Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Physics Communications 172 (2005) 119–132
Computer Physics Communications
www.elsevier.com/locate/cpc

A numerical instability in an ADI algorithm for gyrokinetics

E.A. Belli *, G.W. Hammett

Princeton Plasma Physics Laboratory, Princeton, NJ 08543, USA

Received 14 December 2004; received in revised form 25 May 2005; accepted 26 June 2005

Available online 8 August 2005

Abstract

We explore the implementation of an Alternating Direction Implicit (ADI) algorithm for a gyrokinetic plasma problem and its resulting numerical stability properties. This algorithm, which uses a standard ADI scheme to divide the field solve from the particle distribution function advance, has previously been found to work well for certain plasma kinetic problems involving 1 spatial and 2 velocity dimensions, including collisions and an electric field. However, for the gyrokinetic problem we find a severe stability restriction on the time step. Furthermore, we find that this numerical instability limitation also affects some other algorithms, such as a partially implicit Adams–Bashforth algorithm, where the parallel motion operator $v_\parallel \partial/\partial z$ is treated implicitly and the field terms are treated with an Adams–Bashforth explicit scheme. Fully explicit algorithms applied to all terms can be better at long wavelengths than these ADI or partially implicit algorithms.

PACS: 52.65.Tt; 52.65.-y; 52.35.Ra

Keywords: Gyrokinetics; ADI; Eulerian

1. Introduction

Edge plasmas are known to play a critical role in tokamak confinement. A complete model of fusion edge plasma turbulence requires a full gyrokinetic description for all ions and electrons to accurately capture the large range of spatial scales due to the high degree of variation in the collisionality across the edge region. While computations of the electrostatic gyrokinetic equation with adiabatic electrons can be performed with straightforward numerical schemes, the inclusion of kinetic electrons and electromagnetic effects has been numerically challenging due to the smaller length scales and faster time scales associated with the fast parallel electron dynamics relative to the modes of interest. Furthermore, for edge turbulence codes, the existence of an Alfvén wave in the low β edge/scrape-off region, where the wave is even faster than the thermal electron motion, causes most standard explicit algorithms to

* Corresponding author.
E-mail address: ebelli@pppl.gov (E.A. Belli).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2005.06.007

need very small time steps for numerical stability. A semi-implicit or fast implicit algorithm that could use larger time steps, without excessive computational overhead, would thus be advantageous.

Various hybrid methods have been studied for extending Lagrangian particle-in-cell (PIC) gyrokinetic codes to include kinetic electrons and fully electromagnetic dynamics (see [1–3] and references therein). Recent breakthroughs in the PIC approach to electromagnetic gyrokinetics have been achieved via careful treatment of cancellations that should occur in the magnetic potential field equation [4]. In this paper we focus on algorithms useful for Eulerian codes (where these cancellations are straightforward to ensure and are in fact automatic in some formulations). Eulerian codes are being intensively used for nonlinear electromagnetic gyrokinetic simulations.

Eulerian/continuum codes use finite difference and/or spectral methods on a discrete grid. While there are interesting issues involved in various choices of spatial discretization of the gyrokinetic equation [5–7], here we will focus on the time-advancement algorithm and will just Fourier transform in the spatial directions. Many Eulerian codes use explicit or semi-implicit time-stepping algorithms. For example, the GENE code, which has been useful for studying various regimes of drift-Alfvén and ETG turbulence [8], uses an explicit Lax–Wendroff finite difference technique for the linear terms with a multi-dimensional, second-order upwind method for the nonlinear terms [9]. Other explicit algorithms for the linear terms in GENE have recently been studied [5]. The initial algorithm of the semi-global GYRO code [6] was a fully explicit, 5 stage, fourth-order Runge–Kutta scheme. However, this algorithm was found to be numerically unstable at small $k_\perp \rho_i$ even at time steps well below the electron advective Courant limit ($v_\parallel \Delta t/\Delta_L \ll 1$) since it was mathematically connected with the electrostatic Alfvén branch. Recent implementation of a second-order Implicit–Explicit Runge–Kutta splitting scheme in GYRO [10] using precomputed plasma response matrices for the parallel dynamics (a variant of GS2's approach) has yielded improved stability by naturally cutting off high frequency oscillations, while still asymptotically preserving accuracy in the stiff limit, unlike some higher-order splitting schemes. The Eulerian flux tube-based GS2 code [7,11] was the first implementation of the fully electromagnetic, nonlinear 5D gyrokinetic equations including trapped and passing particle dynamics. It employs a fully implicit treatment of all of the linear terms (parallel dynamics, ω_* diamagnetic terms, and magnetic drifts), and thus has no time restrictions on stability in the linear limit. Of course there is a stability limit from the explicit treatment of the nonlinear terms, but the implicit treatment of the linear terms is still a significant advantage because they contain high frequency waves that do not interact much with the turbulence of interest but still need to be treated in a numerically stable way. While these various Eulerian gyrokinetic codes have been quite successful, there is some overhead in the precomputation of the plasma response implicit matrices, so there is interest in exploring faster semi-implicit algorithms.

