JOURNAL OF COMPUTATIONAL PHYSICS **130,** 231–242 (1997)
ARTICLE NO. CP965580

An Implicit Scheme for Nonideal Magnetohydrodynamics

Ogden S. Jones, Uri Shumlak,* and D. Scott Eberhardt

Department of Aeronautics and Astronautics, University of Washington, Seattle, Washington 98195
E-mail: jones@aa.washington.edu

Received December 12, 1995; revised July 24, 1996

A new implicit algorithm is developed for solving the time-dependent, nonideal magnetohydrodynamic equations. It can also be used as an efficient relaxation scheme for steady state solutions. The algorithm is a finite-volume scheme that uses an approximate Riemann solver for the hyperbolic fluxes and central differencing applied on nested control volumes for the parabolic fluxes that arise from the non-ideal terms (i.e., resistivity and viscosity). In one dimension the scheme is second-order accurate in space and time. In two or three dimensions, the accuracy is between first and second order. For the class of problems considered, the implicit formulation is stable for any size time step, thus allowing efficient tracking of slower transients. The implicit operator is inverted using a lower–upper symmetric Gauss–Seidel iteration. Results from several test cases are presented that show good agreement with analytical solutions and illustrate the advantages of the scheme.

1. INTRODUCTION

Plasmas generally exhibit both collective (fluid) and individual (particle) behavior. In the MHD (magnetohydrodynamic) model, the plasma is treated like a conducting fluid having macroscopic parameters that accurately describe its particle-like interactions. This model is useful in many areas of plasma physics, including fusion plasmas, space and solar plasmas, and electric propulsion. The MHD model comprises a coupled set of nonlinear partial differential equations that must be solved numerically. Time-dependent MHD simulations are particularly challenging because of the wide range of time scales present in the model. In this paper we present an implicit algorithm for numerically solving the full (nonlinear, nonideal, time-dependent) MHD equations, which include dissipative terms due to resistivity and viscosity.

Mathematically, the MHD equations are a mixed set of hyperbolic and parabolic equations. Finite-volume methods are one of several different techniques available to solve these equations. They are simple to implement, easily adaptable to complex geometries, and well-suited to handle nonlinear phenomena such as mode coupling. The most difficult part of creating a finite-volume scheme for MHD is the determination of the hyperbolic fluxes at each cell interface. Approximate Riemann solvers are a class of methods for evaluating these fluxes that take into account the wave nature of hyperbolic equations. In the past, approximate Riemann solvers have been used extensively in the solution of the Euler and Navier–Stokes equations in fluid dynamics. Recently, a number of explicit schemes built around some type of approximate Riemann solver have been developed for the one-dimensional and multidimensional MHD equations [1–7].

While the MHD equations can be used to study plasma phenomena occurring on time scales as short as the transit time of a fast MHD wave, for many problems the important physics occurs on time scales that are much longer. For example, it can be shown that resistive tearing modes, which are important in studying fusion plasmas, evolve on a time scale given by [8]

$$\tau_{\text{tearing}} \propto \tau_A^{2/5}\, \tau_\eta^{3/5} = (\text{Lu})^{3/5}\, \tau_A, \tag{1}$$

where τ_A is the Alfvén time, τ_η is the resistive diffusion time, and Lu is the Lundquist number, which is defined as

$$\text{Lu} = \tau_\eta / \tau_A. \tag{2}$$

If Lu is 10^6, which is typical for laboratory plasmas in fusion applications, the resistive tearing time is approximately 4000 times larger than the Alfvén time. In this case, an explicit scheme would limit the time step to a much smaller value than is needed to accurately resolve the transient behavior. An implicit scheme removes the numerically imposed time-step constraint, allowing much larger time steps.

There have been a number of applications of implicit finite-difference methods to nonlinear multidimensional MHD problems. Lindemuth and Killeen [9] and Schnack and Killeen [10] used fully implicit schemes that employed iterations at each time step to invert the implicit operator. However, for large, three-dimensional problems, inverting the full implicit operators became impractical, so various

* Alternate address is High Energy Plasma Division, Phillips Laboratory, Kirtland Air Force Base, New Mexico 87117-5776.

0021-9991/97 $25.00

semi-implicit schemes were introduced. Weber *et al.* [11] used a time-split method whereby the convective terms were solved explicitly and the diffusive terms were solved implicitly. The MACH3 code [12] also uses a time-splitting scheme, but the equations are split in such a way as to remove the Alfvén speed from the numerical stability considerations. Since the equations are decoupled in these schemes, the equations must be iterated at each time step until they converge. Schnack *et al.,* [13] introduced a class of semi-implicit schemes that used operator-splitting to remove the numerical time-step restrictions. These methods have the advantage of not requiring iterations, but inaccuracies are introduced by the operator splitting that limit the allowable time step.

In this work we have developed a new implicit scheme for solving the time-dependent, nonideal MHD equations. It is among the first implicit schemes for MHD that uses an approximate Riemann solver to evaluate the hyperbolic fluxes. The implicit scheme is unique in that it is based upon a flux-vector splitting of the hyperbolic fluxes. The formulation allows the time step to be chosen based on the time scales one wishes to resolve rather than on the stability of the numerical method. This can be important for problems where the time scales of interest are much longer than the fast MHD transit time. As long as the Reynolds and Lundquist numbers are much larger than one, which is the case for most fusion and space plasmas, this method is numerically stable for any CFL number. As with most implicit schemes, the implicit operator is solved iteratively at each time step. However, due to its simple form in this case, it can be solved using an approximate LU decomposition technique that is extremely efficient. In section 2 we describe the MHD equations and their properties. The details of the new algorithm are presented in section 3. Section 4 summarizes the results of several benchmark tests we performed to validate the new algorithm.

2. MHD EQUATIONS

The three-dimensional, viscous, resistive MHD plasma model is a set of mixed hyperbolic and parabolic equations. When expressed in conservative, nondimensional form, the equation set is

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\mathbf{v}\\ \mathbf{B}\\ e\end{bmatrix}+\nabla\cdot\begin{bmatrix}\rho\mathbf{v}\\ \rho\mathbf{v}\mathbf{v}-\mathbf{B}\mathbf{B}+(p+\mathbf{B}\cdot\mathbf{B}/2)\mathbf{I}\\ \mathbf{v}\mathbf{B}-\mathbf{B}\mathbf{v}\\ (e+p+\mathbf{B}\cdot\mathbf{B}/2)\,\mathbf{v}-(\mathbf{B}\cdot\mathbf{v})\mathbf{B}\end{bmatrix} \tag{3}$$

$$=\nabla\cdot\begin{bmatrix}0\\ (Re_A)^{-1}\,\bar{\bar{\tau}}\\ (Lu)^{-1}\,\bar{\bar{E}}_{res}\\ (Re_A)^{-1}\,\mathbf{v}\cdot\bar{\bar{\tau}}-(Lu)^{-1}\,\bar{\bar{\eta}}\cdot(\nabla\times\mathbf{B})\times\mathbf{B}\\ +(Pe_A)^{-1}\,\bar{\bar{k}}\cdot\nabla T\end{bmatrix}.$$

The variables are density (ρ), velocity ($\mathbf{v}$), magnetic induction ($\mathbf{B}$), pressure (p), energy density (e), and temperature (T). The energy density is

$$e=\frac{p}{\gamma-1}+\rho\frac{\mathbf{v}\cdot\mathbf{v}}{2}+\frac{\mathbf{B}\cdot\mathbf{B}}{2}, \tag{4}$$

where $\gamma=c_p/c_v$ is the ratio of the specific heats. The tensor $\bar{\bar{E}}_{\rm res}$ is defined such that

$$\nabla\cdot\bar{\bar{E}}_{\rm res}=-\nabla\times(\bar{\bar{\eta}}\cdot\nabla\times\mathbf{B}). \tag{5}$$

The other nondimensional tensors are the stress tensor ($\bar{\bar{\tau}}$), the electrical resistivity ($\bar{\bar{\eta}}$), and the thermal conductivity ($\bar{\bar{k}}$), and $\mathbf{I}$ is the identity matrix. The nondimensional numbers are defined as follows:

Lundquist number:	$\mathrm{Lu}\equiv\mu_o c_a L/\eta$	
modified Reynolds number:	$\mathrm{Re}_A\equiv c_a L/\nu$	(6)
modified Péclet number:	$\mathrm{Pe}_A\equiv c_a L/\kappa.$	

The characteristic variables are length (L), Alfvén speed ($c_a=B/\sqrt{\mu_o\rho}$), kinematic viscosity (ν), electrical resistivity (η), and thermal diffusivity ($\kappa=k/\rho c_p$); μ_o is the permeability of free space. Note that the characteristic speed appearing in Re_A and Pe_A is the Alfvén speed, rather than the flow velocity.

