Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 188 (2003) 573–592
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A 2D high-β Hall MHD implicit nonlinear solver

L. Chacón *, D.A. Knoll

T-15 Theoretical Plasma Physics, MS K717, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 28 November 2002; received in revised form 17 March 2003; accepted 18 March 2003

Abstract

A nonlinear, fully implicit solver for a 2D high-β (incompressible) Hall magnetohydrodynamics (HMHD) model is proposed. The task in non-trivial because HMHD supports the whistler wave. This wave is dispersive ($\omega \sim k^2$) and therefore results in diffusion-like numerical stability limits for explicit time integration methods. For HMHD, implicit approaches using time steps above the explicit numerical stability limits result in diagonally submissive Jacobian systems. Such systems are difficult to invert with iterative techniques. In this study, Jacobian-free Newton–Krylov iterative methods are employed for a fully implicit, nonlinear integration, and a semi-implicit (SI) preconditioner strategy, developed on the basis of a Schur complement analysis, is proposed. The SI preconditioner transforms the coupled hyperbolic whistler system into a fourth-order, parabolic, diagonally dominant PDE, amenable to iterative techniques. Efficiency and accuracy results are presented demonstrating that an efficient fully implicit implementation (i.e., faster than explicit methods) is indeed possible *without* sacrificing numerical accuracy.

Keywords: Hall MHD; Implicit differencing; Newton–Krylov; Jacobian-free; Nonlinear PDE; Schur complement

1. Introduction

Hall magnetohydrodynamics (HMHD) treats ions and electrons as separate species, allowing relative drifts between them. This is of importance in instances where ions become demagnetized in a thin plasma boundary layer due to ion inertia or finite ion Larmor radius effects, but electrons remain magnetized. Ion demagnetization results in the decoupling of the ion and electron flows within the boundary layer, and the standard resistive magnetohydrodynamics (MHD) model (which treats ions and electrons as a single species) breaks down.

Numerically, the HMHD formalism is extremely stiff due to the presence of dispersive waves, characterized by dispersion relations of the form $\omega \sim k^2$, with ω is the frequency and k is the wavenumber. In explicit integration methods, such dispersive waves impose Courant–Friedrichs–Lewy (CFL) time step limits for numerical stability that grow quadratically with the mesh increment size. Since HMHD physics results in the formation of extremely thin boundary layers that need to be resolved, it follows that

* Corresponding author. Tel.: +1-505-665-6973; fax: +1-505-665-7150.
E-mail address: chacon@lanl.gov (L. Chacón).

0021-9991/03/$ - see front matter
doi:10.1016/S0021-9991(03)00193-1

extraordinary computing resources are required for acceptable explicit simulation turn-around times. Further, if sufficiently long time scales are of interest, the question of error propagation in explicit computations becomes important due to the large number of time steps ($\sim 10^8$–10^9) required per simulation.

Implicit methods promise to alleviate these issues by decoupling the time step and mesh increment sizes. However, the presence of dispersive waves in HMHD complicates the already difficult endeavor of developing fully implicit solvers for coupled, nonlinear PDE systems. In fact, to the authors' knowledge, the only other comparable effort for HMHD is Ref. [1], where a semi-implicit solver for Hall MHD is proposed. In fact, the semi-implicit operator proposed there is very similar to one of the preconditioning flavors proposed in this work. However, the semi-implicit solver proposed in [1] is implemented with important simplifications in the magnetic field dependence of the semi-implicit operator to streamline its linear algebra treatment (in a similar fashion as in Ref. [20]), and relies substantially on splitting and linearization. These simplifications should be avoided to preserve accuracy with large implicit time steps, in view of the analysis and results presented in [2].

In this study, we propose a fully implicit, nonlinear algorithm for HMHD, based on the Newton–Raphson method for nonlinear convergence, and on Krylov semi-iterative techniques for the required algebraic inversions. A fully implicit formulation ensures linear numerical stability and preserves accuracy by avoiding splitting and/or linearization (thus preserving the character of the continuum formulation in discrete form except for discretization errors [2]). However, the question still remains about efficiency. The latter depends fundamentally on the conditioning of the algebraic (Jacobian) systems involved. Jacobian conditioning is poor for HMHD due to the disparate time scales involved (which results in equally disparate eigenvalues) and the diagonally submissive nature of matrices stemming from hyperbolic PDEs (for implicit time steps larger than the CFL limit). The latter aspect is particularly limiting for implicit formulations, since it renders the matrices unsuitable for an iterative treatment using standard iterative techniques (which is a fundamental aspect of the implementation of implicit solvers in multi-dimensional applications).

Jacobian-free Newton–Krylov (JFNK) iterative methods, however, are suitable for a fully implicit integration of such systems because they allow preconditioning to accelerate convergence. Preconditioning effectively improves the condition number of the Jacobian system by using suitable approximations of the Jacobian inverse. A key feature of preconditioning is that these approximations do not affect the quality of the converged solution, but only the rate of convergence of the Krylov method. Here, we focus on physics-based preconditioning, which employs semi-implicit techniques to reformulate diagonally submissive hyperbolic systems into diagonally dominant parabolic ones. This idea has successfully been employed in [3–5] to deal with stiff hyperbolic systems. Specifically, Chacón et al. [3] successfully developed a physics-based preconditioner to deal with the linear Alfvén wave in resistive MHD. We will follow here a similar strategy for the development of an efficient implicit solver for the whistler wave in HMHD.

Additionally, we will provide an important connection between the physics-based semi-implicit preconditioning approach and a more general algebraic concept: the Schur complement [6]. Briefly, we will show that the semi-implicit operator resulting from the parabolization of a hyperbolic system can be directly interpreted as a Schur complement. This connection generalizes the physics-based concept and provides a solid framework to develop physics-based preconditioners for other stiff-wave applications.

The rest of the paper is organized as follows. Section 2 introduces the specific HMHD of interest. Section 3 discusses the implementation details of the JFNK solver employed. The physics-based preconditioner strategy is derived in Section 4. Numerical results on efficiency and accuracy are presented in Section 5. Finally, we conclude in Section 6.

2. Hall MHD model equations

In our study, we employ an incompressible, constant density 2D HMHD model in slab geometry, using a streamfunction/vorticity formulation. This model is rigorously valid for cold ions ($T_i \ll T_e$) in a large-β

plasma (where β is the ratio of the kinetic pressure to the magnetic pressure). We use Alfvénic units, in which lengths are normalized to $L = L_y$, velocities to the Alfvén speed $v_A = B_{p0}/\sqrt{\rho_0\mu_0}$ (with B_{p0} is the maximum in-plane magnetic field, ρ_0 is the density, and μ_0 is the magnetic permeability), and time to the Alfvén time $\tau_A = L/v_A$. Neglecting variations along the z-axis ($\partial_z = 0$), a 2D streamfunction/vorticity formulation of HMHD reads:

$$\nabla^2 \Phi = \omega, \tag{1}$$

$$(\partial_t + \vec{v}_p \cdot \nabla - \eta\nabla^2 + \nu_e\nabla^4)\Psi + E_0 = d_i\vec{B}_p \cdot \nabla B_z, \tag{2}$$

$$(\partial_t + \vec{v}_p \cdot \nabla - \eta\nabla^2 + \nu_e\nabla^4)B_z + S_{B_z} = \vec{B}_p \cdot \nabla(v_z - d_i J_z), \tag{3}$$

$$(\partial_t + \vec{v}_p \cdot \nabla - \nu\nabla^2)v_z + S_{v_z} = \vec{B}_p \cdot \nabla B_z, \tag{4}$$

$$(\partial_t + \vec{v}_p \cdot \nabla - \nu\nabla^2)\omega + S_\omega = \vec{B}_p \cdot \nabla J_z, \tag{5}$$

where Φ is the in-plane ion velocity streamfunction ($\vec{v}_p = \vec{z} \times \nabla\Phi$), ω is the out-of-plane vorticity ($\omega = \vec{z} \cdot \nabla \times \vec{v}$), Ψ is the in-plane flux function (which gives $\vec{B}_p = \vec{z} \times \nabla\Psi$), $J_z = \nabla^2\Psi$ is the ion current in the ignorable direction, and v_z, B_z are the ion velocity and the magnetic field components in the ignorable direction. Sources E_0 (the applied electric field in the z-direction), S_ω, S_{B_z}, and S_{v_z} can be included to ensure that the initial condition is an equilibrium. The parameter $d_i = c/\omega_{pi}$ is the ion skin depth (with ω_{pi} is the ion plasma frequency and c is the speed of light), which quantifies the importance of Hall MHD effects: MHD is recovered in the limit $d_i \to 0$. The transport parameters (the ion kinematic viscosity ν, the resistivity η, and the electron kinematic viscosity – otherwise known as hyperresistivity – ν_e) are assumed constant. A numerical value is used for ν_e that ensures adequate damping of the whistler wave at the shortest grid scales (and hence controls numerical noise buildup in nonlinear regimes). The derivation is in Appendix A, and yields:

$$\nu_e = 0.2 d_i v_A h^2, \tag{6}$$

where $h = (\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2})^{-1/2}$, and Δx, Δy are the grid spacings in the x and y-directions, respectively. It is important to note that ν_e is "grid-bound" (i.e., it varies with the mesh spacing h). This is a consequence of the requirement that the whistler damping occur at the shortest grid scales. Since $\nu_e \propto h^2$, the hyperresistivity operator $\nu_e\nabla^4$ scales as h^{-2} when discretized, and its numerical stiffness is then comparable to that of a second-order diffusion operator. This is in fact a desirable property for both implicit and explicit solvers, since it avoids unnecessary strain on the solvers when using very fine meshes while still controlling the numerical noise.