Recently, an Alternating Direction Implicit (ADI) algorithm developed by Kupfer et al. [12] has been considered for kinetic edge microturbulence simulations. This two-step scheme splits the treatment of the parallel advection terms from the treatment of the electric field acceleration terms, treating them implicitly on alternating steps. This method has the advantage of avoiding the set up of large plasma response matrices needed for an unsplit implicit treatment of the linear gyrokinetic terms. Kupfer successfully used this ADI algorithm for a kinetic equation for electrons with 1 spatial dimension (in the parallel direction) and 2 velocity dimensions, including collisions with fixed Maxwellian background ions and imposing a quasineutrality constraint. While this model is useful for understanding aspects of scrape-off layer plasmas, the equations used did not contain the Alfvén wave dynamics of the full gyrokinetic equation, which would be needed for a complete 3D simulation of edge plasmas.

ADI algorithms are often useful in solving PDEs where an operator that is difficult to invert can be split into two operators that are much simpler to invert. ADI schemes often have the property that they are absolutely stable for arbitrarily large time step, which makes them relatively robust (though of course there are accuracy limits on the time step). For a general problem of the form $\partial y/\partial t = (L_1 + L_2)y$, where L_1 and L_2 are arbitrary split linear operators, stability of an ADI scheme is guaranteed if L_1 and L_2 are diagonalizable and if all of the eigenvalues of L_1 and L_2 satisfy $\Re(\lambda_{1,j}) \leqslant 0$ and $\Re(\lambda_{2,j}) \leqslant 0$ [13]. Thus, in practice, application of ADI schemes to many physical problems has been successful. For example, an ADI scheme applied to a parabolic problem with two spatial variables (such as the 2D heat conduction equation) can easily be shown to be absolutely stable [14,15]. Furthermore, stabilizing correction schemes, also a class of implicit schemes of alternating direction, can be used to achieve ab-

solute stability for purely parabolic problems of higher dimensionality [15] as well as for some advection–diffusion and advection–reaction equations [13].

In this paper we use a simple kinetic Alfvén wave limit of the gyrokinetic equation to test an ADI algorithm. We find the somewhat surprising result that not only does the ADI algorithm for this equation have a stability limit on the time step, but also that stability limit can be quite short for long wavelength modes, even worse than some fully explicit algorithms. This problem exists even at higher β where the Alfvén wave is slower than the electrons. Thus one needs to look at other options for faster gyrokinetic algorithms.

2. Kinetic Alfvén wave test problem

As a starting point, we consider the test problem of a shear kinetic Alfvén wave at small $k_\perp \rho_i$. Thus, for the simplified starting equations, we consider the gyrokinetic equation [16–18] in the linear, collisionless limit in slab geometry with a uniform magnetic field and uniform background Maxwellian particles. For further simplicity, we also neglect the kinetic equation for ion perturbations, assuming $\omega \gg k_\parallel v_{ti}$. Thus, the only ion contribution to perturbations will be through the ion polarization density. With these assumptions, the kinetic and field equations become:

$$\frac{\partial f_e}{\partial t} + v_\parallel \frac{\partial f_e}{\partial z} = -\frac{Z_e e}{T_{0e}} v_\parallel F_{Me} \left(\frac{\partial \Phi}{\partial z} + \frac{1}{c} \frac{\partial A_\parallel}{\partial t} \right), \tag{1}$$

$$\frac{(Z_i e)^2 n_{0i}}{T_{0i}} k_\perp^2 \rho_i^2 \Phi = Z_e e \int \mathrm{d}^3 v \, f_e, \tag{2}$$

$$k_\perp^2 A_\parallel = \frac{4\pi}{c} Z_e e \int \mathrm{d}^3 v \, v_\parallel f_e. \tag{3}$$

Here, $f_e(z, \vec{v}, t)$ is the electron distribution function, z and $v_\parallel$ are the position and velocity along the magnetic field, Φ is the electrostatic potential, $A_\parallel$ is the parallel component of the perturbed magnetic vector potential, ρ_i is the thermal ion gyroradius, T_{0i} and T_{0e} are the ion and electron temperatures, and F_{Me} is a Maxwellian distribution for the background electrons. For simplicity in this analysis, we also assume that $Z_i = 1$ ($Z_e = -1$). These or very similar equations have been used previously to study kinetic Alfvén waves and various numerical methods [2,5,19–21].

Using a Fourier transform in time and space, i.e. $f_e = \hat{f}_e e^{-i\omega t + i k_\parallel z}$, we find that

$$\hat{f}_e = \frac{Z_e e}{T_{0e}} F_{Me} \frac{k_\parallel v_\parallel (\Phi - \frac{\omega}{k_\parallel c} A_\parallel)}{\omega - k_\parallel v_\parallel}. \tag{4}$$