For convenience, the MHD equation set (Eq. (3)) is rewritten in the compact form

$$\frac{\partial\mathbf{Q}}{\partial t}+\nabla\cdot\bar{\bar{\mathbf{T}}}_h=\nabla\cdot\bar{\bar{\mathbf{T}}}_p, \tag{7}$$

where $\mathbf{Q}$ is the vector of conservative variables, $\bar{\bar{\mathbf{T}}}_h$ is the tensor of hyperbolic fluxes, and $\bar{\bar{\mathbf{T}}}_p$ is the tensor of parabolic fluxes. The forms of these vectors and tensors can be seen from Eq. (3).

3. NUMERICAL METHOD

The method presented here is based on an algorithm that has been applied to the time-dependent, incompressible Navier–Stokes equations [14]. In one dimension the algo-

rithm is second-order accurate in space and time. For multidimensions, the accuracy is less than second order, but greater than first order. In the following description of the algorithm, the accuracies quoted are for one dimension. We will derive the algorithm for two dimensional Cartesian coordinates. The extension to three dimensions and general coordinates is straightforward. To begin, we express the MHD equations as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{F}_p}{\partial x} + \frac{\partial \mathbf{G}_p}{\partial y} = 0, \tag{8}$$

where $\mathbf{F}$ is the hyperbolic flux vector in the x direction (i.e., $\bar{\mathbf{T}}_h = (\mathbf{F}, \mathbf{G}, \mathbf{H})$) and $\mathbf{F}_p$ is the parabolic flux vector in the x direction (i.e., $\bar{\mathbf{T}}_p = -(\mathbf{F}_p, \mathbf{G}_p, \mathbf{H}_p)$). We then discretize Eq. (8) in space and time, evaluating the fluxes at the $n + 1$ time level to get

$$\frac{1}{2\Delta t}(3\mathbf{Q}_{ij}^{n+1} - 4\mathbf{Q}_{ij}^{n} + \mathbf{Q}_{ij}^{n-1}) = -[(\mathbf{R}_h)_{ij}^{n+1} + (\mathbf{R}_p)_{ij}^{n+1}], \tag{9}$$

where $\mathbf{R}_h$ and $\mathbf{R}_p$ are the discretizations of the hyperbolic and parabolic fluxes, respectively.

Equation (9) is implicit and must be solved iteratively. Let $\mathbf{Q}^{n+1,m}$ denote the mth iteration of the solution at the $n + 1$ time level. To derive a recursive expression for the next iteration of $\mathbf{Q}^{n+1}$ in terms of the previous iteration, we rewrite Eq. (9) as

$$\left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^{m+1} = -[(\mathbf{R}_h)_{ij}^{n+1,m+1} + (\mathbf{R}_p)_{ij}^{n+1,m}], \tag{10}$$

where

$$\left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^{m+1} \equiv \frac{1}{2\Delta t}(3\mathbf{Q}_{ij}^{n+1,m+1} - 4\mathbf{Q}_{ij}^{n} + \mathbf{Q}_{ij}^{n-1}). \tag{11}$$

This equation is still implicit, because the time derivative and hyperbolic flux terms are evaluated at the $m + 1$ iteration. (Evaluating the parabolic flux at the old iteration level, m, results in a significantly simpler implicit operator, as will be explained later). These terms are related to the previous iteration by linearizing them using truncated Taylor series expansions. That is,

$$\left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^{m+1} \approx \left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^{m} + \frac{\partial(\partial \mathbf{Q}/\partial t)_{ij}^{m}}{\partial \mathbf{Q}}(\mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}{}^{n+1,m}) \tag{12}$$

and

$$(\mathbf{R}_h)_{ij}^{m+1} \approx (\mathbf{R}_h)_{ij}^{m} + \frac{\partial(\mathbf{R}_h)_{ij}^{m}}{\partial \mathbf{Q}}(\mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}^{n+1,m}). \tag{13}$$

It is important to note that the partial derivatives above are taken with respect to $\mathbf{Q}$ at every cell, not just $\mathbf{Q}_{ij}$. The partial derivative of $\partial \mathbf{Q}/\partial t$ with respect to $\mathbf{Q}$ is simply

$$\frac{\partial(\partial \mathbf{Q}/\partial t)}{\partial \mathbf{Q}} = \frac{3\mathbf{I}}{2\Delta t}, \tag{14}$$

since it only varies with $\mathbf{Q}_{ij}^{n+1,m}$. However, the partial derivative of $\mathbf{R}_h$ is much more difficult to evaluate. In fact, in order to make it tractable, it is evaluated using a first-order accurate approximation of the hyperbolic fluxes, which we will denote as $\hat{\mathbf{R}}_h$, rather than the full second-order accurate discretization ($\mathbf{R}_h$). This first-order approximation can be written generally as

$$(\hat{\mathbf{R}}_h)_{ij} = f(\mathbf{Q}_{ij}, \mathbf{Q}_{i-1,j}, \mathbf{Q}_{i+1,j}, \mathbf{Q}_{i,j-1}, \mathbf{Q}_{i,j+1}), \tag{15}$$

so that the $m + 1$ iteration of $(\hat{\mathbf{R}}_h)_{ij}$ is coupled to five points from the previous iteration. Substituting these expressions back into Eq. (9) and rearranging, we get

$$\left[\frac{\partial(\hat{\mathbf{R}}_h)_{ij}^{m}}{\partial \mathbf{Q}} + \frac{3\mathbf{I}}{2\Delta t}\right]\Delta \mathbf{Q}_{ij}^{m} = -\left[(\mathbf{R}_h)_{ij}^{m} + (\mathbf{R}_p)_{ij}^{m} + \left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^{m}\right], \tag{16}$$

where

$$\Delta \mathbf{Q}_{ij}^{m} \equiv \mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}^{n+1,m}. \tag{17}$$

At each time step, Eq. (16) is iterated until $\Delta \mathbf{Q}^m$ is driven to approximately zero, at which point the original differential equation is approximately satisfied. The implementation of this iteration depends, of course, on the details of the discretization of the hyperbolic and parabolic fluxes ($\mathbf{R}_h$ and $\mathbf{R}_p$) and on the linearization of the hyperbolic fluxes ($\partial \hat{\mathbf{R}}_h/\partial \mathbf{Q}$). The left-hand side of Eq. (16) can be thought of as an implicit operator operating on $\Delta \mathbf{Q}^m$. The implicit operator is a large banded block matrix that is costly to invert directly. Instead, we make a number of simplifications to the operator and then invert it using the lower upper symmetric Gauss–Seidel (LU-SGS) technique [15]. The hyperbolic fluxes are differenced by applying Harten's approximate Riemann solver [16] within the framework of the multidimensional technique developed by Powell [4]. The parabolic fluxes are discretized using central differencing on an offset finite volume mesh. In the following sections, each of these parts of the algorithm are described separately in more detail.

3.1. *Approximate Riemann Solver*

The finite volume discretization of the hyperbolic fluxes can be written as

$$(\mathbf{R}_h)_{ij} = \mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j} + \mathbf{G}_{i,j+1/2} - \mathbf{G}_{i,j-1/2}. \quad (18)$$

Note that in this equation, and all that follow, the grid metric terms (cell areas and volumes) are omitted for clarity. The fluxes at the cell faces in each direction are evaluated by solving a one-dimensional linear Riemann problem defined by the discontinuous jump in $\mathbf{Q}$ between each cell. That is, if we let $\mathbf{Q}_\mathrm{l}$ and $\mathbf{Q}_\mathrm{r}$ denote the states to the left and right of a cell interface, then the x direction flux is determined by solving

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}(\mathbf{Q}_\mathrm{l}, \mathbf{Q}_\mathrm{r}) \frac{\partial \mathbf{Q}}{\partial x} = 0, \quad (19)$$

where $\mathbf{A}$, the Jacobian of the $\mathbf{F}$ with respect to $\mathbf{Q}$, is a function of the left and right states. Similarly, a linear Riemann problem in the y direction is solved to get the y fluxes. There are a number of different types of approximate Riemann solvers that have been developed. In this work, we did not develop a completely new Riemann solver, but rather adapted a solver originally developed for the Euler equations by Harten [16].