A linear analysis of Eqs. (1)–(5) using an equilibrium defined by a constant in-plane magnetic field, no in-plane ion flow, zero perpendicular velocity and magnetic field, and no transport ($\eta = \nu = \nu_e = 0$) yields the following dispersion relation:

$$\omega^2 = v_A^2 k_\parallel^2 \left\{ 1 + \frac{d_i^2 k^2}{2} \pm d_i k \sqrt{1 + \frac{d_i^2 k^2}{4}} \right\}$$

with v_A is the Alfvén speed, k is the wavenumber, and $k_\parallel = \vec{B}_p \cdot \vec{k}/B_p$. This dispersion relation has the following limits:

1. $d_i k \ll 1$ (resistive MHD): then $\omega^2 \approx v_A^2 k_\parallel^2$, and the system decouples into two separate Alfvén waves, one propagated by the ω–Φ–Ψ equations and the other by the v_z–B_z equations.

2. $d_i k \gg 1$ (Hall MHD): in this case the two Alfvén waves bifurcate into the whistler wave and the ion cyclotron wave, as follows:

$$\omega = \begin{cases} v_A k_\parallel d_i k & \text{(Whistler)}, \\ \frac{k_\parallel}{k}\omega_{ci} & \text{(Ion cyclotron)}. \end{cases}$$

The quadratic nature in k of the whistler wave dispersion relation is apparent from this result, whereas the ion cyclotron wave effectively scales as $\mathcal{O}(k^0)$.

The presence of the dispersive whistler wave introduces a very stringent CFL condition for explicit numerical algorithms. The motivation of this work is to develop an efficient, fully implicit (θ-scheme), nonlinear algorithm that is able to step over the whistler normal mode time scale while preserving accuracy.

The system of equations (1)–(5) is discretized in a similar fashion to Ref. [3], using second-order finite volumes in space, and a θ-scheme in time, with $\theta = 1/2$ (fully implicit, second-order Crank–Nicolson). The domain chosen is a rectangle $L_x \times L_y$ in Cartesian geometry with periodic boundary conditions in the x-direction and homogeneous Dirichlet in the y-direction in all quantities.

3. Jacobian-free Newton–Krylov solver

Once the HMHD equations are discretized in time and space, the next step is to find the new-time solution $\vec{x}^{n+1} = \{\vec{\Phi}^{n+1}, \vec{\Psi}^{n+1}, \overrightarrow{B_z}^{n+1}, \overrightarrow{v_z}^{n+1}, \vec{\omega}^{n+1}\}^T$ from the current-time solution $\vec{x}^n$ by solving the nonlinear, coupled system of equations resulting from the fully implicit discretization, and symbolized here by $\vec{G}(\vec{x}^{n+1}) = \vec{0}$ (where $\vec{G} = \{\vec{G}_\Phi, \vec{G}_\Psi, \vec{G}_{B_z}, \vec{G}_{v_z}, \vec{G}_\omega\}^T$).

This is accomplished iteratively with the Newton–Raphson algorithm, which requires the solution of a series of algebraic systems of the form:

$$J_k \delta\vec{x}_k = -\vec{G}(\vec{x}_k). \tag{7}$$

Here, k is the nonlinear iteration level, $J_k = (\partial\vec{G}/\partial\vec{x})_k$ is the Jacobian matrix, $\vec{x}_k$ is the kth state vector, $\delta\vec{x}_k$ is the kth Newton update [from which the $(k+1)$th Newton state vector is obtained, $\vec{x}_{k+1} = \vec{x}_k + \delta\vec{x}_k$], $\vec{G}(\vec{x}_k)$ is the vector of residuals. Nonlinear convergence is achieved when:

$$\|\vec{G}(\vec{x}_k)\|_2 < \epsilon_a + \epsilon_r \|\vec{G}(\vec{x}_0)\|_2 = \epsilon_t, \tag{8}$$

where $\|\cdot\|_2$ is the ℓ_2-norm (Euclidean norm), $\epsilon_a = N \times 10^{-15}$ (with N the total number of mesh points) is an absolute tolerance to avoid converging below roundoff, ϵ_r is the Newton relative convergence tolerance (set to 10^{-4} in this work), and $\vec{G}(\vec{x}_0)$ is the initial residual. Upon convergence, the solution at the new time step is found as $\vec{x}^{n+1} = \vec{x}_{k+1}$.

Each of these iterative steps requires inverting the Jacobian system in Eq. (7). Krylov semi-iterative techniques [7] are ideally suited for this task, because they can be implemented Jacobian-free (i.e., the full Jacobian in never formed nor stored) and can be preconditioned for efficiency. A Jacobian-free implementation exploits the feature that Krylov methods only require the product of the system matrix times a Krylov vector $\vec{v}$, which is provided by the iterative algorithm, to proceed. In Newton's method, the Jacobian-vector product can be calculated using the directional (Gateaux) derivative, approximated here as:

$$J_k\vec{v} \approx \frac{\vec{G}(\vec{x}_k + \epsilon\vec{v}) - \vec{G}(\vec{x}_k)}{\epsilon}, \tag{9}$$

where ϵ is small but finite (discussed later in this section). Thus, the evaluation of the Jacobian-vector product only requires the function evaluation $\vec{G}(\vec{x}_k + \epsilon\vec{v})$, and there is no need to form or store the Jacobian matrix.

Among the various Krylov methods available, GMRES (Generalized Minimal RESiduals) is selected because it guarantees convergence with nonsymmetric, nonpositive definite systems [8] (the case here because of the hyperbolic nature of HMHD), and because it provides normalized Krylov vectors $|\vec{v}| = 1$, thus bounding the error introduced in the difference approximation of Eq. (9) (whose leading error term is proportional to $\epsilon|\vec{v}|^2$) [9]. However, GMRES can be memory intensive (storage increases linearly with the number of GMRES iterations per Jacobian solve) and expensive (computational complexity of GMRES increases with the square of the number of GMRES iterations per Jacobian solve). Restarted GMRES can in principle deal with these limitations; however, it lacks a theory of convergence, and stalling is frequently observed in real applications [10]. Here, we focus on minimizing the number of GMRES iterations per Jacobian solve for efficiency, by: (1) using inexact Newton techniques [11], and (2) improving the condition number of the Jacobian matrix by preconditioning the problem.

The inexact Newton method adjusts the GMRES convergence tolerance at every Newton iteration according to the size of the Newton residual, as follows:

$$\|J_k\delta\vec{x}_k + \vec{G}(\vec{x}_k)\|_2 < \zeta_k\|\vec{G}(\vec{x}_k)\|_2, \tag{10}$$

where ζ_k is the inexact Newton parameter or forcing term. Thus, the convergence tolerance of GMRES is loose when the state vector $\vec{x}_k$ is far from the nonlinear solution, but becomes increasingly tighter as $\vec{x}_k$ approaches the solution. Superlinear convergence rate of the inexact Newton method is possible if the sequence of ζ_k is chosen properly [12]. Here, we employ the following prescription (similar to what is proposed in Section 6.3 of [12]):

$$\begin{aligned}
\zeta_k^A &= \gamma\left(\frac{\|\vec{G}(\vec{x}_k)\|_2}{\|\vec{G}(\vec{x}_{k-1})\|_2}\right)^{\alpha},\\
\zeta_k^B &= \min[\zeta_{\max}, \max(\zeta_k^A, \gamma\zeta_{k-1}^{\alpha})],\\
\zeta_k &= \min\left[\zeta_{\max}, \max\left(\zeta_k^B, \gamma\frac{\epsilon_t}{\|\vec{G}(\vec{x}_k)\|_2}\right)\right],
\end{aligned}$$

with $\alpha = 1.5$, $\gamma = 0.9$, and $\zeta_{\max} = 0.5$. The convergence tolerance ϵ_t is defined in Eq. (8). In this prescription, the first step ensures superlinear convergence (for $\alpha > 1$), the second avoids volatile decreases in ζ_k, and the last avoids oversolving in the last Newton iteration.

Preconditioning consists in operating on the system matrix J_k with an operator P_k^{-1} (preconditioner) such that $J_kP_k^{-1}$ (right preconditioning) or $P_k^{-1}J_k$ (left preconditioning) is well-conditioned. In this study, we use right preconditioning because the vector of residuals $\vec{G}$ is not polluted by the preconditioner. This is straightforward to see when considering the equivalent Jacobian system:

$$(J_kP_k^{-1})(P_k\delta\vec{x}_k) = -\vec{G}(\vec{x}_k). \tag{11}$$

Thus, GMRES will solve:

$$(J_kP_k^{-1})\vec{z} = -\vec{G}(\vec{x}_k), \tag{12}$$

and the Newton update $\delta\vec{x}_k$ is found upon obtaining $\vec{z}$ from $\delta\vec{x}_k = P_k^{-1}\vec{z}$. Notice that the system in Eq. (11) is equivalent to the original system for any nonsingular operator P_k^{-1}. Thus, the choice of P_k^{-1} does not affect the accuracy of the final solution, but crucially determines the rate of convergence of GMRES, and hence the efficiency of the algorithm.

