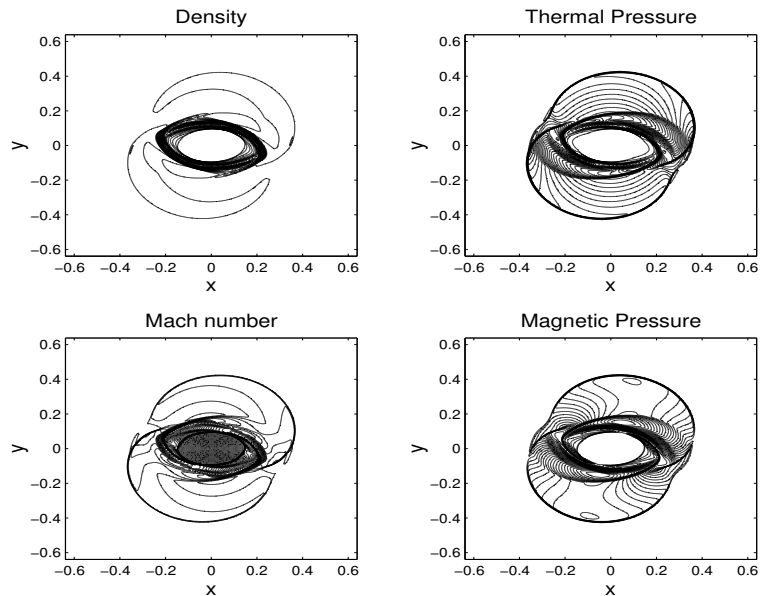


Fig. 10. Rotor problem results reproducing Figure 18 from T00. The plot has the same number of contours and these contours lie between the same limits. A slightly expanded domain was used ($x, y \in [-0.64, 0.64]$) in order to preserve the same grid spacing.

software complexity. Furthermore, it is not obvious how to discretize diffusion operators to include non-ideal MHD effects on staggered grids. Lastly, we add that the effective advection scheme for the magnetic field components is not of the standard type, and it is unclear what its properties are in the presence of under-resolved gradients.

The complexity and cost of the scheme relative to others is an important consideration. Our scheme requires three characteristic tracing steps plus twelve Riemann solution steps in 3-D. This is comparable to the nine tracing and three Riemann steps required in a 3-step Runge-Kutta scheme. The computational cost of projection is not insignificant, and adds additional software complexity. We note, however, that recent work with Poisson solvers on adaptive grids ([3,20]) have made this computational cost smaller than or comparable

to that of the hyperbolic update.

The Hodge projection of fields either at cell-edges (MAC projection) or cell-centers (in our case via an approximate projection) was found to be essential to accuracy and stability. The MAC projection was essential for accuracy of the eight-wave code. This is not surprising given the source terms added by the scheme; similar results were obtained by Tóth (2000). The seven-wave code results on the inclined shock tube were less sensitive to the type of projection used. The inclined flux tube results painted essentially the same picture, though the seven-wave MAC projected result was somewhat more robust.

Use of a filter alone on linear problems reduces computational cost and does not affect accuracy. We find that both filter and projection are required on strongly nonlinear problems such as the incline flux tube. Also essential to the accuracy of the scheme are the modifications suggested by Stone – linear waves not propagating along cardinal directions are not second-order accurate without it.

We found that in determining the accuracy of our code, both measuring the rate of convergence on nonlinear problems and comparing absolute magnitude of the error were required. Significant differences were found between the convergence rates even of individual primitive variables. These data can be crucial in choosing between different algorithmic variants of a base code. We encourage future authors to incorporate convergence testing, taking care to calculate errors in each variable separately, of simple nonlinear problems such as the shock tubes run here into their test suites.

Overall the ideal MHD code with a projection, either MAC or approximate, and a conservative filter performed best on the suite of tests presented. Filtering of the magnetic field was important to code stability in the nonlinear problems, and accuracy in the deficient wave problems. A filtering coefficient as high as $C = 0.1$ left the code second-order on linear problems.

The magnetized, perturbed flux tube constitutes a strong test of the stability of our schemes. The combination of a stationary discontinuity, low ratio of thermal to magnetic pressure (beta), and imposed perturbation caused significant problems for the base unsplit code without projection or filter. With a suitably chosen value of the filtering coefficient, the conservative filter helped stability and worked to decrease the magnitude of a centered-difference measure of the divergence of the field.

6 Acknowledgements

The authors would like to thank Jim Stone for pointing out the missing multidimensional MHD terms.