To solve the linear Riemann problem defined above, we write $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{X}^{-1} \mathbf{\Lambda} \mathbf{X}, \quad (20)$$

where $\mathbf{X}$ is a matrix whose columns are the right eigenvectors of $\mathbf{A}$, $\mathbf{X}^{-1}$ is its inverse (its rows are the left eigenvectors of $\mathbf{A}$), and $\mathbf{\Lambda}$ is a matrix having the eigenvalues of $\mathbf{A}$ along the diagonal. The eigenvalues of $\mathbf{A}$ are

$$\lambda = (v_x, 0, v_x \pm c_f, v_x \pm c_s, v_x \pm c_{a,x})^\mathrm{T}, \quad (21)$$

where c_f and c_s are the fast and slow magnetosonic speeds in the x direction, and $c_{a,x}$ is the Alfvén speed based on the x component of the magnetic field. These can be expressed as

$$c_f^2 = \frac{1}{2}\{a^2 + c_a^2 + [(a^2 + c_a^2)^2 - 4a^2 c_{a,x}^2]^{1/2}\} \quad (22)$$

$$c_s^2 = \frac{1}{2}\{a^2 + c_a^2 - [(a^2 + c_a^2)^2 - 4a^2 c_{a,x}^2]^{1/2}\} \quad (23)$$

$$c_{a,x}^2 = B_x^2/\mu_o\rho. \quad (24)$$

Here, a is the ion acoustic speed, which for a perfect gas is

$$a^2 = \gamma p/\rho. \quad (25)$$

The zero eigenvalue arises from the fact that the $\mathbf{j} \times \mathbf{B}$ force acts perpendicularly to the directions of $\mathbf{j}$ and $\mathbf{B}$, so that the $\mathbf{F}$ flux vector has a zero term corresponding to B_x. For approximate Riemann solvers, there are basically two approaches to solving this problem. The most common is to drop B_x from $\mathbf{Q}$ when applying the Riemann solver to the x fluxes, drop B_y from $\mathbf{Q}$ when applying it to the y fluxes, and so forth. That is, in each direction a different seven-variable Riemann problem is solved.

An alternate approach, which we are using in this work, has been developed by Powell [4]. In this approach the Jacobians are modified in order to remove the zero eigenvalue singularity, and a source term is added that exactly cancels the terms introduced by the modification of the Jacobians. The eigenvalues of the modified Jacobian, $\tilde{\mathbf{A}}$, are

$$\lambda = (v_x, v_x, v_x \pm c_f, v_x \pm c_s, v_x \pm c_{a,x})^\mathrm{T}. \quad (26)$$

Thus the modification of $\mathbf{A}$ has changed the zero eigenvalue to the flow speed, while keeping the others unchanged. In the same way, the zero eigenvalue for the y flux is changed to v_y. The source term, $\mathbf{S}_\mathrm{div}$, is given by

$$\mathbf{S}_\mathrm{div} = -\begin{bmatrix} 0 \\ \mathbf{B} \\ \mathbf{v} \\ \mathbf{v} \cdot \mathbf{B} \end{bmatrix} \nabla \cdot \mathbf{B}. \quad (27)$$

It is proportional to $\nabla \cdot \mathbf{B}$, which is zero analytically, but not numerically. It can be shown [4] that the artificial eighth wave is associated with the convection of nonzero $\nabla \cdot \mathbf{B}$ produced by truncation errors.

The main reason we chose the eight-variable Riemann solver approach is that it was simpler to incorporate into our implicit iteration scheme. An additional benefit of this approach for the problems we have considered, is that the formulation automatically insures that $\nabla \cdot \mathbf{B}$ remains approximately zero and does not grow, because any finite $\nabla \cdot \mathbf{B}$ is convected out of the domain [4]. However, it should be noted that for more complicated flows having stagnation points and recirculation regions, this technique would not guarantee that $\nabla \cdot \mathbf{B}$ would remain zero everywhere in the domain. For those types of problems, we would likely have to solve an auxilliary equation between time steps in order to remove any non-solenoidal components of the field, as is commonly done in other MHD Riemann solvers (e.g., [3]).

With the addition of the source term, $\mathbf{R}_h$ becomes

$$(\mathbf{R}_h)_{ij} = \mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j} + \mathbf{G}_{i,j+1/2} - \mathbf{G}_{i,j-1/2} + (\mathbf{S}_\mathrm{div})_{ij}. \quad (28)$$

The fluxes are calculated by using the modified Jacobians with Harten's scheme in the form used by Yee *et al.* [17]. In this form, the x direction flux at the $i + \frac{1}{2}$ face is

$$\mathbf{F}_{i+1/2} = \tfrac{1}{2}[\mathbf{F}_{i+1} + \mathbf{F}_i] + \tfrac{1}{2}\sum_k \phi^k_{i+1/2}\mathbf{r}^k_{i+1/2}, \tag{29}$$

where $\mathbf{r}^k$ is the k^{th} right eigenvector of $\tilde{\mathbf{A}}$, and ϕ^k is a nonlinear dissipation term that is designed to provide just enough dissipation near sharp gradients to suppress numerical oscillations while still achieving higher order spatial accuracy throughout the rest of the domain. In smooth regions of the solution, ϕ is approximately zero, so that $\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}$ yields second-order accurate central differencing. However, near steep gradients and discontinuities, ϕ makes the flux approximately equal to its first-order accurate upwind form given by

$$\mathbf{F}_{i+1/2} = \tfrac{1}{2}[\mathbf{F}_{i+1} + \mathbf{F}_i] - \tfrac{1}{2}\sum_k \psi(\lambda^k_{i+1/2})\alpha^k_{i+1/2}\mathbf{r}^k_{i+1/2}, \tag{30}$$

where

$$\alpha^k_{i+1/2} = \mathbf{l}^k_{i+1/2}\cdot(\mathbf{Q}_{i+1} - \mathbf{Q}_i), \tag{31}$$

λ^k is the k^{th} eigenvalue of $\tilde{\mathbf{A}}$, $\mathbf{l}^k$ is the k^{th} left eigenvector of $\tilde{\mathbf{A}}$, and ψ is the absolute value function with a smoothing term that adds a small amount of dissipation in order to enforce the entropy condition. It is given by

$$\psi(z) = \begin{cases} \tfrac{1}{2}\left[\dfrac{z^2}{\varepsilon} + \varepsilon\right] & \text{if } |z| \le \varepsilon \\ |z| & \text{if } |z| > \varepsilon \end{cases}, \tag{32}$$

where ε is typically set to 0.01. Note that the values at the cell interface ($i + \frac{1}{2}$) are obtained by a simple average of the neighboring cells. It is possible to evaluate the average state at the cell interface using a more complicated and potentially more robust averaging such as the "Roe average." However, we have found in practice that simple averaging is sufficient, except for the most extreme cases, such as for very strong shocks.

This full expression for ϕ is

$$\phi^k_{i+1/2} = g^k_i + g^k_{i+1} - \psi(\lambda^k_{i+1/2} + \gamma^k_{i+1/2})\alpha^k_{i+1/2}, \tag{33}$$

where

$$g_i = \frac{S}{2}\cdot\max[0, \min[\psi(\lambda^k_{i+1/2})|\alpha^k_{i+1/2}|, S\,\psi(\lambda^k_{i-1/2})\alpha^k_{i-1/2}]], \tag{34}$$

$$S = \operatorname{sgn}(\alpha^k_{i+1/2}), \tag{35}$$

and

$$\gamma^k_{i+1/2} = \frac{g^k_{i+1} - g^k_i}{\alpha^k_{i+1/2}}. \tag{36}$$

It is essentially an upwind-weighted minmod function.