To solve Eq. (12) using GMRES, it is required to compute the Jacobian-vector product $(J_kP_k^{-1})\vec{v}_j$ (where $\vec{v}_j$ is the Krylov vector of the jth Krylov iteration) to proceed. This is implemented with two matrix-vector products:

1. Compute $\vec{y} = P_k^{-1}\vec{v}_j$. This is the so-called preconditioning step. Often, P_k^{-1} is not an exact inverse of any particular matrix, but an approximate inverse – obtained, for instance, using operator splitting and/or low-complexity MG methods – of the exact Jacobian, or even an approximate inverse of an approximation of the Jacobian. The specifics of the formation of the preconditioner operator P_k^{-1} for this application are discussed in Section 4.
2. Compute $J_k\vec{y}$ using the Jacobian-free approximation: $J_k\vec{y} \approx \{\vec{G}(\vec{x}_k + \epsilon\vec{y}) - \vec{G}(\vec{x}_k)\}/\epsilon$, where ϵ is small but finite. Newton convergence is insensitive to ϵ within a 2–3 orders of magnitude window; ϵ is calculated here as:

$$\epsilon = 10^{-5}\left(1 + \frac{\|\vec{x}_k\|_2}{\|\vec{y}\|_2}\right).$$

The first step determines the efficiency of the algorithm (and leaves room for exploration, since P_k^{-1} is in principle an arbitrary nonsingular operator), while the second step determines the accuracy of the solution [according to the discretization of the nonlinear system $\vec{G}(\vec{x}^{n+1}) = \vec{0}$]. In this particular JFNK implementation, we also use the preconditioner P_k^{-1} to provide a good initial guess for the GMRES solver, $\delta\vec{x}_0 = P_0^{-1}[-\vec{G}(\vec{x}_0)]$.

To maximize efficiency, the preconditioning operator P_k^{-1} should approximate the inverse of the Jacobian J_k while being relatively inexpensive. Here, we propose *physics-based methods*, based on semi-implicit techniques to deal with wave stiffness [3–5]. The next section describes in more detail the nature of the approximations employed here to construct the physics-based preconditioner. We emphasize that these approximations have no bearing on the accuracy of the Newton-converged solution (which is fully implicit and nonlinear), but only on the convergence rate of GMRES in each Newton step.

4. "Physics-based" preconditioner

Implicit differencing ensures absolutely stable numerical descriptions, for any time step and level of mesh refinement, by introducing dispersion in waves and by treating elliptic operators (such as diffusion) nonlocally. However, some of the mechanisms that are sources of numerical instabilities in explicit methods continue to manifest themselves in implicit schemes in the form of ill-conditioned algebraic systems, which iterative techniques have difficulty in handling. (Direct solvers are in principle suitable to deal with poorly conditioned matrices; however, they do not scale adequately for sparse systems in 2D and 3D [13,14].)

There are two sources of ill-conditioning in the system of MHD equations: elliptic operators and hyperbolic couplings. The former manifests itself in a power scaling N^α (with $\alpha > 1$) of the computational complexity of iterative solver techniques. Elliptic stiffness is dealt with here with multigrid preconditioning (MG), which employs low-complexity multilevel solvers [15] to invert the elliptic operators approximately. The multilevel aspect of MG (which employs a "divide and conquer" approach by which the different scales of the global solution are decoupled in multiple grids of varying mesh refinement) results, as a solver, in an optimal $\mathcal{O}(N)$ scaling of the computational complexity [16], and, as a preconditioner, in a number of Krylov iterations virtually independent of the problem size (see [3,15,17] and results in Section 5).

Ill-conditioning from hyperbolic couplings manifests itself in a loss of diagonal dominance due to short-wavelength harmonics when the implicit time step is larger than the explicit wave CFL limit (see [3] for an in-depth explanation of this issue). It is possible, however, to reformulate the physical equations so that the resulting algebraic systems are better conditioned. The basic idea is to produce a well-conditioned (diagonally-dominant) parabolic operator from an ill-conditioned hyperbolic system of equations [3]. The procedure can be understood easily with a first-order hyperbolic linear system:

$$\partial_t u = \partial_x v,$$
$$\partial_t v = \partial_x u.$$

Differencing implicitly in time (with backward Euler for simplicity), we have:

$$u^{n+1} = u^n + \Delta t \partial_x v^{n+1}, \tag{13}$$

$$v^{n+1} = v^n + \Delta t \partial_x u^{n+1}. \tag{14}$$

It is now possible to substitute the second equation into the first to obtain the following parabolic equation:

$$(I - \Delta t^2 \partial_{xx}) u^{n+1} = u^n + \Delta t \partial_x v^n, \tag{15}$$

which is equivalent to the set of two discretized equations, but much better conditioned because the parabolic operator is diagonally dominant. A similar idea is behind the method of differential approximations [18], and the semi-implicit solvers developed in the MHD context [1,19,20] (although there the semi-implicit operator is obtained and implemented in an ad-hoc manner).

It is useful at this point, using this simple example, to establish an important connection between the parabolic semi-implicit operator obtained above and the useful concept of the Schur complement. We start by writing Eqs. (13) and (14) in matrix form:

$$\begin{bmatrix} I & -\Delta t \partial_x \\ -\Delta t \partial_x & I \end{bmatrix} \begin{bmatrix} u^{n+1} \\ v^{n+1} \end{bmatrix} = \begin{bmatrix} u^n \\ v^n \end{bmatrix}.$$

The matrix can be factorized as follows:

$$\begin{bmatrix} I & -\Delta t \partial_x \\ -\Delta t \partial_x & I \end{bmatrix} = \begin{bmatrix} I & -\Delta t \partial_x \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{SC} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\Delta t \partial_x & I \end{bmatrix},$$

where $P_{SC} = I - \Delta t^2 \partial_x^2$ is the so-called Schur complement. The connection between the semi-implicit operator an the Schur complement is now obvious. The factorized matrix is trivial to invert (pre- and post-triangular matrices are trivially invertible in an exact manner – they correspond to forward elimination and backward substitution, respectively – and only the block diagonal matrix requires an iterative treatment), and yields Eq. (15) as a result. The usefulness of this association for the present application will be demonstrated shortly. We note at this point that the semi-implicit preconditioner proposed in [3] for the Alfvén wave in resistive MHD can also be formulated as a Schur complement [21]. Schur complement approaches have also been shown to be very effective as preconditioners in other applications [22].

The goal here is to find a semi-implicit formulation for the HMHD equations that removes the stiffness associated with the whistler wave in the preconditioning stage. Although the general guiding principle is the same as in the simple example above, specifics of the HMHD model (such as the presence of advection and diffusion) make this task difficult. In particular, due to the dispersive nature of the whistler wave, the parabolic system obtained will be of fourth-order, thus requiring special approaches for its efficient inversion. The next sections describe these issues further.

4.1. Approximate formulation of the HMHD system

For preconditioning purposes, instead of attempting to derive a semi-implicit formulation of the HMHD system valid for arbitrary implicit times steps (which would require approximating the full formulation in Eqs. (1)–(5)), we restrict ourselves to implicit time steps Δt that satisfy the ordering:

$$\Delta t^{\mathrm{w}}_{\mathrm{CFL}} \ll \Delta t \lesssim \Delta t_{\mathrm{A}}, \tag{16}$$

where $\Delta t^{\mathrm{w}}_{\mathrm{CFL}}$ is the explicit CFL condition associated with the dispersive whistler wave, and Δt_{A} is the CFL limit associated with the Alfvén speed (for subAlfvénic flows) or the flow speed (for superAlfvénic flows). Owing to the dispersive nature of the whistler wave, Eq. (16) implies that $C_1 h^2 \ll \Delta t \lesssim C_2 h$. While restricting our preconditioner to the condition in Eq. (16) effectively limits the implicit time step for efficiency, large CPU speedups over purely explicit approaches are nevertheless expected (and realized, see Section 5) for sufficiently refined grids by virtue of the different scalings in h. In addition, Hall physics typically allows fast dynamical processes with time scales comparable to the Alfvén time scale in the system, and hence implicit time steps of the order of Δt_{A} are not so limiting in practice.

The time step ordering in Eq. (16) is the first piece of physics insight that allows us to simplify the HMHD system for the purpose of developing an effective preconditioner. The complete Jacobian matrix for the HMHD system reads, in block symbolic form (with the algebraic system of equations $\vec{G}(\vec{x}) = \vec{0}$ ordered first by grid nodes and then by equations):

$$J_k = \begin{bmatrix} D_\Phi & 0 & 0 & 0 & I \\ L_{\Phi,\Psi} & D_\Psi & U_{B_z,\Psi} & 0 & 0 \\ L_{\Phi,B_z} & L_{\Psi,B_z} & D_{B_z} & U_{v_z,B_z} & 0 \\ L_{\Phi,v_z} & L_{\Psi,v_z} & L_{B_z,v_z} & D_{v_z} & 0 \\ L_{\Phi,\omega} & L_{\Psi,\omega} & 0 & 0 & D_\omega \end{bmatrix}. \tag{17}$$

For the ordering chosen in Eqs. (1)–(5), the Jacobian is almost a lower block triangular system, except for three blocks (I, $U_{B_z,\Psi}$, and U_{v_z,B_z}), which correspond to wave couplings and are responsible for the wave dispersion relation in Section 2. In the limit $d_{\mathrm{i}} \to 0$, the system supports two Alfvén waves, propagated by I, $L_{\Phi,\Psi}$, and $L_{\Psi,\omega}$ on the one hand, and U_{v_z,B_z} and L_{B_z,v_z} on the other. The time step ordering chosen earlier allows us to neglect the upper-triangular blocks I and U_{v_z,B_z} responsible for the Alfvén waves. This renders the following simplified Jacobian matrix:

$$P_k = \begin{bmatrix} D_\Phi & 0 & 0 & 0 & 0 \\ L_{\Phi,\Psi} & D_\Psi & U_{B_z,\Psi} & 0 & 0 \\ L_{\Phi,B_z} & L_{\Psi,B_z} & D_{B_z} & 0 & 0 \\ L_{\Phi,v_z} & L_{\Psi,v_z} & L_{B_z,v_z} & D_{v_z} & 0 \\ L_{\Phi,\omega} & L_{\Psi,\omega} & 0 & 0 & D_\omega \end{bmatrix}. \tag{18}$$