Using this result in the field equations and expanding to lowest non-trivial order in the limit of $\omega \gg k_\parallel v_{te}$, we obtain the dispersion relation

$$\omega^2 = \frac{k_\parallel^2 v_A^2}{1 + \frac{2}{\beta_e} \frac{m_e}{m_i} k_\perp^2 \rho_s^2}, \tag{5}$$

where $\beta_e = 8\pi n_0 T_{0e}/B^2$, $\rho_s^2 = c_s^2/\Omega_{ci}$ is the sound-based gyroradius, $c_s^2 = T_{0e}/m_i$ is the sound speed, and $v_A^2 = (2/\beta_e) c_s^2$ is the Alfvén speed. Note that this is just the dispersion relation for a simple shear Alfvén wave in a straight magnetic field, with some finite gyroradius corrections. (A gyrofluid version of this derivation can be found in [19].) If $\beta_e < 2 m_e/m_i$, such as in regions of very low density edge and scrape-off plasmas, then the Alfvén wave is faster than the thermal electron speed. A stable treatment of this wave in this regime is important. However, consideration of these equations in the pure electrostatic limit (i.e. $\beta_e \to 0$) yields the high frequency electrostatic shear Alfvén wave [21], i.e. $\omega^2 = k_\parallel^2 v_{te}^2/(k_\perp^2 \rho_s^2)$. Thus, to avoid this excessively high frequency mode at low $k_\perp$, it is useful to include magnetic perturbations from $A_\parallel$, as we do here.

Following the approach used in most existing continuum gyrokinetic codes (such as the GENE code [8] and the GYRO code [6]), we eliminate the $\partial A_\parallel/\partial t$ term from the electron kinetic equation by defining $g_e = f_e + \frac{Z_e e}{T_{0e}} F_{Me} \frac{v_\parallel}{c} A_\parallel$. (Other codes, such as GS2 [7], which uses a linearly fully implicit algorithm, retain the $\partial A_\parallel/\partial t$ term, though both formulations are equivalent.) With this substitution, the equations for our kinetic Alfvén wave test problem become:

$$\frac{\partial g_e}{\partial t} + v_\parallel \frac{\partial g_e}{\partial z} = -\frac{Z_e e}{T_{0e}} v_\parallel F_{Me} \left(\frac{\partial \Phi}{\partial z} - \frac{v_\parallel}{c} \frac{\partial A_\parallel}{\partial z} \right), \tag{6}$$

$$\frac{(Z_i e)^2 n_{0i}}{T_{0i}} k_\perp^2 \rho_i^2 \Phi = Z_e e \int \mathrm{d}^3 v \, g_e, \tag{7}$$

$$\left(k_\perp^2 + \frac{4\pi}{c^2} \frac{(Z_e e)^2 n_{0e}}{m_e} \right) A_\parallel = \frac{4\pi}{c} Z_e e \int \mathrm{d}^3 v \, v_\parallel g_e. \tag{8}$$

The equivalent standard normalized equations are given in Appendix A.

3. Implementing and testing an ADI algorithm

Kupfer's ADI algorithm as applied to the kinetic equation is a two-step algorithm: the first step is implicit in the parallel advection term and explicit in the field terms, while the second step is explicit in the parallel advection and implicit in the fields. (This is equivalent to a standard view of ADI as an "alternating direction" approach, since the electric field term represents motion in the velocity direction of $(z, v_\parallel)$ phase space.) Thus, the discrete equations are:

$$\frac{1}{\Delta t/2} \left(g_e^{n+1/2} - g_e^n \right) + v_\parallel \frac{\partial g_e^{n+1/2}}{\partial z} = -\frac{Z_e e}{T_{0e}} v_\parallel F_{Me} \left(\frac{\partial \Phi^n}{\partial z} - \frac{v_\parallel}{c} \frac{\partial A_\parallel^n}{\partial z} \right), \tag{9}$$

$$\frac{1}{\Delta t/2} \left(g_e^{n+1} - g_e^{n+1/2} \right) + v_\parallel \frac{\partial g_e^{n+1/2}}{\partial z} = -\frac{Z_e e}{T_{0e}} v_\parallel F_{Me} \left(\frac{\partial \Phi^{n+1}}{\partial z} - \frac{v_\parallel}{c} \frac{\partial A_\parallel^{n+1}}{\partial z} \right), \tag{10}$$

$$\frac{(Z_i e)^2 n_{0i}}{T_{0i}} k_\perp^2 \rho_i^2 \Phi^{n+1} = Z_e e \int \mathrm{d}^3 v \, g_e^{n+1}, \tag{11}$$

$$\left(k_\perp^2 + \frac{4\pi}{c^2} \frac{(Z_e e)^2 n_{0e}}{m_e} \right) A_\parallel^{n+1} = \frac{4\pi}{c} Z_e e \int \mathrm{d}^3 v \, v_\parallel g_e^{n+1}. \tag{12}$$

Using the ansatz $e^{ik_\parallel z}$ for the perturbed quantities, we combine Eqs. (9) and (10) to express the time advanced distribution function g_e^{n+1} in terms of the fields:

$$\begin{aligned} g_e^{n+1} = {} & \frac{1 - ik_\parallel v_\parallel \Delta t/2}{1 + ik_\parallel v_\parallel \Delta t/2} \left[g_e^n - \frac{Z_e e}{T_{0e}} (ik_\parallel v_\parallel \Delta t/2) F_{Me} \left(\Phi^n - \frac{v_\parallel}{c} A_\parallel^n \right) \right] \\ & - \frac{Z_e e}{T_{0e}} (ik_\parallel v_\parallel \Delta t/2) F_{Me} \left(\Phi^{n+1} - \frac{v_\parallel}{c} A_\parallel^{n+1} \right). \end{aligned} \tag{13}$$