The research of Robert K. Crockett was supported in part by A Division at Lawrence Livermore National Laboratory.

The research of Christopher F. McKee was supported in part by NSF grant AST-0098365.

Work at the Lawrence Berkeley National Laboratory is sponsored by the US Department of Energy Applied Mathematical Sciences program under contract DE-AC03-76SF00098 and by the NASA Earth and Space Sciences Computational Technologies Program under interagency agreement number S-44830-X.

The research of Richard I. Klein was supported in part by a NASA ATP grant NAG5-12042. Both Richard I. Klein and Robert T. Fisher are supported under the auspices of the US Department of Energy at the Lawrence Livermore National Laboratory under contract W-7405-ENG-48.

A Appendix A: Divergence Constraints, Modified Equation Analysis, and Eigenvector Deficiencies

In this section, we will present a heuristic analysis of the effect of numerical errors in the divergence-free constraint on the stability of finite-difference methods for the ideal MHD equations. Our starting point will be the modified equation approach to analyzing the effect of truncation error on solution error. For any finite difference method, the modified equation is given by the original system of PDEs, with forcing terms given by the truncation error. In the present setting, the modified equation takes the following form:

$$\partial_t U^{Mod} + \nabla \cdot \vec{F}(U^{Mod}) = \tau_U(U^{Mod}) \tag{A.1}$$

$$\nabla \cdot U^{Mod} = \tau_D(U^{Mod}). \tag{A.2}$$

Here τ_U is the usual truncation error for the numerical method obtained from applying the difference operator to a solution to the differential equation evaluated on the grid. The equation for the evolution of τ_D is obtained by taking the divergence of the modified equation for the evolution of U^{Mod} ,

$$\partial_t \tau_D = \nabla \cdot \tau_U. \tag{A.3}$$

The truncation error forcing terms mimic the effect of numerical error on the computed solution. Specifically, we expect U^{Mod} , the solution to the modified equation, to satisfy $\|U^{\Delta x} - U^{Mod}\| = \mathcal{O}(\Delta x^{p+1})$, where $U^{\Delta x}$, the solution obtained from the p^{th} -order scheme on a grid with spacing Δx , satisfies $\|U^{\Delta x} - U\| = \mathcal{O}(\Delta x^p)$.

In particular, for MHD, the effect of numerical error can be understood in terms of the truncation-error forcing in the modified equation causing the solution to violate the divergence-free constraint. Without that constraint being satisfied, the remaining ideal MHD equations can exhibit eigenvector deficiencies in the linearized-coefficient matrix \mathbf{A} , leading to anomalous loss of regularity and ill-posedness. In the numerical simulation, this translates into loss of accuracy and possibly instability of the underlying difference method.

To see this, we consider the case of a small-amplitude wave corresponding to one of the eigenmodes of \mathbf{A} (see Equation 11):

$$W(\vec{x}, t) = W_0 + \alpha(x - \lambda_k t)r_k \quad (\text{A.4})$$

$$\mathbf{A}_0 r_k = \lambda_k r_k \quad r_k = (\tilde{r}_k, 0)^T. \quad (\text{A.5})$$

Then $W(\vec{x}, t)$ satisfies the MHD equations up to terms of $\mathcal{O}(\alpha^2)$.

Without loss of generality, we take the direction of propagation to be in the x -direction in 3-D. However, we allow our computational spatial grid to have an arbitrary orientation in space. In that case, the modified equation corresponding to our numerical solution to the PDE in primitive variables W is given by (see Equation 10)

$$\partial_t W^{Mod} + \mathbf{A}_0 \partial_x W^{Mod} = \tau_W \quad (\text{A.6})$$

$$W^{Mod} = (\tilde{W}^{Mod}, B_x^{Mod}). \quad (\text{A.7})$$

If we define the new variable $\alpha^{Mod} = l_k \cdot (\tilde{W}^{Mod} - \tilde{W}_0)$, the modified equation dynamics can be reduced to the following system of two equations:

$$\partial_t \alpha^{Mod} + \lambda_k \partial_x \alpha^{Mod} + (l_k \cdot a_B) \partial_x B_x^{Mod} = l_k \cdot \tilde{\tau} \quad (\text{A.8})$$

$$\partial_t B_x^{Mod} = \tau_B \quad (\text{A.9})$$

In the system here, if one of the computational spatial grid coordinate axes is aligned with \hat{x} , $\tau_B \equiv 0$. However, if the direction of propagation is not aligned with one of the computational coordinates, then in general $\tau_B \neq 0$. If $l_k \cdot a_B \neq 0$ and $\lambda = 0$, then the left-hand side of A.8-A.9 is an example of a first-order system with an eigenvector deficiency. Such systems have an obvious loss of spatial regularity: α^{Mod} grows like the derivative of τ_B . This

is in contrast to the behavior of well-posed hyperbolic systems, in which the solution has the same spatial regularity as the forcing. In other words, since regularity implies that there are as many derivatives in the solution as in the forcing term, discontinuous forcing of a hyperbolic system leaves the problem ill-posed. Discontinuity in τ_B implies that α^{Mod} can grow without bound, owing to its dependence on the derivative of τ_B .