The inversion of this Jacobian is reduced to the inversion of three diagonal blocks D_Φ, D_{v_z}, and D_ω, and the 2×2 diagonal block matrix,

$$\begin{bmatrix} D_\Psi & U_{B_z,\Psi} \\ L_{\Psi,B_z} & D_{B_z} \end{bmatrix}. \tag{19}$$

This block matrix is in fact the one responsible of the propagation of the whistler wave, and is amenable to a Schur decomposition, yielding:

$$\begin{bmatrix} D_\Psi & U_{B_z,\Psi} \\ L_{\Psi,B_z} & D_{B_z} \end{bmatrix} = \begin{bmatrix} I & 0 \\ L_{\Psi,B_z} D_\Psi^{-1} & I \end{bmatrix} \begin{bmatrix} D_\Psi & 0 \\ 0 & P^{Bz}_{\mathrm{SC}} \end{bmatrix} \begin{bmatrix} I & D_\Psi^{-1} U_{B_z,\Psi} \\ 0 & I \end{bmatrix}, \tag{20}$$

where $P^{Bz}_{\mathrm{SC}} = D_{B_z} - L_{\Psi,B_z} D_\Psi^{-1} U_{\Psi,B_z}$ is the Schur complement. Thus, the inversion of Eq. (19) is reduced to the inversion of the diagonal block D_Ψ and of the Schur complement P^{Bz}_{SC}. The subscript B_z in the Schur complement denotes that this decomposition is not unique, since we can also write:

$$\begin{bmatrix} D_\Psi & U_{B_z,\Psi} \\ L_{\Psi,B_z} & D_{B_z} \end{bmatrix} = \begin{bmatrix} I & U_{B_z,\Psi}D_{B_z}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{\mathrm{SC}}^{\Psi} & 0 \\ 0 & D_{B_z} \end{bmatrix} \begin{bmatrix} I & 0 \\ D_{B_z}^{-1}L_{\Psi,B_z} & I \end{bmatrix}, \tag{21}$$

where $P_{\mathrm{SC}}^{\Psi} = D_\Psi - U_{\Psi,B_z}D_{B_z}^{-1}L_{\Psi,B_z}$. Which Schur complement is preferable is not obvious at this stage, and requires specific knowledge of the physics terms involved. This is the goal of the next section.

4.2. Formulation of the semi-implicit preconditioner

The ultimate goal of this exercise is to develop an inexpensive, approximate Jacobian inverse. We have already accomplished much along this line, since we have simplified the inversion of the whole Jacobian into the inversion of a few diagonal blocks and a Schur complement. While the diagonal blocks (which contain advection–diffusion terms) are relatively easy to invert using approximate multigrid (MG) techniques (as in [3]), the Schur complements (semi-implicit operators) in their current form are not suitable for a MG treatment yet, since the formulation of the Schur operator itself requires the inversion of D_Ψ (or D_{B_z}).

At this point, it is useful to define the symbolic operators that play a role in the Schur decomposition. These are:

$$\begin{aligned}
L_{\Psi,B_z} &= \theta\Big\{ -\vec{\omega}_0\cdot\nabla + d_{\mathrm{i}}[(\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2 - (\vec{z}\times\nabla J_{z0})\cdot\nabla]\Big\},\\
U_{\Psi,B_z} &= -\theta d_{\mathrm{i}}\vec{B}_{\mathrm{p}0}\cdot\nabla,\\
D_\Psi &= \frac{1}{\Delta t} + \theta[\vec{v}_{\mathrm{e}0}\cdot\nabla - \eta\nabla^2 + v_{\mathrm{e}}\nabla^4],\\
D_{B_z} &= \frac{1}{\Delta t} + \theta[\vec{v}_{\mathrm{p}0}\cdot\nabla - \eta\nabla^2 + v_{\mathrm{e}}\nabla^4],
\end{aligned}$$

where $\vec{\omega}_0 = \nabla\times\vec{v}_0$ is the ion vorticity, $\vec{v}_{\mathrm{e}0} = \vec{v}_{\mathrm{p}0} - d_{\mathrm{i}}\vec{J}_0$ is the electron velocity, and $\theta = 1/2$ is the θ-scheme parameter for Crank–Nicolson. The first important simplification in the Schur complement operators is to approximate $D_\Psi^{-1}\sim\Delta t$, $D_{B_z}^{-1}\sim\Delta t$ (one could also keep other diagonal entries from upwinded advection and diffusion, although this is not done here). This is consistent with the time step ordering in Eq. (16) (except for the electron viscosity term), and yields:

$$\begin{aligned}
P_{\mathrm{SC}}^{Bz} &\approx D_{B_z} - \Delta t L_{\Psi,B_z}U_{\Psi,B_z},\\
P_{\mathrm{SC}}^{\Psi} &\approx D_\Psi - \Delta t U_{\Psi,B_z}L_{\Psi,B_z}.
\end{aligned}$$

Further, and also consistently with the time step ordering, we neglect the $\vec{\omega}_0\cdot\nabla$ flow term in L_{Ψ,B_z} to find:

$$\begin{aligned}
P_{\mathrm{SC}}^{Bz} &\approx D_{B_z} + \Delta t\theta^2 d_{\mathrm{i}}^2[(\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2 - (\vec{z}\times\nabla J_{z0})\cdot\nabla](\vec{B}_{\mathrm{p}0}\cdot\nabla),\\
P_{\mathrm{SC}}^{\Psi} &\approx D_\Psi + \Delta t\theta^2 d_{\mathrm{i}}^2(\vec{B}_{\mathrm{p}0}\cdot\nabla)[(\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2 - (\vec{z}\times\nabla J_{z0})\cdot\nabla].
\end{aligned}$$

These results show, in either case, the fourth-order nature of the whistler semi-implicit operator. These can be simplified further by either approximating $(\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2 - (\vec{z}\times\nabla J_{z0})\cdot\nabla \approx \nabla^2(\vec{B}_{\mathrm{p}0}\cdot\nabla)$ [3] or by neglecting the $(\vec{z}\times\nabla J_{z0})\cdot\nabla$ term (which does not contribute to the propagation of the whistler wave). With these approximations, we have two possibilities for each Schur complement:

$$[(\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2 - (\vec{z}\times\nabla J_{z0})\cdot\nabla](\vec{B}_{\mathrm{p}0}\cdot\nabla) \approx \begin{cases} (\vec{B}_{\mathrm{p}0}\cdot\nabla)\nabla^2(\vec{B}_{\mathrm{p}0}\cdot\nabla) \\ \nabla^2(\vec{B}_{\mathrm{p}0}\cdot\nabla)^2 \end{cases},$$

for $P_{\mathrm{SC}}^{B_z}$, and

$$(\vec{B}_{\mathrm{p0}}\cdot\nabla)[(\vec{B}_{\mathrm{p0}}\cdot\nabla)\nabla^2-(\vec{z}\times\nabla J_{z0})\cdot\nabla]\approx\begin{cases}(\vec{B}_{\mathrm{p0}}\cdot\nabla)\nabla^2(\vec{B}_{\mathrm{p0}}\cdot\nabla)\\(\vec{B}_{\mathrm{p0}}\cdot\nabla)^2\nabla^2\end{cases},$$

for P_{SC}^{Ψ}. A priori, we disregard the form $\nabla^2(\vec{B}_{\mathrm{p0}}\cdot\nabla)^2$, because it violates a fundamental property of the forcing terms in Eqs. (1)–(5), namely, that they have no average effect along magnetic field lines (because $\oint\frac{dl}{B}\vec{B}\cdot\nabla=0$). The whistler semi-implicit operator $\nabla^2(\vec{B}_{\mathrm{p0}}\cdot\nabla)^2$ does not satisfy this property, and, if used in the preconditioner, would result in noise being fed in the null space of the forcing terms. Such noise can only be filtered by the external GMRES solver, spoiling the efficiency of the algorithm.