Defining a complex amplification factor per time step as $a \equiv e^{-i\omega \Delta t}$ and using the further ansatz for all fields that $g_e^n = a^n \hat{g}_e(v)$, we find that

$$\hat{g}_e = \frac{Z_e e}{T_{0e}} F_{Me} \left(\hat{\Phi} - \frac{v_\parallel}{c} \hat{A}_\parallel \right) \frac{k_\parallel v_\parallel}{\hat{\omega} - k_\parallel v_\parallel} \left(1 + k_\parallel v_\parallel \hat{\omega} (\Delta t/2)^2 \right), \tag{14}$$

where $\hat{\omega}$ is an effective frequency defined such that $\frac{a-1}{a+1} \equiv -i\hat{\omega}\Delta t/2$, or

$$a = \frac{1 - i\hat{\omega}\Delta t/2}{1 + i\hat{\omega}\Delta t/2}. \tag{15}$$

For real $\hat{\omega}$, $|a| = 1$ and perfect stability with no artificial damping is obtained, even for arbitrarily large time step. Absolute stability $|a| \leqslant 1$ also occurs for all modes with $\mathrm{Im}(\hat{\omega}) \leqslant 0$. However, we will find that if the time step is too large, then the numerical dispersion relation for the ADI algorithm has roots with $\mathrm{Im}(\hat{\omega}) > 0$, which correspond to $|a| > 1$ and thus a numerical instability.

Using the result of Eq. (14) in the field equations and again expanding to lowest non-trivial order in $k_\parallel v_{te}/\hat{\omega}$, we find that

$$\hat{\omega}^2 = \left[\frac{k_\parallel^2 v_A^2 (k_\perp \rho_s)^2 (1 + 3(k_\parallel v_{te}\Delta t/2)^2)}{(\frac{2}{\beta_e}\frac{m_e}{m_i} k_\perp^2 \rho_s^2 + 1 + 3(k_\parallel v_{te}\Delta t/2)^2)((k_\perp \rho_s)^2 - (k_\parallel v_{te}\Delta t/2)^2)} \right]. \tag{16}$$

Note that this discrete version of the dispersion relation agrees with the analytic result in Eq. (5) in the limit $\Delta t \to 0$. In the electrostatic limit ($\beta_e \to 0$), the dispersion relation becomes

$$\hat{\omega}^2 = (k_\parallel v_{te})^2 \left[\frac{1 + 3(k_\parallel v_{te}\Delta t/2)^2}{(k_\perp \rho_s)^2 - (k_\parallel v_{te}\Delta t/2)^2} \right]. \tag{17}$$

Thus, for both the electrostatic limit and the general electromagnetic case of Eq. (16), the algorithm is numerically unstable if $\Delta t/2 > |\frac{k_\perp \rho_s}{k_\parallel v_{te}}|$. Figs. 1 and 2 show plots of $\hat{\omega}$ and $|a|$ as a function of the temporal resolution $k_\parallel v_{te}\Delta t$ for the right-moving wave for a set of standard parameters, showing the onset of the numerical instability at $|k_\parallel v_{te}\Delta t| \sim |2k_\perp \rho_s| \sim 0.06$. This result implies that, with this algorithm, the electrostatic shear Alfvén wave must be fully resolved for stability in both the electrostatic and electromagnetic limits. Overall, this can be a potentially severe limitation for numerical simulations employing this type of ADI algorithm.

While the results thus far have focused on the limit of low $(\beta_e/2)(m_i/m_e)$, where the Alfvén wave is faster than the electron thermal speed and expansions in $k_\parallel v_{te}/\omega \ll 1$ could be done, we have also analyzed the numerical

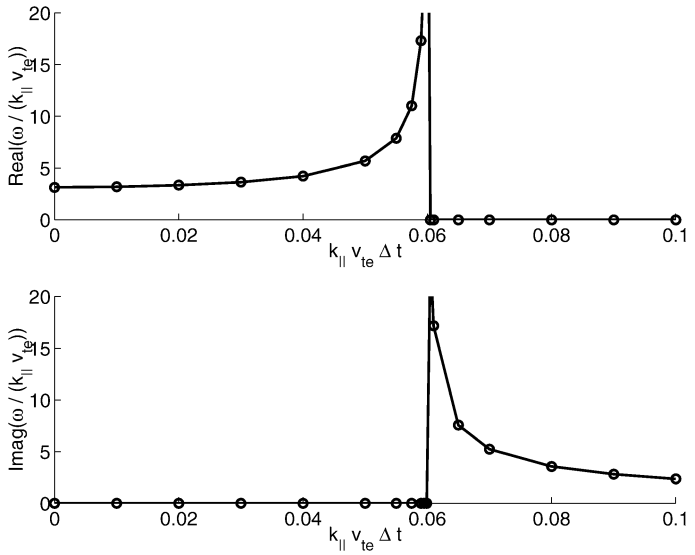

Fig. 1. Normalized mode frequency vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ and $(\beta_e/2)(m_i/m_e) = 0.1$ using an ADI algorithm.