In terms of a numerical method, we expect that the presence of such terms would lead to either an anomalous loss of accuracy or numerical instability. In the latter case, the forcing of α^{Mod} in A.8 takes the form of a finite difference operator applied to B_x^{Mod} , whose spatial variation is due entirely to τ_B . If τ_B fails to be smooth, either because of lack of smoothness in the initial data or in the finite difference formulae (eg. limiters), such lack of smoothness is immediately amplified.

This discussion also provides an explanation for the behavior of the method described here. The use of the MAC projection and the filter does not eliminate the truncation error terms that lead to the eigenvector deficiency, but regularizes it by smoothing. For example, the application of the filter in the plane-wave example corresponded to adding a diffusion term to the equation for B_x^{Mod} ,

$$\partial_t B_x^{Mod} = \tau_B + \eta \partial_x^2 B_x^{Mod}, \quad (\text{A.10})$$

with $\eta = \mathcal{O}(\Delta x)$. The use of the filter alone is sufficient to stabilize the small-amplitude plane wave solution in Section 4.1, and in that case leads to a second-order accurate result. The MAC projection performs a more drastic smoothing, but only on the intermediate form of B_x^{Mod} used to compute $\partial_x B_x^{Mod}$ in equation A.8.

References

- [1] A. S. Almgren, J. B. Bell, and W. Szymczak, A numerical Method for the Incompressible Navier-Stokes Equations Based on an Approximate Projection, *SIAM J. Sci. Comput.* **17** (1996) 358.
- [2] D. S. Balsara, Total Variation Diminishing Scheme for Adiabatic and Isothermal Magnetohydrodynamics, *Astrophys. J. Suppl.* **116** (1998) 133.
- [3] G. Balls and P. Colella, A Finite Difference Domain Decomposition Method Using Local Corrections for the Solution of Poisson's Equation, *J. Comput. Phys.* **180** (2002) 25-53.
- [4] D. S. Balsara and D.S. Spicer, A Staggered Mesh Algorithm Using

- High Order Godunov Fluxes to Ensure Solenoidal Magnetic Fields in Magnetohydrodynamic Simulations, *J. Comput. Phys.* **149** (1999) 270.
- [5] J. B. Bell, P. Colella, and L. H. Howell, An Efficient Second-Order Projection Method for Viscous Incompressible Flow, in *AIAA 10th Computational Fluid Dynamics Conference Proceedings*, June 1991, 360.
- [6] J. B. Bell, P. Colella, and J. A. Trangenstein, Higher Order Godunov Methods for General Systems of Conservation Laws, *J. Comput. Phys.* **82** (1989) 362.
- [7] J. U. Brackbill and D. C. Barnes, The Effect of Nonzero $\text{div}(\mathbf{B})$ on the Numerical Solution of the Magnetohydrodynamic Equations, *J. Comp. Phys.* **35** (1980) 426.
- [8] M. Brio and C.C. Wu. An Upwind Differencing Scheme for the Equations of Ideal Magnetohydrodynamics. *J. Comp. Phys.* **75** (1988), 400.
- [9] P. Colella, Multidimensional Upwind Methods for Hyperbolic Conservation Laws, *J. Comput. Phys.* **87** (1990) 171.
- [10] W. Dai and P. R. Woodward, An Approximate Riemann Solver for Ideal Magnetohydrodynamics, *J. Comput. Phys.* **111** (1994) 354.
- [11] W. Dai and P. R. Woodward, Extension of the Piecewise Parabolic Method to Multidimensional Ideal Magnetohydrodynamics. *J. Comput. Phys.* **115** (1994), 485.
- [12] W. Dai and P. R. Woodward, A Simple Finite Difference Scheme for Multidimensional Magnetohydrodynamical Equations, *J. Comput. Phys.* **142** (1998), 331.
- [13] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, Hyperbolic Divergence Cleaning for the MHD Equations, *J. Comp. Phys* 175 (2002) 645.
- [14] C. R. Evans and J. F. Hawley, Simulation of Magnetohydrodynamic Flows: A Constrained Transport Method, *Astrophys. J.* **332** (1988) 659.
- [15] S. A. Falle, Rarefaction Shocks, Shock Errors, and Low Order of Accuracy in ZEUS, *Astrophys. J.* **577** (2002), L123.
- [16] S. K. Godunov, The Symmetric Form of Magnetohydrodynamics Equation, *Numer. Methods Mech. Contin. Media* **1** (1972) 26.
- [17] T. I. Gombosi, K. G. Powell, and D. L. DeZeeuw, Axisymmetric Modeling of Cometary Mass Loading of an Adaptively Refined Grid: MHD Results, *J. Geophys. Res.* **99** (1994) 21, 525.
- [18] F. Harlow and J. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluids with Free Surfaces, *Physics of Fluids* **8** (1965) 2182.
- [19] L. H. Howell and J. B. Bell, An Adaptive-Mesh Projection Method for Viscous Incompressible Flow, *SIAM J. Sci. Comp.* **18** (1997), 996-1013.