The other two alternatives, however, do satisfy this property, and are both good candidates for the whistler semi-implicit operator. At this point, we focus our discussion on the Ψ Schur complement, since it allows both suitable forms of the semi-implicit operator. The choice at this point is guided by implementation and performance issues. Here, we will consider both operators, since each has advantages and disadvantages, and each excels over the other one depending on the problem at hand. The preconditioner strategy based on P_{SC}^{Ψ} is straightforwardly obtained by unrolling the Schur decomposition in Eq. (21) in the block inversion of the approximate Jacobian system, $\delta\vec{x}\approx P_k^{-1}[-\vec{G}(\vec{x}_k)]$, where P_k is defined in Eq. (18), and is summarized as follows:

$$\begin{aligned}
&\delta\Phi\approx D_{\Phi}^{-1}(-G_{\Phi}),\\
&\delta\Psi\approx\left(P_{\mathrm{SC}}^{\Psi}\right)^{-1}\Big[-\theta\delta\vec{v}_{\mathrm{p}}\cdot\nabla\Psi_0+\theta d_{\mathrm{i}}(\vec{B}_{\mathrm{p0}}\cdot\nabla)D_{\Psi}^{-1}[-\theta\delta\vec{v}_{\mathrm{p}}\cdot\nabla B_{z0}-G_{B_z}]-G_{\Psi}\Big],\\
&\delta B_z\approx D_{B_z}^{-1}\Big[-\theta\delta\vec{v}_{\mathrm{p}}\cdot\nabla B_{z0}-\theta d_{\mathrm{i}}[(\vec{B}_{\mathrm{p0}}\cdot\nabla)\nabla^2\delta\Psi+\delta\vec{B}_{\mathrm{p}}\cdot\nabla J_{z0}]+\theta\delta\vec{B}_{\mathrm{p}}\cdot\nabla v_{z0}-G_{B_z}\Big],\\
&\delta v_z\approx D_{v_z}^{-1}\Big[-\theta\delta\vec{v}_{\mathrm{p}}\cdot\nabla v_{z0}+\theta\delta\vec{B}_{\mathrm{p}}\cdot\nabla B_{z0}+\theta\vec{B}_{\mathrm{p0}}\cdot\nabla\delta B_z-G_{v_z}\Big],\\
&\delta\omega\approx D_{\omega}^{-1}\Big[-\theta\delta\vec{v}_{\mathrm{p}}\cdot\nabla\omega_0+\theta(\vec{B}_{\mathrm{p0}}\cdot\nabla)\nabla^2\delta\Psi+\theta\delta\vec{B}_{\mathrm{p}}\cdot\nabla J_{z0}-G_{\omega}\Big],
\end{aligned}$$

where $\delta\vec{v}_{\mathrm{p}}=\vec{z}\times\nabla\delta\Phi$, $\delta\vec{B}_{\mathrm{p}}=\vec{z}\times\nabla\delta\Psi$, and

$$\begin{aligned}
&D_{\Phi}=\nabla^2,\\
&D_{v_z}=D_{\omega}=\frac{1}{\Delta t}+\theta[\vec{v}_{\mathrm{p0}}\cdot\nabla-\nu\nabla^2].
\end{aligned}$$

The generalization of the operation of P_k^{-1} on a generic Krylov vector (as required by the GMRES algorithm) is straightforward. We stress that the inversion of the full Jacobian matrix in Eq. (17) has been simplified to the formation of the right-hand side terms and the inversion of the diagonal blocks D_{Φ}, D_{B_z}, D_{v_z}, D_{ω}, and the whistler semi-implicit operator (Schur complement) P_{SC}^{Ψ}. All these diagonal blocks are by design parabolic or elliptic operators, and hence amenable to MG methods. The diagonal blocks D_{v_z} and D_{ω} are inverted using approximate scalar MG techniques as described in [3]. This preconditioning strategy will be effective for $\Delta t_{\mathrm{CFL}}^{\mathrm{w}}\ll\Delta t\lesssim\Delta t_{\mathrm{A}}$ (Eq. (16)).

Despite the tremendous simplification that the previous algorithm embodies, the issue of dealing with several fourth-order operators is still unresolved. Specifically, as formulated, the preconditioner needs to deal with the anisotropic fourth-order operator in the Schur complement plus the isotropic biharmonic associated with the electron viscosity. The following sections deal with these and other implementation issues.

4.2.1. Implementation issues of $(\vec{B}_{p0}\cdot\nabla)\nabla^2(\vec{B}_{p0}\cdot\nabla)$

This form of the whistler semi-implicit operator is positive definite and contains the most physics (because the $(\vec{z}\times\nabla J_{z0})\cdot\nabla$ term has not been completely neglected). However, a MG treatment is challenging because it cannot be formulated as two second-order coupled systems (see next section), and its strong

anisotropy (due to the $\vec{B}_{\mathrm{p0}} \cdot \nabla$ operators) has so far hindered the development of an effective smoother. Furthermore, its discretization stencil requires, depending on the specifics, a minimum of 21 points, making its sparse implementation cumbersome and expensive.

To avoid these issues, we opt to invert the Schur complement associated with this semi-implicit operator, given by:

$$\left[\frac{1}{\Delta t} + \theta\vec{v}_{\mathrm{e0}} \cdot \nabla - \theta\eta\nabla^2 + \theta v_{\mathrm{e}}\nabla^4 + \Delta t\theta^2 d_{\mathrm{i}}^2(\vec{B}_{\mathrm{p0}} \cdot \nabla)\nabla^2(\vec{B}_{\mathrm{p0}} \cdot \nabla)\right]\delta\Psi = \mathrm{rhs}_\Psi \tag{22}$$

approximately by operator splitting, as follows:

$$\left[\frac{1}{\Delta t} + \Delta t\theta^2 d_{\mathrm{i}}^2(\vec{B}_{\mathrm{p0}} \cdot \nabla)\nabla^2(\vec{B}_{\mathrm{p0}} \cdot \nabla)\right]\delta\Psi^* = \mathrm{rhs}_\Psi,$$

$$\left[\frac{1}{\Delta t} + \theta(\vec{v}_{\mathrm{e0}} \cdot \nabla - \eta\nabla^2)\right]\delta\Psi^{**} = \frac{\delta\Psi^*}{\Delta t},$$

$$\left(\frac{1}{\Delta t} + \theta v_{\mathrm{e}}\nabla^4\right)\delta\Psi^{n+1} = \frac{\delta\Psi^{**}}{\Delta t}.$$

The first stage is symmetric positive definite (SPD), and is approximately inverted using unpreconditioned CG (Conjugate Gradient) to a relative tolerance of 10^{-3} (iterating further is useless due to the errors introduced by the many approximations involved in deriving this algorithm). The second stage can be effectively dealt with by approximate MG, in the same manner as in [3]. The final stage is dealt with a few (about 20) passes of Jacobi; the rationale behind this choice is that the purpose of the electron viscosity term is to smooth noise generated nonlinearly by the whistler terms, and the electron viscosity coefficient is "grid-bound" (Eq. (6)) and designed to act on the shortest grid scales. The inversion of the diagonal block D_{B_z} (which also contains the electron viscosity term) is performed using the last two steps of the split algorithm above. From now on, this preconditioning alternative will be identified by CGSI (CG semi-implicit).

CGSI does not require forming any matrix for the fourth-order operators, since Jacobi can be implemented matrix-light [3,15] and unpreconditioned CG can be implemented matrix-free. (A *matrix-light* implementation requires one to form and store the main diagonal only, as opposed to a *matrix-free* implementation, which does not store any element of the matrix.) This represents a substantial simplification in the coding of the algorithm, and significant memory savings. We note in passing that the use of CG inside the preconditioner stage requires using flexible GMRES [23] (FGMRES) on the outside. FGMRES has higher memory requirements than GMRES, since two sets of Krylov vector subsets need to be stored, but requires one less preconditioner execution per GMRES call. This saves CPU time when the preconditioner represents a large fraction of the CPU time per time step, as is the case here.

While the simplicity of CGSI is appealing, the preconditioner will likely fail with large d_{i} (since the coefficients of both fourth-order operators increase with d_{i}) and/or with very fine grids. Failure will occur due to inefficiencies in the unpreconditioned CG step (which is not scalable, as the number of CG iterations scale with some power of the number of unknowns – see [14] and results in Section 5) and to errors in the splitting algorithm itself. CGSI is expected to work well for moderate d_{i} and moderately refined grids.

4.2.2. Implementation issues of $(\vec{B}_{p0} \cdot \nabla)^2\nabla^2$

Contrary to CGSI, this whistler semi-implicit operator can be reformulated into two coupled second-order systems. The Schur complement associated with this semi-implicit operator is given by:

$$\left[\frac{1}{\Delta t}+\theta\vec{v}_{e0}\cdot\nabla-\theta\eta\nabla^2+\theta v_e\nabla^4+\Delta t\theta^2 d_i^2(\vec{B}_{p0}\cdot\nabla)^2\nabla^2\right]\delta\Psi=\mathrm{rhs}_\Psi, \tag{23}$$

which can be rewritten as:

$$\left[\frac{1}{\Delta t}+\theta\vec{v}_{e0}\cdot\nabla-\theta\eta\nabla^2\right]\delta\Psi+\left[\frac{v_e}{\Delta t\theta d_i^2}\nabla^2+(\vec{B}_{p0}\cdot\nabla)^2\right]\xi=\mathrm{rhs}_\Psi,$$
$$\Delta t\theta^2 d_i^2\nabla^2\delta\Psi-\xi=0.$$

We note that $v_e/\Delta t d_i^2\sim\Delta t_{CFL}^w/\Delta t\sim h\ll 1$ (using Eq. (16) and that, in Alfvénic units, $v_A=1$), which shows that the relative importance of the hyperresistivity term decreases with grid refinement (and so does the numerical stiffness associated with this term). This coupled system of equations is suitable for coupled MG with coupled Jacobi as a smoother, since all blocks are diagonally dominant (advection terms are upwinded in the preconditioner [3]). The inversion of D_{B_z} is also performed using coupled MG, as follows:

$$\left[\frac{1}{\Delta t}+\theta\vec{v}_{e0}\cdot\nabla-\theta\eta\nabla^2\right]\delta\Psi+\nabla^2\xi=\mathrm{rhs}_\Psi,$$
$$\theta v_e\nabla^2\delta\Psi-\xi=0.$$

Accordingly, we identify this semi-implicit preconditioning alternative as MGSI (multigrid semi-implicit).