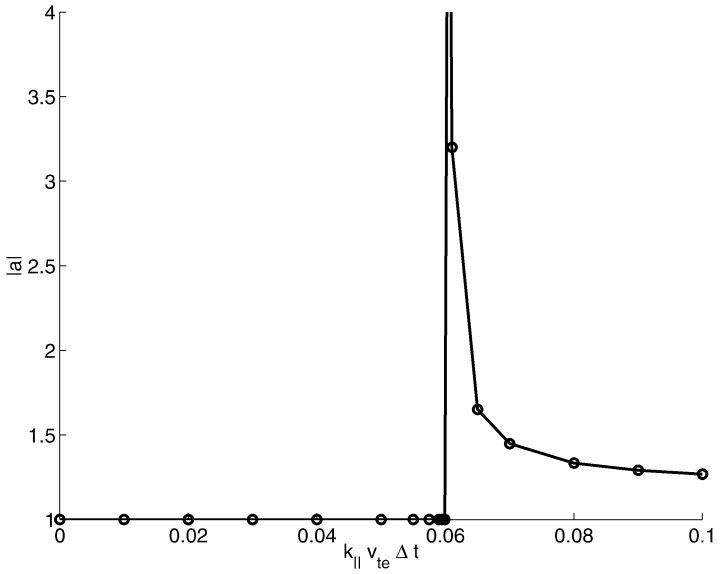

Fig. 2. Amplification factor vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ and $(\beta_e/2)(m_i/m_e) = 0.1$ using an ADI algorithm.

stability of the ADI algorithm more generally, including the regime of high $(\beta_e/2)(m_i/m_e)$, where the Alfvén wave is slower than the thermal electron speed. This analysis was performed via consideration of an nth-order generalized Lorentzian approximation (i.e. $e^{-x} \approx (1 + \frac{x}{n})^{-n}$) for the parallel component of the Maxwellian in Eq. (14). It can be shown that $n \geqslant 3$ is necessary for convergence of the velocity integrals over the Maxwellian terms in the field equations. For consistency in the transformation of the field equations upon elimination of the $\partial A_\parallel/\partial t$ term, we have added normalization constants to the Lorentzian approximation to ensure that the density and pressure integrals are exact. Specifically, we assume that

$$F_{Me}(v_\parallel) \approx \frac{n_{0e}}{\sqrt{2\pi}\, v_{te}} \frac{C_0}{(1 + C_1 \frac{v_\parallel^2}{2v_{te}^2})^n}, \tag{18}$$

where

$$C_0 = \frac{\Gamma(n)\Gamma^{1/2}(n - 3/2)}{\Gamma^{3/2}(n - 1/2)}, \tag{19}$$

$$C_1 = \frac{\Gamma(n - 3/2)}{\Gamma(n - 1/2)}. \tag{20}$$

For this analysis, a third-order Lorentzian approximation was used and the field equations using Eq. (14) with Eq. (18) as an approximation for $\hat{g}_e$ were solved numerically using Maple [22] for given values of $k_\perp \rho_s$ and $(\beta_e/2)(m_i/m_e)$. (If an exact Maxwellian is used, the integrals cannot be evaluated analytically but can be written in terms of the plasma dispersion Z function. With a Lorentzian $F_{Me}(v_\parallel)$, Maple is able to do the integrals analytically, resulting in essentially a multipole approximation to the Z function. Alternatively, one can interpret the resulting dispersion relation as exact for an equilibrium distribution function given by this generalized Lorentzian, and so it is a physically realizable exact dispersion relation. Since Alfvén waves should be physically stable even with this Lorentzian equilibrium according to the Penrose stability criterion, this provides a useful test of the stability of numerical algorithms.)

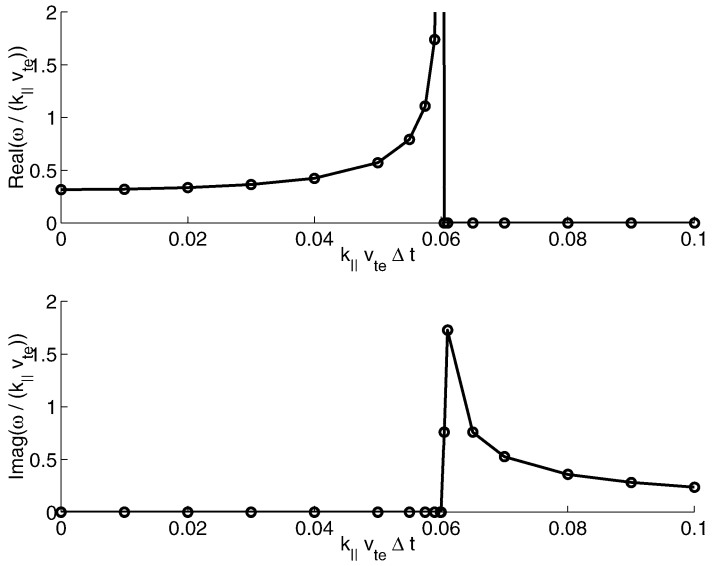

Fig. 3. Normalized mode frequency vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ and $(\beta_e/2)(m_i/m_e) = 10$ using an ADI algorithm.