It is important that the eigenvectors of the modified Jacobians be properly normalized to ensure that they remain well defined and form a complete set for various degenerate cases. One form of the normalized eigenvectors has been published previously [18]. Here, we have used a slightly different normalization developed by Balsara and Roe [19]. The details of how their normalization handles all the possible degenerate cases are contained in the above reference and are not repeated here. However, for completeness, the eigenvectors in this form are shown in the appendix.

3.2. *LU-SGS Relaxation Scheme*

As mentioned previously, the LU-SGS scheme is based on a simpler first-order approximation to the hyperbolic fluxes, rather than on Harten's higher order discretization described above. The first-order approximation, denoted by $\hat{\mathbf{R}}_h$, is actually a flux-vector splitting [20] of the hyperbolic fluxes. This splitting of the fluxes is closely related to Harten's first-order upwind form. If we set ε in Eq. (32) to zero, then Harten's first order flux becomes

$$\mathbf{F}_{i+1/2} = \tfrac{1}{2}\left[\mathbf{F}_{i+1} + \mathbf{F}_i - \sum_k |\lambda|^k_{i+1/2}\alpha^k_{i+1/2}\mathbf{r}^k_{i+1/2}\right]. \tag{37}$$

Equivalently, this can also be expressed as

$$\mathbf{F}_{i+1/2} = \tfrac{1}{2}[\mathbf{F}_{i+1} + \mathbf{F}_i - |\mathbf{A}|_{i+1/2}(\mathbf{Q}_{i+1} - \mathbf{Q}_i)]. \tag{38}$$

Here,

$$|\mathbf{A}| \equiv \mathbf{A}^+ - \mathbf{A}^-, \tag{39}$$

where

$$\mathbf{A}^\pm \equiv \mathbf{X}^{-1}\,\mathbf{\Lambda}^\pm\mathbf{X} \tag{40}$$

and $\mathbf{\Lambda}^\pm$ is a matrix having either all positive or all negative eigenvalues of $\tilde{\mathbf{A}}$ along its diagonal.

Flux-vector splitting was developed for the Euler equations, which, for the ideal gas equation of state, have the property that the flux function is a homogenous function of degree one in $\mathbf{Q}$ [21] and thus can be written as

$$\mathbf{F} \equiv \mathbf{A}\mathbf{Q}. \tag{41}$$

For the MHD equations, this property no longer holds. However, in this work we have found that it is a good enough approximation to form the basis of a relaxation scheme. That is, we say that

$$\mathbf{F} \approx \mathbf{A}\mathbf{Q}, \tag{42}$$

so that we can define

$$\mathbf{F}^{\pm} \equiv \mathbf{A}^{\pm}\mathbf{Q}. \tag{43}$$

Using this type of splitting is equivalent to evaluating $|\mathbf{A}|$ in Eq. (38) at the cell centers i and $i + 1$, rather than at the cell interface, $i + 1/2$. With this change, Eq. (38) becomes

$$\mathbf{F}_{i+1/2} = \tfrac{1}{2}[\mathbf{F}_{i+1} + \mathbf{F}_i - (|\mathbf{A}|_{i+1}\mathbf{Q}_{i+1} - |\mathbf{A}|_i\mathbf{Q}_i)]. \tag{44}$$

However, since $|\mathbf{A}| = \mathbf{A}^+ - \mathbf{A}^-$, the flux simplifies to

$$\mathbf{F}_{i+1/2} = \mathbf{A}^-_{i+1}\mathbf{Q}_{i+1} + \mathbf{A}^+_i\mathbf{Q}_i = \mathbf{F}^-_{i+1} + \mathbf{F}^+_i. \tag{45}$$

In general, $\hat{\mathbf{R}}_h$ can be written as

$$(\hat{\mathbf{R}}_h)_{ij} = (\mathbf{F})_{i+1/2,j} - (\mathbf{F})_{i-1/2,j} + (\mathbf{G})_{i,j+1/2} - (\mathbf{G})_{i,j-1/2}. \tag{46}$$

Substituting Eq. (44) and similar expressions for $\mathbf{F}_{i-1/2}$, $\mathbf{G}_{i+1/2}$, and $\mathbf{G}_{i-1/2}$ into Eq. (46), we get

$$\begin{aligned}(\hat{\mathbf{R}}_h)_{ij} = {} & \mathbf{F}^+_{ij} - \mathbf{F}^+_{i-1,j} + \mathbf{F}^-_{i+1,j} - \mathbf{F}^-_{ij} \\ & + \mathbf{G}^+_{ij} - \mathbf{G}^+_{i,j-1} + \mathbf{G}^-_{i,j+1} - \mathbf{G}^-_{ij}.\end{aligned} \tag{47}$$

Next, we form $\partial\hat{\mathbf{R}}_h/\partial\mathbf{Q}$ and substitute it into the left-hand side of Eq. (16) to obtain

$$\begin{aligned}\left\{\frac{3\mathbf{I}}{2\Delta t} + \mathbf{A}^+_{ij} - \mathbf{A}^+_{i-1,j} + \mathbf{A}^-_{i+1,j} - \mathbf{A}^-_{ij} + \mathbf{B}^+_{ij} - \mathbf{B}^+_{i,j-1}\right. \\ \left. + \mathbf{B}^-_{i,j+1} - \mathbf{B}^-_{i,j}\right\} \times \Delta\mathbf{Q}^n_{ij} = -\mathbf{R}^n_{ij}.\end{aligned} \tag{48}$$

where all of the terms on the right-hand side of the equations have been lumped together in $\mathbf{R}$. Here, $\mathbf{B}$ is the Jacobian of $\mathbf{G}$. This is a block pentadiagonal matrix, and the blocks themselves are fairly difficult to evaluate. To simplify the matrix, we approximate $\mathbf{A}^{\pm}$ and $\mathbf{B}^{\pm}$ as

$$\mathbf{A}^{\pm} \approx \tfrac{1}{2}(\tilde{\mathbf{A}} \pm \rho_A\mathbf{I}) \tag{49}$$

$$\mathbf{B}^{\pm} \approx \tfrac{1}{2}(\tilde{\mathbf{B}} \pm \rho_B\mathbf{I}), \tag{50}$$

where ρ_A is the maximum eigenvalue of $\tilde{\mathbf{A}}$, which is simply $v_x + c_f$, and ρ_B is defined similarly. The result of this approximation is to reduce the convergence rate of the relaxation because the operator is less closely coupled to the detailed wave structure of the right hand-side fluxes. However, the reduction in work per iteration more than offsets the reduction in convergence rate. An important point here is that the approximation to the implicit operator does not reduce the accuracy of the solution at each time step. As long as the scheme converges at each time step, the accuracy of the solution is determined by the time step size and the right-hand side fluxes.

With this approximation,

$$\mathbf{A}^+ - \mathbf{A}^- = \rho_A\mathbf{I}, \tag{51}$$

so that Eq. (48) simplifies to

$$\{\mathbf{D}_{ij} + \mathbf{A}^-_{i+1,j} + \mathbf{B}^-_{i,j+1} - \mathbf{A}^+_{i-1,j} - \mathbf{B}^+_{i,j-1}\}\,\Delta\mathbf{Q}^n_{ij} = -\mathbf{R}^n_{ij}, \tag{52}$$

where

$$\mathbf{D}_{ij} = \left(\frac{3}{2\Delta t} + \rho_A + \rho_B\right)\mathbf{I}_{ij}. \tag{53}$$

This block matrix equation can be solved in two steps using a forward Gauss–Seidel sweep followed by a backward sweep. The resulting algorithm can be written as

$$\begin{aligned}\{\mathbf{D}_{ij} - \mathbf{A}^+_{i-1,j} - \mathbf{B}^+_{i,j-1}\}\{\mathbf{D}_{ij} + \mathbf{A}^-_{i+1,j} + \mathbf{B}^-_{i,j+1}\} \\ \times \Delta\mathbf{Q}^m_{ij} = -(\mathbf{D}_{ij})\mathbf{R}^m_{ij}.\end{aligned} \tag{54}$$