The discretization of the blocks is done using second-order finite volumes. The coupled MG routine is implemented matrix-light, and only the diagonal is stored for smoothing purposes in a preconditioner setup stage (the same diagonal is used in every GMRES iteration for a given Newton iteration). The Jacobi underrelaxation parameter is set to 0.75 for optimal smoothing rates, and we employ 10 Jacobi iterations per smoothing call. We employ 2 MG V-cycles per coupled MG call. Piecewise constant restriction is employed, since it is consistent with the conservative formulation of the HMHD system. Second-order spline interpolation is employed for the prolongation step.

Clearly, each MGSI call is more expensive than CGSI. However, contrary to CGSI, MGSI is scalable and should be able to deal with very stiff systems with very refined meshes and/or large values of d_i.

5. Numerical results

The approximations involved in the derivation of the preconditioning strategies above affect the convergence rate of GMRES (and hence the efficiency of the implicit solver), but not the accuracy of the converged solution (which is determined by the discretization chosen: Crank–Nicolson in this instance). Numerical tests of the accuracy and efficiency of the implicit algorithm are performed on a 2D rectangular domain $L_x\times L_y$, discretized uniformly using finite volumes. Boundary conditions are periodic in x and homogeneous Dirichlet in y for all quantities. Physically, Dirichlet boundary conditions imply perfect conductor ($\Psi=0$), impenetrable wall for ions ($\Phi=0$) and for electrons ($B_z=0$), and no stress ($\omega=0$).

Numerical experiments are performed on two different problems, aimed to test different features of the Hall MHD solver. The first problem of choice is the flux-bundle coalescence (FBC) problem [24,25]. Flux bundles are initialized with the following flux function:

$$\Psi_0(x,y)=C\lambda\left\{\exp\left[-\frac{[(x-x_1)^2+(y-y_1)^2]^2}{\lambda^4}\right]+\exp\left[-\frac{[(x-x_2)^2+(y-y_2)^2]^2}{\lambda^4}\right]\right\} \tag{24}$$

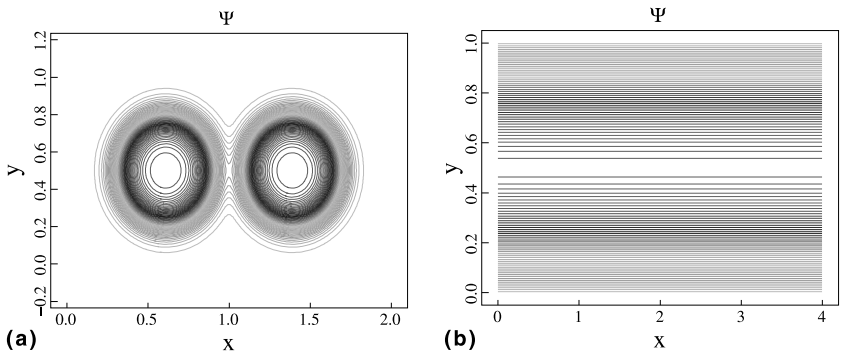

Fig. 1. Initial condition for the magnetic flux Ψ in the (a) FBC problem and (b) KHT problem.

with $x_1 = L_x/2 + 1.3\lambda$, $x_2 = L_x/2 - 1.3\lambda$, $y_1 = y_2 = L_y/2$, and C determined so that the maximum magnetic field in the domain is normalized to unity. For FBC, we fix the following parameters: $L_x = 2$, $L_y = 1$, $\lambda = 0.3$, $\eta = \nu = 10^{-4}$. No perturbation is required to start the simulation, since Eq. (24) is not an equilibrium. Eq. (24) is contoured in Fig. 1(a). This problem tests the HMHD preconditioner in the presence of highly structured magnetic field configurations.

The second problem of interest is the Kelvin–Helmholtz/tearing problem (KHT) [26], defined by the following equilibrium:

$$\Psi_0(x,y) = -\lambda_\Psi \ln\left[\frac{\cosh\{(y-(1/2))/\lambda_\Psi\}}{\cosh(1/2\lambda_\Psi)}\right] + \frac{(y-(1/2))^4 - (1/16)}{3\lambda_\Psi \cosh^2(1/2\lambda_\Psi)}, \tag{25}$$

$$\Phi_0(x,y) = M_A\left\{ -\lambda_\Phi \ln\left[\frac{\cosh\{(y-(1/2))/\lambda_\Phi\}}{\cosh(1/2\lambda_\Phi)}\right] + \frac{(y-(1/2))^4 - (1/16)}{3\lambda_\Phi \cosh^2(1/2\lambda_\Phi)}\right\} \tag{26}$$

and $\omega_0(x,y) = \nabla^2\Phi_0(x,y)$, $B_{z,0} = v_{z,0} = 0$. These initial conditions impose sheared, parallel magnetic and *ion* fluid flow fields. Here, M_A is the Alfvén Mach number, and λ_Ψ and λ_Φ are the gradient scale lengths for the current and vorticity sheets, respectively. The correction to the natural logarithm in Eqs. (25) and (26) is to ensure the equilibrium satisfies homogeneous Dirichlet boundary conditions in y. For KHT, we fix the following parameters: $\lambda_\Psi = \lambda_\Phi = \lambda = 0.2$, $L_x = 4$, $L_y = 1$, $M_A = 1.5$, and $\eta = \nu = 10^{-4}$. KHT is an equilibrium, and the simulation is jump-started with a perturbation in the vorticity, $\delta\omega = 10^{-5}\sin(\pi y)\cos((2\pi/L_x)x)$. Eq. (25) is contoured in Fig. 1(b). This problem tests the performance of the HMHD solver in problems with strong, anisotropic magnetic fields and flows.

Our focus will be on *both* the accuracy and efficiency of the HMHD implicit solver. The following sections deal separately with each of these topics.

5.1. Efficiency

Unless otherwise specified, results of numerical tests in this section are averaged over five implicit time steps. The explicit algorithm employed for comparison is the same as used in [3], but with the explicit CFL limit defined as:

$$\Delta t^{\mathrm{w}}_{\mathrm{CFL}} = 0.9 \min\left[\left(\frac{v_{x,\max}}{\Delta x}+\frac{v_{y,\max}}{\Delta y}+\left[\frac{B_{x,\max}}{\Delta x}+\frac{B_{y,\max}}{\Delta y}\right]\left(1+2\frac{d_{\mathrm{i}}}{h}\right)\right)^{-1}, \left(2\frac{\max(\eta,v)}{h^2}+8\frac{v_{\mathrm{e}}}{h^4}\right)^{-1}\right], \quad (27)$$

where (v_x, v_y) are the velocity components, (B_x, B_y) are the magnetic field components, and $h = (\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2})^{-1/2}$. Explicit calculations are extended over the same time span as the corresponding implicit calculation.

Tables 1 and 2 present results for both preconditioning strategies and both test problems with several grid refinements and $d_{\mathrm{i}} = 0.2$. Implicit time steps Δt are chosen to satisfy the ordering in Eq. (16). Several comments are in order from these tables:

1. Both CGSI and MGSI result in scalable solvers in terms of Newton iterations and GMRES iterations per time step. However, in terms of CPU time, MGSI performs much better than CGSI for the FBC problem, while CGSI outperforms MGSI for the KHT problem. Notice the difference in the number of CG iterations per GMRES iteration (CG/GM) of CGSI between FBC and KHT: CG/GM is an order of magnitude larger in FBC. The reason can be traced to the fact that, in KHT, the magnetic field is aligned with the coarsest grid direction (since $B_x \gg B_y \approx 0$ and $\Delta x \gg \Delta y$ for the geometry of interest), and hence the whistler semi-implicit operator is much less stiff than in FBC, where the magnetic field is not aligned with the grid.

Table 1
Efficiency results for the two preconditioning strategies using the FBC problem and $d_{\mathrm{i}} = 0.2$

Grid	Δt	Newton/Δt	GM/Δt	CG/GM	CPU (s)	$\mathrm{CPU}_{\mathrm{exp}}$/CPU	$\Delta t/\Delta t^{\mathrm{w}}_{\mathrm{CFL}}$
CGSI							
64×64	0.02	4.0	1.2	47	12	3.8	74
128×128	0.01	4.0	3.8	106	100	3.9	147
256×256	0.005	4.0	3.8	232	650	5.4	294
MGSI							
64×64	0.02	3.0	0.8	–	14	3.3	74
128×128	0.01	2.6	0.6	–	46	8.5	147
256×256	0.005	2.0	0	–	123	28.0	294

In this and subsequent tables, Newton/Δt indicates the number of Newton iterations per time step, GM/Δt indicates the number of GMRES iterations per time step, CG/GM indicates the number of unpreconditioned CG iterations per GMRES iteration, CPU is the implicit CPU time, and $\mathrm{CPU}_{\mathrm{exp}}$ is the explicit CPU time.