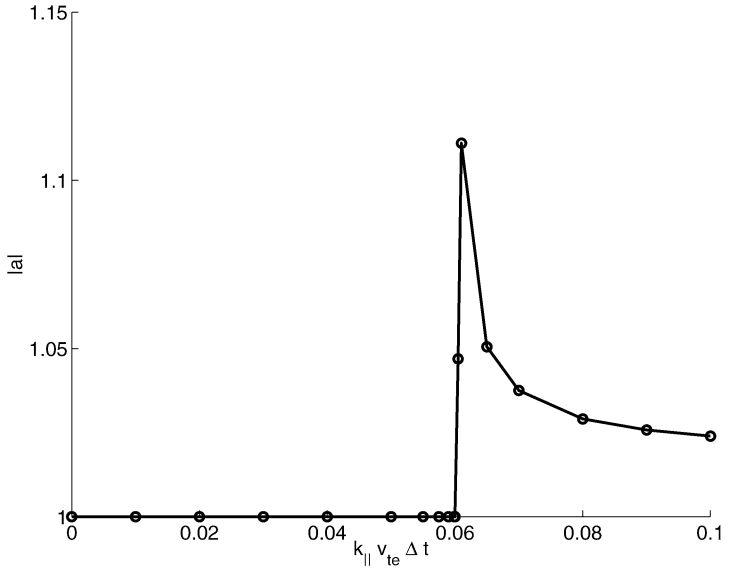

Fig. 4. Amplification factor vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ and $(\beta_e/2)(m_i/m_e) = 10$ using an ADI algorithm.

The result, demonstrated in Figs. 3 and 4, surprisingly also shows a severe stability limit on the ADI algorithm of $|k_\parallel v_{te} \Delta t| \sim |2k_\perp \rho_s| \sim 0.06$, i.e. the same stability criterion as found in the $\beta_e = 0$ limit in Eq. (17), even though the Alfvén wave is slower than the electron thermal velocity at high $(\beta_e/2)(m_i/m_e)$. A time step of $k_\parallel v_{te} \Delta t = 0.06$ corresponds to $\omega \Delta t = 0.02$, so both the parallel electron motion time scale and the actual mode frequency would appear to be very well resolved, yet still there is a numerical instability.

4. Simpler illustration of the difficulties

Here we further illustrate the numerical difficulties of an ADI algorithm by an equivalent application of the algorithm to a Landau-fluid approximation to the kinetic equation. This reduces the operators involved to 2×2 matrices. This limit is useful for understanding why the ADI algorithm in this case has a stability limit at all, unlike other applications where an ADI algorithm is absolutely stable.

Integrating the normalized Eqs. (A.1)–(A.3) over velocity and defining the perturbed density $\rho = \int \mathrm{d}v_\parallel\, g_e$, the perturbed fluid velocity $u = \int \mathrm{d}v_\parallel\, g_e v_\parallel$, and the perturbed pressure $p = \int \mathrm{d}v_\parallel\, g_e v_\parallel^2$ leads to the fluid equations:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial u}{\partial z} - \frac{\partial A_\parallel}{\partial z}, \tag{21}$$

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial z} + \frac{\partial \Phi}{\partial z}, \tag{22}$$

$$k_\perp^2 \Phi = -\rho, \tag{23}$$

$$\left(k_\perp^2 + \hat{\beta}\right) A_\parallel = -\hat{\beta} u. \tag{24}$$

Assuming a general closure approximation of the form $-\partial p/\partial z = -\Gamma \partial \rho/\partial z - \nu |k_\parallel| u$ (see [23] and references therein for a discussion of closure approximations that model kinetic effects such as Landau-damping) and Fourier transforming in space leads to

$$\frac{\partial \rho}{\partial t} = -ik_\parallel u - ik_\parallel A_\parallel, \tag{25}$$

$$\frac{\partial u}{\partial t} = -ik_\parallel \Gamma \rho - \nu |k_\parallel| u + ik_\parallel \Phi \tag{26}$$

which can be written as

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ u \end{pmatrix} = \begin{pmatrix} 0 & -ik_\parallel \\ -i\Gamma k_\parallel & -\nu |k_\parallel| \end{pmatrix}\begin{pmatrix} \rho \\ u \end{pmatrix} + \begin{pmatrix} 0 & \frac{ik_\parallel \hat{\beta}}{k_\perp^2 + \hat{\beta}} \\ -\frac{ik_\parallel}{k_\perp^2} & 0 \end{pmatrix}\begin{pmatrix} \rho \\ u \end{pmatrix}. \tag{27}$$

Denoting the first matrix on the RHS by $\mathbf{P}$, which represents the spatial propagation operator, and the second matrix on the RHS by $\mathbf{E}$, which represents the electric field term, and denoting the state vector $\vec{y} = (\rho, u)$, this can be written as

$$\frac{\partial \vec{y}}{\partial t} = \mathbf{P}\vec{y} + \mathbf{E}\vec{y}. \tag{28}$$