The forward sweep is equivalent to inverting a lower block diagonal matrix (the first braced term in Eq. (54)), and the backward sweep is equivalent to inverting an upper block diagonal matrix (second braced term in Eq. (54)). That is, the operator has been split according to the sign of the eigenvalues. Note that this is not a directional splitting. The fluxes in the different directions are evaluated simultaneously. The appeal of this scheme is that it does not require any block matrix inversions, since the blocks along the diagonal, $\mathbf{D}$, contain only diagonal elements. Thus, a single LU-SGS iteration requires only slightly more (about 10% more) computations than a single explicit time step. We note here that the evaluation of the parabolic fluxes, $\mathbf{R}_p$, at the old iteration level (m) in Eq. (16) was motivated by the desire to achieve this diagonal form, since linearization of $\mathbf{R}^{m+1}_p$ would have introduced some off-diagonal terms. The price for this simplification is that for low Re_A or Lu, the time step may be limited by the numerical stability of the parabolic terms. However, for Re_A and Lu much larger than one, the CFL number is not limited

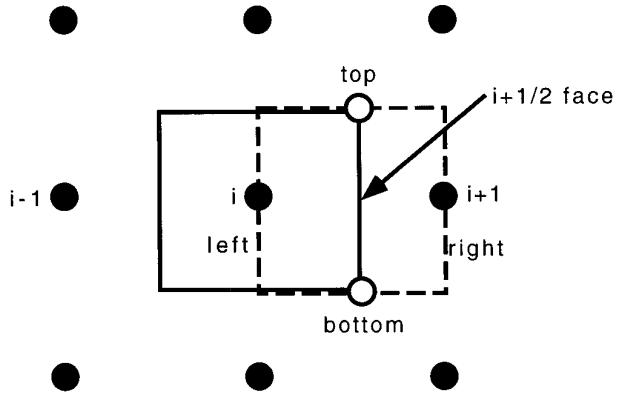

FIG. 1. Grid stencil for parabolic fluxes. Dashed box is the offset mesh used to determine flux at $i + 1/2$ face.

by numerical stability. This simple structure also lends itself well to parallelization using domain decomposition. In addition, the relaxation scheme is highly vectorizable if we sweep through the computational domain along lines of constant $i + j$ (in 2D), so that each term along these lines is independent of the others and depends only on data that has already been updated during the current sweep.

3.3. *Parabolic Terms*

The parabolic fluxes are differenced using a second-order accurate finite-volume scheme. The discretization of the parabolic fluxes, $\mathbf{R}_p$, is given by

$$(\mathbf{R}_p)_{ij} = (\mathbf{F}_p)_{i+1/2,j} - (\mathbf{F}_p)_{i-1/2,j} + (\mathbf{G}_p)_{i,j+1/2} - (\mathbf{G}_p)_{i,j-1/2}. \tag{55}$$

The face-centered fluxes that appear in the expression for $\mathbf{R}_p$ are obtained by performing a flux balance on an offset, face-centered mesh. Figure 1 shows the nine-point grid stencil for a uniformly spaced Cartesian grid. The solid dots show the locations of the cell centers. The solid lines indicate the boundaries of the cell surrounding point (i, j). The dashed lines indicate the offset mesh used to calculate the parabolic flux at the $i + \frac{1}{2}$ face. The faces of the offset mesh are labeled top, bottom, left, and right. To illustrate how the offset mesh is used to obtain the interface flux, we consider as an example the second term of the $\mathbf{F}_p$ vector, τ_{xx}, which, for two dimensions, is given by

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right), \tag{56}$$

where μ is the dynamic viscosity. The derivatives at the cell face are

$$\frac{\partial v_x}{\partial x} = \frac{[(v_x)_{i+1} A_{\text{right}} - (v_x)_i A_{\text{left}}]}{(\text{Vol})_{\text{fc}}} \tag{57}$$

and

$$\frac{\partial v_y}{\partial y} = \frac{[(v_y)_{\text{top}} A_{\text{top}} - (v_y)_{\text{bottom}} A_{\text{bottom}}]}{(\text{Vol})_{\text{fc}}}, \tag{58}$$

where $(\text{Vol})_{\text{fc}}$ is the volume of the face-centered cell, A_{right} is the area of its right face, and so forth. The variables at the top and bottom faces, which are denoted by open circles in the figure, are obtained by averaging over the four cells surrounding each face.

4. CODE VALIDATION TESTS

4.1. 1*D Coplanar MHD Riemann Problem*

This test problem was solved to verify that Harten's approximate Riemann solver worked for the MHD equations. One-dimensional ideal MHD (variations in x only) is described by a system of seven equations, since B_x is constant (in order to satisfy $\nabla \cdot \mathbf{B} = 0$). The coplanar MHD equations are obtained from the one-dimensional ideal MHD equations by setting B_z and v_z to zero, thus allowing only planar flow and fields. This eliminates the $v_x \pm c_{a,x}$ eigenvalues, leaving a system of five equations with five eigenvalues. Mathematically, the Riemann problem is an initial boundary value problem in which there is initially a discontinuous jump in the data such that the left half of the domain is at one state and the right half of the domain is at another state. As the solution evolves in time, shock waves and rarefaction waves form that travel at speeds related to the wave speeds of the system. Although not physically realizable in plasmas, this problem is analogous to a shock tube in hydrodynamics.

For the full five-wave case, there is not a closed form analytical solution. Instead, the solution must be checked by calculating generalized Riemann invariants across the rarefaction waves and Rankine–Hugoniot jump conditions across the shock waves. Since this has already been done by Brio and Wu [1] for a specific set of conditions, we used those same initial conditions in order to allow direct comparison with their published solution. The initial left state was $p = 1$, $\rho = 1$, and $B_y = 1$. The initial right state was $p = 0.1$, $\rho = 0.125$, and $B_y = -1$. The velocities were zero and B_x was 0.75. Figure 2 shows the initial density and transverse magnetic field distributions and their numerical solution after 400 time steps on an 800 point grid with a CFL number of 0.8. Since there was no advantage to using the implicit scheme for this problem (due to the small time steps required for accuracy), the solution was computed using a simple first-order (in time) explicit scheme given by

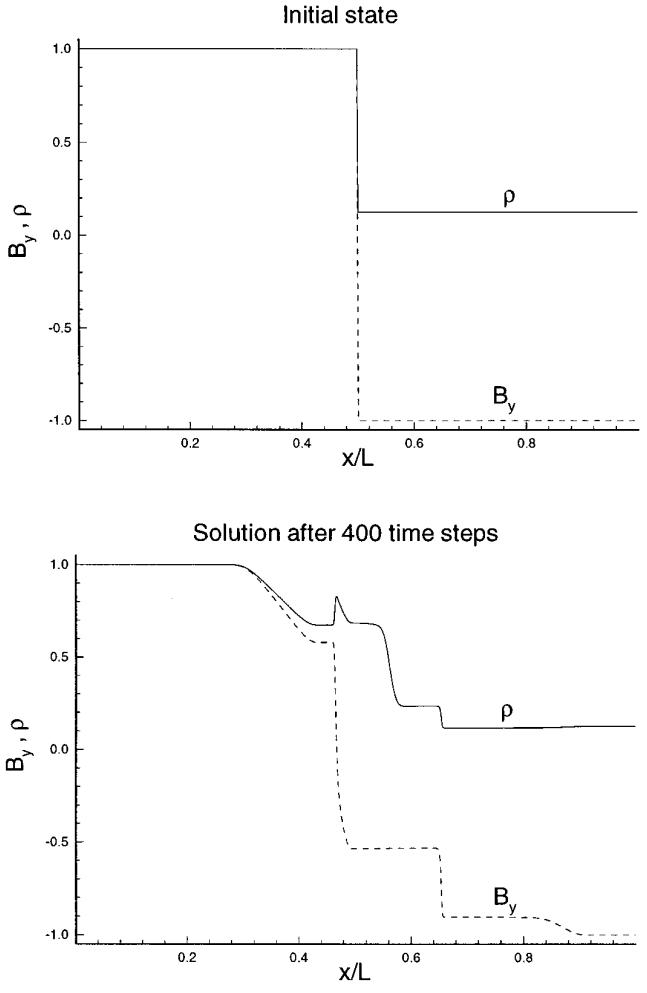

FIG. 2. Numerical solution of coplanar Riemann problem. Density and transverse magnetic field are shown initially and after solution has evolved for 400 time steps.

$$\frac{\mathbf{Q}_{ij}^{n+1} - \mathbf{Q}_{ij}^{n}}{\Delta t} = -(\mathbf{R}_h)_{ij}^{n}. \tag{59}$$

The solution clearly shows five waves formed corresponding to the five eigenvalues. They are a fast rarefaction wave, a slow shock, a contact surface moving to the right, a slow compound wave (rarefaction and shock), and a fast rarefaction wave moving to the left. Note that the numerical method is able to resolve the shocks over a few grid points without introducing numerical oscillations. The computed solution overlaid exactly on Brio and Wu's published solution.