Table 2
Efficiency results for the two preconditioning strategies using the KHT problem and $d_{\mathrm{i}} = 0.2$

Grid	Δt	Newton/Δt	GM/Δt	CG/GM	CPU (s)	$\mathrm{CPU}_{\mathrm{exp}}$/CPU	$\Delta t/\Delta t^{\mathrm{w}}_{\mathrm{CFL}}$
CGSI							
64×64	0.02	3	0	4	5	4.2	64
128×128	0.01	3	1.2	13	28	6	125
256×256	0.005	2.8	1.4	24	130	10.5	250
MGSI							
64×64	0.02	3	0.2	–	12	1.8	64
128×128	0.01	3.6	2.4	–	85	2.0	125
256×256	0.005	3.6	2.8	–	367	3.7	250

2. Large CPU speedups (CPU_{exp}/CPU) are possible for fine meshes: up to a factor of ~30 is reported for the FBC problem, and ~10 for the KHT problem. The difference in implicit vs. explicit performance between FBC and KHT is due to a larger explicit CFL limit in KHT, again because the magnetic field in KHT is aligned with the coarsest grid direction.
3. CGSI presents a power scaling of CG/GM due to the lack of preconditioning in the CG treatment of the whistler semi-implicit operator. However, the power scaling is relatively weak: in these numerical tests, CG/GM increases by a factor of 2 when refining the mesh by a factor of 2 in each dimension [i.e., it scales as $N^{1/2}$, where N is the total number of mesh points]. The fact that the coefficient of the semi-implicit operator is proportional to Δt^2 (as obtained when multiplying Eqs. (22) and (23) by Δt), and that $\Delta t^2 \sim N^{-1}$ by Eq. (16), implies that the semi-implicit coefficient gets smaller under refinement, and contributes to such a mild scaling of CG/GM with N.
4. In some instances, the *average* number of GMRES iterations per time step ($GM/\Delta t$) is less than unity. This reflects the fact that the preconditioner provides an extremely good initial guess (recall that the preconditioner is also employed to provide a good initial guess for GMRES – see Section 3) and the inexact Newton stage converges without requiring a single GMRES iteration.

Similar conclusions can be extracted from results for $d_i = 0.4$, shown in Tables 3 and 4. In this case, and for the FBC problem, the performance of CGSI degrades with mesh refinement, whereas MGSI performs comparably to the $d_i = 0.2$ case. For the KHI problem, CGSI still outperforms MGSI in raw CPU time, but the scaling of CGSI with grid refinement is showing signs of stalling while the scaling of MGSI is not (i.e., MGSI remains scalable).

The effects of using time steps larger than suggested by the time step ordering in Eq. (16) on the efficiency of the solver are presented in Table 5. The results are obtained for the FBC problem with $d_i = 0.2$ and the

Table 3
Efficiency results for the two preconditioning strategies using the FBC problem and $d_i = 0.4$

Grid	Δt	Newton/Δt	GM/Δt	CG/GM	CPU (s)	CPU_{exp}/CPU	$\Delta t/\Delta t^{w}_{CFL}$
CGSI							
64×64	0.02	5.4	16	90	60	1.44	154
128×128	0.01	5.2	15.4	211	416	3.4	294
256×256	0.005	4.8	20	462	3930	1.6	588
MGSI							
64×64	0.02	4.0	4	–	27	3.1	154
128×128	0.01	3.4	2.2	–	80	17.2	294
256×256	0.005	3.0	1.2	–	248	26.0	588

Table 4
Efficiency results for the two preconditioning strategies using the KHT problem and $d_i = 0.4$

Grid	Δt	Newton/Δt	GM/Δt	CG/GM	CPU (s)	CPU_{exp}/CPU	$\Delta t/\Delta t^{w}_{CFL}$
CGSI							
64×64	0.02	3.0	0.2	14.8	6	5.8	128
128×128	0.01	3.0	1.0	28.4	32	10.5	250
256×256	0.005	2.8	3.0	70.7	250	10.0	500
MGSI							
64×64	0.02	3.2	1.4	–	16	2.1	128
128×128	0.01	4.0	5.6	–	132	2.5	250
256×256	0.005	4.0	3.8	–	447	5.6	500

Table 5
Effect of surpassing the implicit time step limit Δt_A on the efficiency of the solver. The FBC problem with $d_i = 0.2$ and the MGSI preconditioner are chosen for the study

$\Delta t/\Delta t_A$	Newton/Δt	GM/Δt	CPU (s)	CPU_{exp}/CPU	$\Delta t/\Delta t^{\text{w}}_{\text{CFL}}$
128×128					
1	2.6	0.6	46	8.5	147
2	3.6	1.8	78	9.4	294
4	4.8	5.8	147	9.3	588
256×256					
1	2	0	123	28.0	294
2	2.8	0.8	214	30.0	588
4	4.2	3.8	460	26.5	1176

MGSI preconditioner, and show that increasing the time step above the prescribed limits does not immediately degrade the solver efficiency, since the CPU speedup over the explicit solver remains essentially constant. However, given a constant speedup, it is preferable to run at the time step limit Δt_{A} to enhance the accuracy of the calculation.

5.2. Accuracy

The previous efficiency results show implicit time steps several hundred times larger than explicit CFL limits. With such large time steps, accuracy might be an issue. As recently shown [2], the second-order implicit time integration of multiple-time-scale systems (such as this) remains accurate as long as the "dynamical" time scale of the problem τ_{dyn} is respected. The two problems selected previously (FBC, KHT) have well-defined dynamical time scales (stemming from ideal or resistive instabilities), well above any normal mode time scales supported by HMHD, and hence are suitable for implicit integration.

Here, we focus on the FBC problem because of the ideal nature of the driving (coalescence) instability, which results in faster dynamical time scales and thus is more prone to inaccuracies in the implicit solver. We use the following figures of merit for accuracy: (1) the reconnection rate at the magnetic field X-point (local diagnostic), defined as $E = \partial_t \Psi|_X$, and (2) the ℓ_2-norm of the magnetic flux perturbation (global diagnostic), $\ln(\|\delta\Psi\|_2)$. In the geometry of Fig. 1, the X-point is located in the middle of the domain, right where the two bundles overlap, and this position is fixed during the linear and moderately nonlinear phase of the coalescence instability (eventually the two flux bundles merge due to the presence of resistivity and electron viscosity, and the X-point transforms into an O-point). We note that the reconnection rate E is the time derivative of a fundamental quantity Ψ, and as such, it is very sensitive to potential inaccuracies introduced by the integration algorithm.

We test accuracy of the implicit algorithm by comparing the implicit time histories of these figures of merit with their explicit counterparts. Time histories are obtained from a numerical simulation of the FBC problem with $d_i = 0.2$ in a 128×128 grid, running up to a simulation time of $t = 1\tau_{\text{A}}$. The results are presented in Fig. 2, and show that the time histories of both figures of merit (local and global) for the explicit and implicit algorithms agree exceedingly well. In addition, the comparison of contour plots of selected quantities at $t = 0.5\tau_{\text{A}}$ show no appreciable difference between the explicit and implicit results. This is powerful evidence that the implicit code is indeed accurate despite the fact that the implicit time step for this configuration is more than two orders of magnitude larger than the explicit CFL limit (Table 1). This is consistent with the fact that the implicit time step respects the dynamical time scale of the problem, since $\Delta t/\tau_{\text{dyn}} \sim 10^{-2}$.

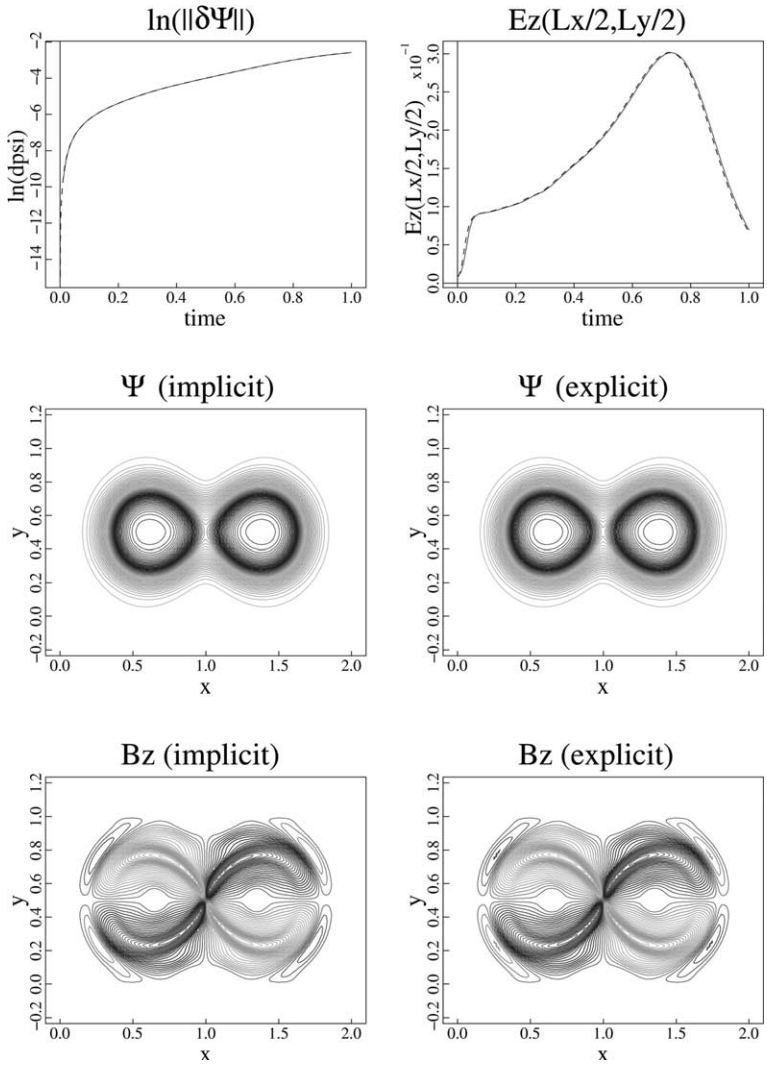

Fig. 2. Comparison between implicit and explicit results to determine loss of accuracy in the implicit algorithm. Simulations are performed using the FBC problem with $d_i = 0.2$ on a 128×128 grid up to a final time of $t = 1\tau_A$. Selected quantities include time histories of local and global diagnostics (see text) and contour plots at $0.5\tau_A$. In the time-history plots, the solid line corresponds to the implicit calculation, and the dashed line to the explicit calculation.