- [20] J. F. Huang and L. Greengard, A Fast Direct Solver for Elliptic Partial Differential Equations on Adaptively Refined Meshes, *SIAM Journal on Scientific Computing*. **21**(4) (2000) 1551-1566.
- [21] M. F. Lai, A Projection Method for Reacting Flow in the Zero Mach Number Limit, Ph.D. Thesis, UC Berkeley 1993.
- [22] Londrillo, P. and Del Zanna, L., High-Order Upwind Schemes for Multidimensional Magnetohydrodynamics, *Astrophys. J.* **530** (2000), 508.
- [23] B. Marder, A Method for Incorporating Gauss' Law into Electromagnetic PIC Codes, *J. Comput. Phys.* **68** (1987) 48.
- [24] Miller, G. H. and Colella, P. A High-Order Eulerian Godunov Method for Elastic-Plastic Flow in Solids, *J. Comput. Phys.* **167** (2001) 131.
- [25] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, D. L. De Zeeuw, A Solution-Adaptive Upwind Scheme for Ideal Magnetohydrodynamics, *J. Comput. Phys.* **154** (1999), 284.
- [26] W. J. Rider, Filtering Non-solenoidal Modes in Numerical Solutions of Incompressible Flows. *IJNMF* **28**, 789.
- [27] P. L. Roe and D. S. Balsara, Notes on the Eigensystem of MHD, *SIAM J. Appl. Math.* **56** (1996) 57.
- [28] D. Ryu, T. W. Jones, and A. Frank, Numerical Magnetohydrodynamics in Astrophysics: Algorithm and Tests for Multidimensional Flow. *Astrophys. J* **452** (1995) 785.
- [29] D. Ryu, F. Miniati, T. W. Jones, and A. Frank. A Divergence-free Upwind Code for Multidimensional Magnetohydrodynamic Flows, *Astrophys. J.* **509** (1998) 244.
- [30] F. Shu. *The Physics of Astrophysics: Gas Dynamics*. University Science Books (1992).
- [31] J. M. Stone and M. L. Norman, ZEUS2D: A Radiation Magnetohydrodynamics Code for Astrophysical Flows in Two Space Dimensions. II The Magnetohydrodynamic Algorithms and Tests, *Astrophys. J. Suppl.* **80** (1992) 791.
- [32] J. M. Stone, Personal Communication.
- [33] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics : A Practical Introduction*. Springer-Verlag (1999).
- [34] G. Tóth, The div(B) Constraint in Shock-Capturing Magnetohydrodynamics Codes. *J. Comp. Phys.* **161** (2000) 605.
- [35] G. Tóth, Conservative and Orthogonal Discretization of the Lorentz Force. *J. Comp. Phys.* **182** (2002) 346.

- [36] P. R. Woodward, Piecewise Parabolic Methods for Astrophysical Fluid Dynamics, in *Astrophysical Radiation Hydrodynamics*, K.-H. Winkler and M. Norman, eds. Reidel (1986) 245.
- [37] A. L. Zachary, A. Malagoli, and P. Colella, A Higher-order Godunov Method for Multidimensional Ideal Magnetohydrodynamics, *SIAM J. Sci. Comput.* **15** (1994) 263.
- [38] A. L. Zachary and P. Colella, A Higher-order Godunov Method for the Equations of Ideal Magnetohydrodynamics, *J. Comput. Phys.* **99** (1992) 341.