4.2. *Oblique Shock*

Figure 3 shows the geometry and boundary conditions for the oblique shock test problem. This steady state problem served primarily as a test of our implementation of

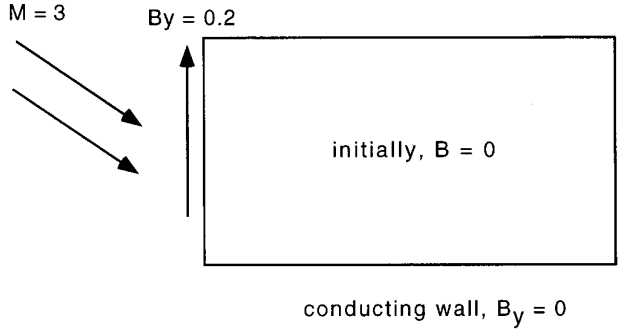

FIG. 3. Geometry and boundary conditions for oblique shock test problem.

the approximate Riemann solver in two dimensions. In this problem **v** and **B** are initially zero in the interior of the domain. A super-Alfvénic ($M = 3$), inviscid, perfectly conducting plasma flows in from the left and upper boundaries and impinges at an angle of 25° on a perfectly conducting plate that forms the bottom boundary. The incoming flow carries with it a vertical field of $B_y = 0.2$. The boundary conditions for the perfectly conducting plate are $v_y = 0$ and $B_y = 0$. In general, boundary conditions were implemented by using a single layer of ghost cells surrounding the computational domain. The boundaries were treated explicitly. That is, the ghost cells were evaluated using values from the previous iteration. So, for example, $B_y = 0$ at the conducting plate was achieved by setting

$$(B_y)_{\text{ghost cell}} = -(B_y)_{\text{first cell}}. \tag{60}$$

Figure 4 shows the steady state solution of this problem. Contours of density and magnetic field lines are plotted. The density contours show that an oblique shock forms, as expected. Outside of the shock, the field is convected in from the boundary. At the shock, the field lines bend due to the change in direction of the flow at the shock.

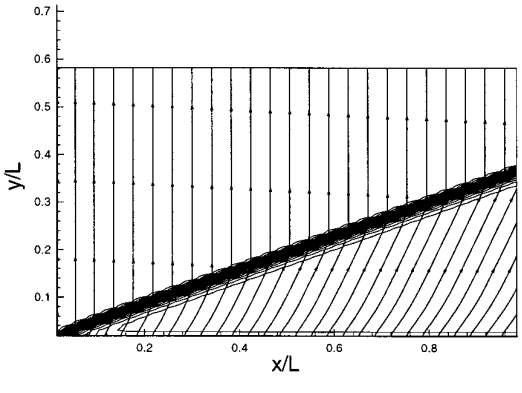

FIG. 4. Density contours and field lines for an $M = 3$ flow impinging on a perfectly conducting plate at an angle of 25°.

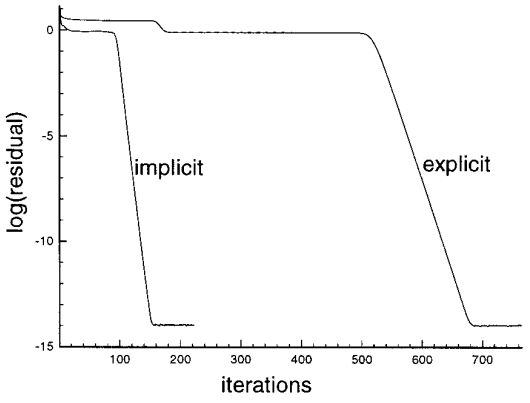

FIG. 5. Logarithm of the two-norm of the energy equation residual per cell plotted as a function of iteration number for explicit and implicit solutions of oblique shock problem.

We verified that the divergence was less than 10^{-14} throughout the domain.

This solution was obtained from the implicit scheme, Eq. (16), with the parabolic terms and the time derivative terms all set to zero, so that the scheme becomes

$$\frac{\partial(\hat{\mathbf{R}}_h)_{ij}^m}{\partial \mathbf{Q}}[\mathbf{Q}_{ij}^{m+1} - \mathbf{Q}_{ij}^m] = -(\mathbf{R}_h)_{ij}^m. \tag{61}$$

To determine the efficiency of the implicit scheme as a steady state solver, the oblique shock was also solved with the simple explicit scheme (Eq. (59)) at a CFL number of 0.8. It is recognized that Eq. (59) is not an optimized explicit scheme. It is used here simply to provide a rough benchmark for evaluating the implicit scheme. Figure 5 is a plot of the logarithm of the two-norm of the average residual of the energy equation as a function of the number of iterations (or time steps, in the case of the explicit scheme). In this case, the two-norm of the average residual is

$$\|R_{h,\mathrm{en}}\|_2 = \frac{\sqrt{\sum_i \sum_j (R_{h,\mathrm{en}})_{ij}^2}}{N_{\mathrm{cells}}}. \tag{62}$$

The implicit scheme converged to 10^{-14} in about 150 iterations, whereas the explicit scheme required about 700 time steps (iterations). Since one implicit iteration takes only about 10% more CPU time than one explicit time step, the implicit scheme required roughly four times less CPU time than the explicit scheme to converge to the steady state solution.

4.3. *Hartmann Flow*

The validation of the parabolic terms consisted of applying the code to the Hartmann flow problem, whose steady state solution can be solved analytically. The problem geometry is shown in Fig. 6. It consists of two infinite parallel plates, each moving in opposite directions at velocity V_0, and with an in-plane magnetic field B_o between them that is normal to the plates in the y direction.

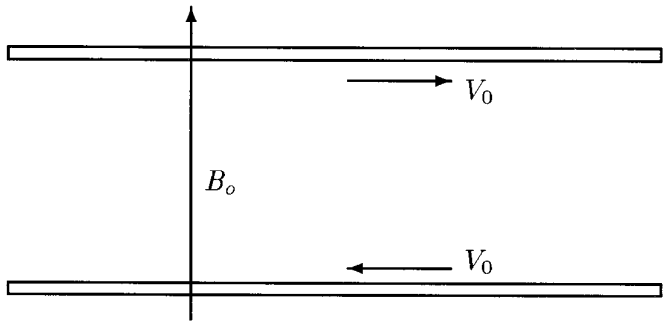

FIG. 6. The Hartmann flow geometry showing the moving parallel plates and the cross magnetic field.

For the case with equal and opposite plate velocities, the Hartmann flow is described by the differential equations

$$\frac{\partial^2 v_x}{\partial y^2} - \left(\frac{H^2}{L^2}\right) v_x = 0 \tag{63}$$

$$\frac{\partial B_x}{\partial y} = -(Rm)\, v_x, \tag{64}$$

where H, the Hartmann number, is

$$H \equiv \frac{B_o L}{\sqrt{\rho \nu \eta}} = \sqrt{(\mathrm{Re}_A)(\mathrm{Lu})}, \tag{65}$$

and Rm, the magnetic Reynolds number, is

$$\mathrm{Rm} \equiv \frac{\mu_o L V}{\eta}. \tag{66}$$

Figure 7 shows the results from a simulation with $H =$

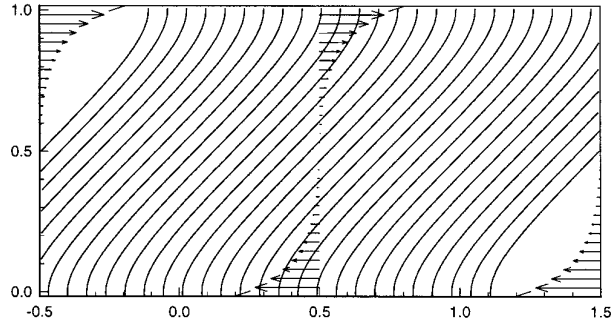

FIG. 7. Hartmann flow simulation with H = 10. Flow velocity vectors and magnetic field lines are shown. The flow velocity only exists close to the plates. The magnetic field lines are linear around the midplane.