6. Conclusions

An accurate and efficient fully implicit, nonlinear solver strategy for Hall MHD has been presented. The solver is based on Jacobian-free Newton–Krylov methods, employing flexible GMRES [23] as the Krylov solver and converging the nonlinear couplings with an inexact Newton approach. This work expands on the work for resistive MHD in [3].

As is well-known, Krylov solvers require preconditioning for efficiency. A useful preconditioning framework has been developed based on physics insight and a Schur complement approach. The connection of this approach with the physics-based preconditioning concept has been pointed out, as well as its

relationship with previous work [1,3]. As a result, two different preconditioners have been proposed: one based on unpreconditioned CG, and other based on coupled MG.

The efficiency of the solver with these preconditioners is compared using two different model problems: the flux-bundle coalescence problem and the collisionless tearing mode problem with flow. Each model problem tests the solver under very different configurations (structured vs. directed magnetic fields). The results show that each preconditioner excels in different conditions. Generally, CPU speedups over an explicit approach of an order of magnitude are common, and up to a factor of 30 in some instances. Accuracy results indicate that these speedups are obtained without compromising accuracy despite the large time steps employed (which in some instances approach 600 times the explicit CFL condition).

Acknowledgements

The authors acknowledge useful conversations with D.C. Barnes, J.M. Finn, and D. Keyes. This work has been supported by the Los Alamos National Laboratory Directed Research and Development Program. Los Alamos National Laboratory is operated by the US Department of Energy under Contract W-7405-ENG-36.

Appendix A. Functional form of the electron viscosity

Ion viscosity and/or resistivity, in combination with the whistler wave, are not able to define a dissipation length scale, since their corresponding time scales all scale as k^2, with k is the wavenumber [27,28]. A dissipation scale length for this wave at the shortest grid scales requires either electron viscosity or electron inertia. The choice does not fundamentally affect the overall reconnection dynamics in the system. Electron viscosity is chosen here. However, in practice, the coefficient v_e is not known *a priori*, and an *ad-hoc* definition, based on the grid spacing, is necessary.

To derive the correct functional form of v_e, we start by isolating the terms in Eqs. (1)–(5) that are responsible for the propagation of the whistler wave. Linearized about a given magnetic field $\vec{B}_0$, these are:

$$\partial_t \delta\Psi + v_e \nabla^4 \delta\Psi = d_i \vec{B}_0 \cdot \nabla \delta B_z,$$
$$\partial_t \delta B_z + v_e \nabla^4 \delta B_z = -d_i \vec{B}_0 \cdot \nabla \delta J,$$

where the "δ" quantities mean perturbations about some given state, denoted by the subscript "0". We are interested in the balance of the terms in these equations in a neighborhood around a rational surface $\vec{B}_0 \cdot \nabla = B_0 k_\parallel = 0$, where dissipation becomes important. Using standard Fourier analysis, we replace $\partial_t \sim \gamma$ and $\nabla \sim \mathrm{i}\vec{k}$ to find:

$$\gamma \delta\Psi + v_e k^4 \delta\Psi = \mathrm{i} d_i k_\parallel B_0 \delta B_z,$$
$$\gamma \delta B_z + v_e k^4 \delta B_z = \mathrm{i} d_i k^2 k_\parallel B_0 \delta\Psi,$$

Away from the rational surface, the diffusion term is negligible, and we find $\gamma \approx \mathrm{i} d_i B_0 k_\parallel k$, as expected (it is the whistler wave dispersion relation with $v_A = B_0$). At the rational surface, the inertia term γ is balanced by the diffusion term, yielding $\gamma \approx -v_e k^4$. For the diffusion term to be able to damp the whistler wave, the magnitude of these γ should be comparable, yielding:

$$v_e \approx \frac{d_i B_0 k_\parallel}{k^3} \approx \frac{d_i v_A}{k^2}$$

Setting $k \sim 1/h$, where $h = (\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2})^{-1/2}$ and Δx, Δy are the grid spacings in the x and y directions, respectively, we find:

$$v_e = C d_i v_A h^2. \tag{A.1}$$

The coefficient is in practice set to $C = 0.2$ to prevent that the electron viscosity term imposes a more stringent explicit time step limit than the whistler wave.

We note at this point that Eq. (A.1) is of the same functional form as would be obtained by its explicit form, $v_e \sim h^4/\Delta t^w_{CFL}$, with $\Delta t^w_{CFL} \propto h^2/d_i v_A$ the explicit time step. In either case, we note that v_e scales as h^2, not as h^4 as a naive choice of $v_e \sim h^4/\Delta t$ would dictate, where Δt is the implicit time step. Since $h \ll 1$, the latter scaling would result in v_e decreasing too rapidly under grid refinement, and hence in insufficient dissipation and in pollution of the simulation due to numerical noise.

References

[1] D.S. Harned, Z. Mikic, Accurate semi-implicit treatment of the Hall effect in magnetohydrodynamic computations, J. Comput. Phys. 83 (1989) 1–15.
[2] D.A. Knoll, L. Chacón, L. Margolin, V.A. Mousseau, On balanced approximations for the time integration of multiple time scale systems, J. Comput. Phys. 185 (2003) 583–611.
[3] L. Chacón, D.A. Knoll, J.M. Finn, Implicit, nonlinear reduced resistive MHD nonlinear solver, J. Comput. Phys. 178 (1) (2002) 15–36.
[4] V.A. Mousseau, D.A. Knoll, J.M. Reisner, An implicit nonlinearly consistent method for the two-dimensional shallow-water equations with Coriolis force, Mon. Weather Rev. 130 (11) (2002) 2611–2625.
[5] D.A. Knoll, W.B. Vanderheyden, V.A. Mousseau, D.B. Kothe, On preconditioning Newton–Krylov methods in solidifying flow applications, SIAM J. Sci. Comput. 23 (2) (2001) 381–397.
[6] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[7] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[8] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[9] P.R. McHugh, D.A. Knoll, Inexact Newton's method solution to the incompressible Navier–Stokes and energy equations using standard and matrix-free implementations, AIAA J. 32 (12) (1994) 2394–2400.
[10] D.A. Knoll, P.R. McHugh, Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow, SIAM J. Sci. Comput. 19 (1998) 291–301.
[11] R. Dembo, S. Eisenstat, R. Steihaug, Inexact Newton methods, J. Numer. Anal. 19 (1982) 400.
[12] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.
[13] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Chapman & Hall, London, 1989.
[14] L. Chacón, D.C. Barnes, D.A. Knoll, G.H. Miley, An implicit energy-conservative 2D Fokker–Planck algorithm: II-Jacobian-free Newton–Krylov solver, J. Comput. Phys. 157 (2) (2000) 654–682.
[15] D.A. Knoll, G. Lapenta, J.U. Brackbill, A multilevel iterative field solver for implicit, kinetic, plasma simulation, J. Comput. Phys. 149 (1999) 377–388.
[16] W.L. Briggs, A Multigrid Tutorial, SIAM, Philadelphia, 1987.
[17] D.A. Knoll, W.J. Rider, A multigrid preconditioned Newton–Krylov method, SIAM J. Sci. Comput. 21 (2) (1999) 691–710.
[18] E.J. Caramana, Derivation of implicit difference schemes by the method of differential approximation, J. Comput. Phys. 96 (2) (1991) 484–493.
[19] D.S. Harned, W. Kerner, Semi-implicit method for three-dimensional compressible magnetohydrodynamic simulation, J. Comput. Phys. 60 (1985) 62–75.
[20] D.S. Harned, D.D. Schnack, Semi-implicit method for long time scale magnetohydrodynamic computations in three dimensions, J. Comput. Phys. 65 (1986) 57–70.
[21] M. Pernice, Private communication.
[22] P.N. Brown, C.S. Woodward, Preconditioning strategies for fully implicit radiation diffusion with material-energy transfer, SIAM J. Sci. Comput. 23 (2) (2001) 499–516.
[23] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1992) 461–469.
[24] D. Biskamp, E. Schwarz, J.F. Drake, Ion-controlled collisionless magnetic reconnection, Phys. Rev. Lett. 75 (21) (1995) 3850–3853.

[25] D. Biskamp, E. Schwarz, J.F. Drake, Two-fluid theory of collisionless magnetic reconnection, Phys. Plasmas 4 (4) (1997) 1002–1009.
[26] L. Chacón, D.A. Knoll, J.M. Finn, Hall MHD effects in the 2-D Kelvin–Helmholtz/tearing instability, Phys. Lett. A 308 (2–3) (2003) 187–197.
[27] J. Birn, J.F. Drake, M.A. Shay, B.N. Rogers, R.E. Denton, M. Hesse, M.M. Kuznetsova, Z.W. Ma, A. Bhattacharjee, A. Otto, P.L. Pritchett, Geospace Environment Modeling (GEM) magnetic reconnection challenge, J. Geophys. Res. 106 (A3) (2001) 3715–3719.
[28] J. Birn, M. Hesse, Geospace Environment Modeling (GEM) magnetic reconnection challenge: Resistive tearing, anisotropic pressure and Hall effects, J. Geophys. Res. 106 (A3) (2001) 3737–3750.