Applying the ADI algorithm to this in an equivalent way as used in Eqs. (9)–(10), where the electric field is explicit on the first half step while all other terms are implicit and then vice versa on the next step, leads to

$$\frac{\vec{y}^{n+1/2} - \vec{y}^n}{\Delta t/2} = \mathbf{P}\vec{y}^{n+1/2} + \mathbf{E}\vec{y}^n, \tag{29}$$

$$\frac{\vec{y}^{n+1} - \vec{y}^{n+1/2}}{\Delta t/2} = \mathbf{P}\vec{y}^{n+1/2} + \mathbf{E}\vec{y}^{n+1}. \tag{30}$$

Combining these two steps of the ADI algorithm gives

$$\vec{y}^{n+1} = \left(1 - \frac{\Delta t}{2}\mathbf{E}\right)^{-1}\left(1 + \frac{\Delta t}{2}\mathbf{P}\right)\left(1 - \frac{\Delta t}{2}\mathbf{P}\right)^{-1}\left(1 + \frac{\Delta t}{2}\mathbf{E}\right)\vec{y}^n. \tag{31}$$

In common ADI cases where the operators being split are diagonalizable and have eigenvalues with zero or negative real part, the ADI algorithm is absolutely stable for arbitrarily large time step (though of course there are

accuracy restrictions), because Δt appears in the numerators of the RHS of this equation. Consider our case first in the electrostatic $\hat{\beta} = 0$ limit. Then the operator corresponding to the electric field $\mathbf{E}$ is not diagonalizable because its only non-zero entry is off-diagonal. This operator has the property that $\mathbf{E}^n = 0$ for all $n > 1$ (i.e. $\mathbf{E}$ is a nilpotent matrix). This means that $(1 - \frac{\Delta t}{2}\mathbf{E})^{-1} = 1 + \frac{\Delta t}{2}\mathbf{E}$ and what appeared to be an implicit step was actually equivalent to an explicit step. Thus the ADI algorithm for this case will be unstable if the time step Δt is too big. (In fact, analysis of the eigenvalues of the amplification matrix corresponding to Eq. (31) for this case recovers the onset of a numerical instability at $|k_\parallel \Delta t| \sim |2k_\perp|$, as found for the kinetic equations.) For the more general electromagnetic case, though $\mathbf{E}$ is diagonalizable for nonzero $\hat{\beta}$, the eigenvalues λ of $\mathbf{E}$ are given by $\lambda^2 = (k_\parallel/k_\perp)^2\hat{\beta}/(k_\perp^2 + \hat{\beta})$, and the positive branch gives an instability. This is in contrast to the behavior of the unsplit operator $\mathbf{P} + \mathbf{E}$, which has negative values of λ^2 (in the $\nu = 0$ limit for simplicity), which correspond to stable oscillations.

5. Comparison with Adams–Bashforth algorithms

For comparison, we perform a similar analysis of the kinetic Alfvén wave test problem using an Adams–Bashforth algorithm. We first consider a partially implicit scheme, in which the parallel derivative term is treated implicitly and time-centered while the field terms are treated fully explicitly with a second-order Adams–Bashforth algorithm. Thus, the single-step discrete kinetic equation becomes

$$\frac{1}{\Delta t/2}\left(g_e^{n+1} - g_e^n\right) + v_\parallel \frac{\partial}{\partial z}\frac{1}{2}\left(g_e^{n+1} + g_e^n\right) = -\frac{Z_e e}{T_{0e}} v_\parallel F_{Me} \frac{\partial}{\partial z}\left(\frac{1}{2}(3\Phi^n - \Phi^{n-1}) - \frac{v_\parallel}{c}\frac{1}{2}\left(3A_\parallel^n - A_\parallel^{n-1}\right)\right). \tag{32}$$

Again using the ansatz $e^{ik_\parallel z}$ and defining $g_e^n = a^n \hat{g}_e(v)$, we find that

$$\hat{g}_e = \frac{Z_e e}{T_{0e}} F_{Me}\left(\hat{\Phi} - \frac{v_\parallel}{c}\hat{A}_\parallel\right)\frac{k_\parallel v_\parallel}{\hat{\omega} - k_\parallel v_\parallel}(1 - i\hat{\omega}\Delta t)\frac{(1 + i\hat{\omega}\Delta t/2)}{(1 - i\hat{\omega}\Delta t/2)}, \tag{33}$$

where $\hat{\omega}$ is again defined in agreement with Eq. (15).

We again examine the stability of the algorithm in both the low and high $(\beta_e/2)(m_i/m_e)$ limits. For the low $(\beta_e/2)(m_i/m_e)$ analysis, substitution of Eq. (33) into the field equations and expansion to lowest order in $k_\parallel v_{te}/\hat{\omega}$ yields a fourth-order complex equation for $\hat{\omega}$, which we solve numerically with Maple using our standard parameters. Analysis in the high $(\beta_e/2)(m_i/m_e)$ limit is likewise performed as before, using a third-order Lorentzian approximation for the parallel component of the Maxwellian term in (33) and using Maple to numerically solve the field equations with this approximation.