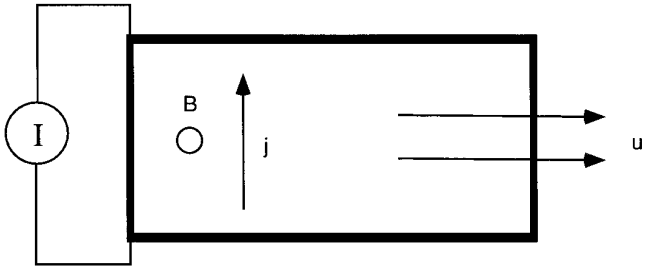

FIG. 8. Geometry of the magnetic field diffusion problem.

10. Near the plates there is a boundary layer with a scale length of L/H in which the velocity falls off rapidly to zero and the field develops a swayed shape as it is dragged by the fluid. In the center region of the channel the velocity is zero and the field has a uniform slope. In the limit of small Hartmann number the boundary layer extends to the opposite wall and a linear velocity profile develops. For large Hartmann numbers, the boundary layer shrinks to zero and the field has a uniform slope throughout the channel. The calculated solution converged to the analytical solution to within machine roundoff errors.

4.4. *Magnetic Field Diffusion*

The three problems described to this point show that the various pieces of the algorithm work correctly and that the implicit solver is an efficient relaxation scheme for steady problems. This next example shows the utility of the scheme for unsteady problems, which is the primary purpose of the algorithm. The problem geometry is shown in Fig. 8. In this figure, positive x is to the right, positive y is up, and positive z is out of the page. Initially, $\mathbf{v}$ and $\mathbf{B}$ are zero throughout the computational domain, which is indicated by the thick-lined rectangle. A current sheet, with a total current per unit z of I, is applied to the left boundary ($x = 0$). The current density is

$$j_y = -\frac{\partial B_z}{\partial x}. \tag{67}$$

The current sheet at the left boundary is applied by setting $B_z = \mu_o I$ in the ghost cells, so that the initial current sheet is represented by the jump in B_z between the ghost cell and the first interior cell. The resulting $\mathbf{j} \times \mathbf{B}$ force accelerates the plasma in the positive x direction, so that it flows out at the right boundary. The upper and lower boundaries are perfectly conducting walls, whose boundary conditions were discussed previously. There is no viscosity, so the problem is one-dimensional in x.

For a Lundquist number much larger than one, the plasma will be accelerated up to some exit velocity on a fast hydromagnetic time scale, while the magnetic field and current will diffuse into the domain on a slower time scale related to the resistive diffusion time. It is assumed for the purposes of this demonstration problem that we are only interested in the dynamics of the field diffusion and are willing to smear over some of the details of the faster dynamics by advancing the solution at a large CFL number. This problem was solved with the time-accurate implicit scheme at a CFL number of 100. Recall that the time-accurate implicit scheme is

$$\left[\frac{\partial (\mathbf{R}_h)_{ij}^m}{\partial \mathbf{Q}} + \frac{3\mathbf{I}}{2\Delta t}\right] \Delta \mathbf{Q}_{ij}^m = -\left[(\mathbf{R}_h)_{ij}^m + (\mathbf{R}_p)_{ij}^m + \left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ij}^m\right]. \tag{68}$$

For purposes of making a rough comparison, the problem was also solved at a CFL number of 1 using the first-order explicit scheme given by

$$\frac{\mathbf{Q}_{ij}^{n+1} - \mathbf{Q}_{ij}^n}{\Delta t} = -[(\mathbf{R}_h)_{ij}^n + (\mathbf{R}_p)_{ij}^n]. \tag{69}$$

Figure 9 shows the evolution of the magnetic field at a Lundquist number of 100 for the two simulations. The upper plot shows the results from the explicit scheme, while the lower plot shows the results from the implicit scheme. The diffusion of the magnetic field is captured equally well at the larger CFL number, as expected. The explicit simulation took 2600 time steps to advance the solution to $t = 10.17$, while the implicit simulation required 26 time steps. However, the implicit scheme required approximately 30 iterations at each time step, and each iteration required about 10% more CPU time than an explicit time step. So, in terms of CPU time, the implicit scheme ran roughly three times faster than the explicit scheme for this problem.

5. SUMMARY

We presented a new implicit algorithm for solving the nonideal MHD equations. The algorithm is designed for solving problems at high Lundquist and Reynolds numbers. For this class of problems, it is stable for any CFL number. The algorithm features an approximate Riemann solver for the hyperbolic terms. The approximate Riemann solver combines Powell's multidimensional technique with Harten's discretization of the hyperbolic fluxes. The parabolic terms are discretized with a finite volume technique that uses an offset, face-centered mesh to calculate the interface fluxes. The implicit operator is inverted by using the LU-SGS iteration.

We then showed the results of several code validation test cases. The first was a one-dimensional MHD Riemann problem that verified that Harten's fluxes were correctly

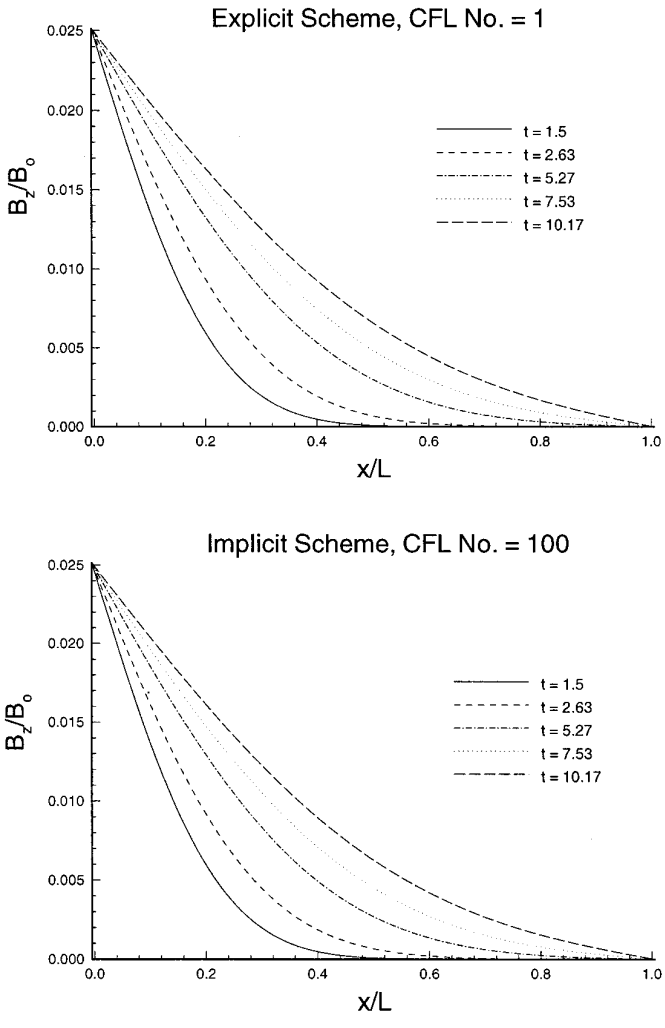

FIG. 9. Comparison of explicit and implicit solutions of magnetic field diffusion problem.

implemented. The second was an oblique shock problem that demonstrated that the approximate Riemann solver worked in two dimensions. It also showed that the implicit scheme was roughly four times more efficient at relaxing to the steady state oblique shock solution than a particular explicit scheme. A Hartmann flow problem was solved to validate the resistive and viscous terms. Finally, a transient problem involving the diffusion of a magnetic field demonstrated that the implicit technique could accurately track the field diffusion while taking time steps 100 times larger than allowed by the explicit scheme, which translated into a factor of 3 in CPU savings when the work required for the implicit iterations was accounted for.

The chief advantage of this algorithm over an explicit approximate Riemann solver is the flexibility in choosing the time step. That is, if one is following dynamics that are occurring on time scales much longer than the shortest MHD time scales, then one can select the time step based on the desired accuracy, rather than on the numerical stability.

APPENDIX

In this appendix the normalized eigenvectors of the modified Jacobian, $\tilde{\mathbf{A}}$, are presented. The eigenvectors for the y direction fluxes can easily be derived from these. The eigenvectors have a simpler form when written in terms of the primitive variables, $\mathbf{W}$, rather than the conserved variables, $\mathbf{Q}$, where

$$\mathbf{W} = (\rho, v_x, v_y, v_z, B_x, B_y, B_z, p)^{\mathrm{T}}. \tag{70}$$

Therefore, we present here the eigenvectors of the primitive variable Jacobian, $\tilde{\mathbf{A}}_p$, which is related to $\tilde{\mathbf{A}}$ through

$$\tilde{\mathbf{A}}_p = \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}\right]^{-1} \tilde{\mathbf{A}} \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}\right]. \tag{71}$$

The eigenvectors are related through

$$\mathbf{r}_{\mathrm{c}} = \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}\right] \mathbf{r}_{\mathrm{p}} \tag{72}$$

and

$$\mathbf{l}_{\mathrm{c}} = \mathbf{l}_{\mathrm{p}} \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}\right]^{-1}, \tag{73}$$

where p refers to the primitive variable form and c refers to the conserved variable form.

Following Roe and Balsara [19], we introduce the dimensionless paramaters

$$\alpha_f^2 = \frac{a^2 - c_s^2}{c_f^2 - c_s^2}, \tag{74}$$

$$\alpha_s^2 = \frac{c_f^2 - a^2}{c_f^2 - c_s^2}, \tag{75}$$

$$\beta_y = \frac{B_y}{B_\perp}, \tag{76}$$

$$\beta_z = \frac{B_z}{B_\perp}, \tag{77}$$

where

$$B_\perp = \sqrt{B_y^2 + B_z^2}. \tag{78}$$

For certain degenerate cases, these parameters are indeter-

minate. For those cases, special limiting values can be defined, as discussed in detail in Roe and Balsara.

The eigenvalues of $\tilde{\mathbf{A}}_p$ (and $\tilde{\mathbf{A}}$) are

$$\lambda = (v_x, v_x, v_x \pm c_f, v_x \pm c_s, v_x \pm c_{a,x})^{\mathrm{T}}. \tag{79}$$

One of the v_x eigenvalues of $\tilde{\mathbf{A}}_p$ corresponds to the entropy wave, and its normalized left and right eigenvectors are

$$\mathbf{l}_{\mathrm{ent}} = (1, 0, 0, 0, 0, 0, 0, 0), \tag{80}$$

$$\mathbf{r}_{\mathrm{ent}} = (1, 0, 0, 0, 0, 0, 0, 1/a^2)^{\mathrm{T}}. \tag{81}$$

The other v_x eigenvalue is a result of the modification to the Jacobian, and thus corresponds to the transport of the divergence of $\mathbf{B}$. Its normalized eigenvectors are

$$\mathbf{l}_{\mathrm{div}} = (0, 0, 0, 0, 1, 0, 0, 0), \tag{82}$$

$$\mathbf{r}_{\mathrm{div}} = (0, 0, 0, 0, 1, 0, 0, 0)^{\mathrm{T}}. \tag{83}$$

The Alfvén wave eigenvectors are also very simple. They are

$$\mathbf{l}_A = \tfrac{1}{2}(0, 0, \pm\beta_z, \mp\beta_y, 0, -\beta_z \mathrm{sgn}(B_x)/\sqrt{\rho}, \beta_y \mathrm{sgn}(B_x)/\sqrt{\rho}, 0), \tag{84}$$

$$\mathbf{r}_A = (0, 0, \pm\beta_z, \mp\beta_y, 0, -\beta_z \mathrm{sgn}(B_x)\sqrt{\rho}, \beta_y \mathrm{sgn}(B_x)\sqrt{\rho}, 0)^{\mathrm{T}}. \tag{85}$$

The eigenvectors for the fast magnetosonic wave are

$$\mathbf{l}_{\mathrm{f}}^{\pm} = \frac{1}{2a^2} \begin{bmatrix} 0 \\ \pm\alpha_{\mathrm{f}} c_{\mathrm{f}} \\ \mp\alpha_{\mathrm{s}} c_{\mathrm{s}} \beta_y \mathrm{sgn}(B_x) \\ \mp\alpha_{\mathrm{s}} c_{\mathrm{s}} \beta_z \mathrm{sgn}(B_x) \\ 0 \\ \alpha_{\mathrm{s}} a \beta_y / \sqrt{\rho} \\ \alpha_{\mathrm{s}} a \beta_z / \sqrt{\rho} \\ \alpha_{\mathrm{f}} / \rho \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{r}_{\mathrm{f}}^{\pm} = \begin{bmatrix} \alpha_{\mathrm{f}} \rho \\ \pm\alpha_{\mathrm{f}} c_{\mathrm{f}} \\ \mp\alpha_{\mathrm{s}} c_{\mathrm{s}} \beta_y \mathrm{sgn}(B_x) \\ \mp\alpha_{\mathrm{s}} c_{\mathrm{s}} \beta_z \mathrm{sgn}(B_x) \\ 0 \\ \alpha_{\mathrm{s}} a \beta_y \sqrt{\rho} \\ \alpha_{\mathrm{s}} a \beta_z \sqrt{\rho} \\ \alpha_{\mathrm{f}} \rho a^2 \end{bmatrix}. \tag{86}$$

The slow magnetosonic eigenvectors are obtained from the above expressions by interchanging α_{s} and α_{f} and by replacing c_{s} with c_{f}.

ACKNOWLEDGMENTS

The authors thank K. G. Powell, B. van Leer, and P. L. Roe for their helpful discussions regarding the implementation of the approximate Riemann solver. This work was supported by the U.S. Air Force Office of Scientific Research (AFOSR) under Contract AFOSR-F49620-95-0126.

REFERENCES

1. M. Brio and C. C. Wu, *J. Comput. Phys.* **75,** 400 (1988).
2. A. L. Zachery and P. Colella, *J. Comput. Phys.* **99,** 341 (1992).
3. A. L. Zachery, A. Malagoli, and P. Colella, *SIAM J. Sci. Comput.* **15**(2), 263 (1994).
4. K. G. Powell, ICASE Report No. 94-24 (1994).
5. W. Dai and P. R. Woodward, *J. Comput. Phys.* **111,** 354 (1994).
6. W. Dai and P. R. Woodward, *J. Comput. Phys.* **115,** 485 (1994).
7. W. Dai and P. R. Woodward, *J. Comput. Phys.* **121,** 51 (1995).
8. H. P. Furth, J. Killeen, and M. N. Rosenbluth, *Phys. Fluids* **6,** 479 (1963).
9. I. Lindemuth and J. Killeen, *J. Comput. Phys.* **13,**, 181 (1973).
10. D. Schnack and J. Killeen, *J. Comput. Phys.* **35,** 110 (1980).
11. W. J. Weber, J. P. Boris, and J. H. Gardner, *Comput. Phys. Commun.* **16,** 243 (1979).
12. U. Shumlak, T. W. Hussey, G. J. Marklin, and R. E. Peterkin, *Bull. Am. Phys. Soc.* **38,** 1894 (1993).
13. D. D. Schnack, D. C. Barnes, Z. Mikic, D. S. Harned, and E. J. Caramana, *J. Comput. Phys.* **70,** 330 (1987).
14. H. Ok and D. S. Eberhardt, "Solution of Unsteady Incompressible Navier-Stokes Equations Using an LU Decomposition Scheme," AIAA-91-1611, in *AIAA Fluid Dynamics, Plasma Dynamics, and Lasers Conference,* 1991.
15. S. Yoon and A. Jameson, *AIAA J.* **26,** 1025 (1988).
16. A. Harten, *J. Comput. Phys.* **49,** 357 (1983).
17. H. C. Yee, R. F. Warming, and A. Harten, *J. Comput. Phys.* **57,** 327 (1985).
18. T. I. Gambosi, K. G. Powell, and D. L. De Zeeuw, *J. Geophys. Res.* **99,** 21525 (1994).
19. P. L. Roe and D. S. Balsara, *SIAM J. Appl. Math.*, **56,** 57 (1996).
20. T. J. Barth, "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," AIAA-87-0595, in *AIAA Aerospace Sciences Meeting,* 1987.
21. J. L. Steger and R. F. Warming, *J. Comput. Phys.* **40,** 263 (1981).