Figs. 5 and 6 show the results of the analysis for the kinetic Alfvén wave. Though there is a slight artificial decay for both the low and high $(\beta_e/2)(m_i/m_e)$ cases, the numerical instability that we saw for the ADI scheme does not occur in either regime for these roots. However, the discrete dispersion relation contains multiple roots and these plots are for the eigenmode corresponding to the physical Alfvén wave only. Furthermore, an Adams–Bashforth algorithm introduces unphysical "computational modes" which must be also damped or there will still be a numerical instability. For both the low and high $(\beta_e/2)(m_i/m_e)$ cases, the physical modes found in the analysis are numerically stable over the range of Δt studied. These include the right and left moving kinetic Alfvén waves and a heavily damped entropy mode related to Landau damping (there are 3 physical roots of the analytic dispersion relation for a third-order Lorentzian equilibrium). However, one of the computational mode becomes numerically unstable. The amplification factor as a function of normalized time step for this mode is shown in Fig. 7 for $k_\perp \rho_s = 0.01$, 0.03 (our standard case), and 0.05, for both $(\beta_e/2)(m_i/m_e) = 0.1$ and $(\beta_e/2)(m_i/m_e) = 10$. The onset of the numerical instability occurs at $|k_\parallel v_{te}\Delta t| \sim |k_\perp \rho_s|$, as indicated by the rapid rise in the modulus of the amplitude above 1. Thus, the partially implicit Adams–Bashforth algorithm is subject to a stability limit which is

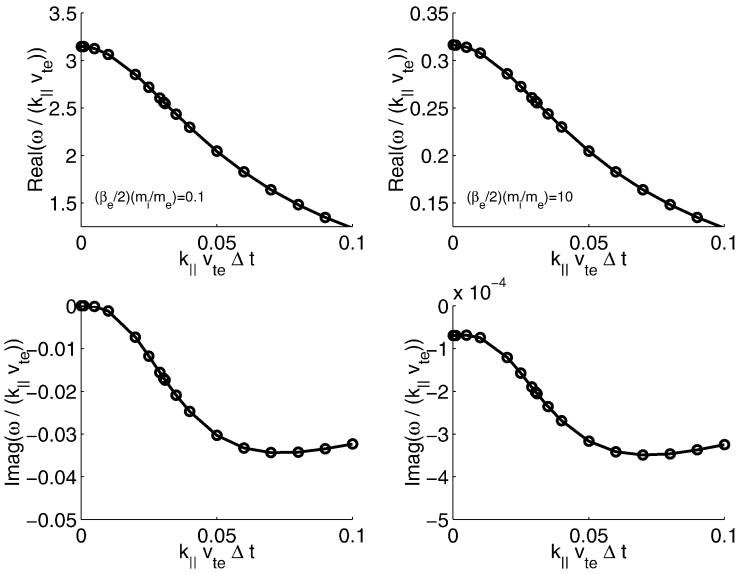

Fig. 5. Normalized mode frequency vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ using a partially implicit Adams–Bashforth algorithm.

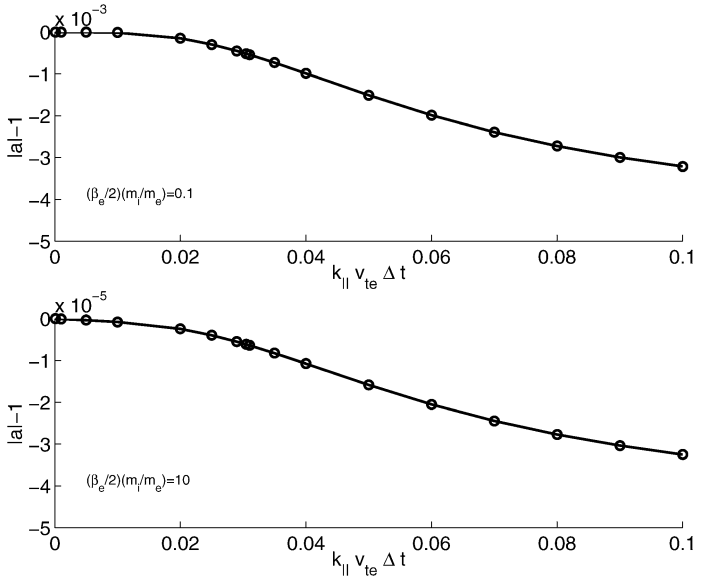

Fig. 6. Amplification factor vs. normalized time step for the kinetic Alfvén wave at $k_\perp \rho_s = 0.03$ using a partially implicit Adams–Bashforth algorithm.

twice as strict as that found for the ADI algorithm. Though it is the physical mode which becomes numerically unstable for the ADI algorithm, while just the computational mode (which is introduced solely as a result of the numerical discretization) becomes unstable for the partially implicit Adams–Bashforth algorithm, the more severe stability limit for the partially implicit Adams–Bashforth algorithm makes it highly unpractical for edge gyrokinetic simulations. We have also tried a partially implicit algorithm using a third-order Adams–Bashforth for

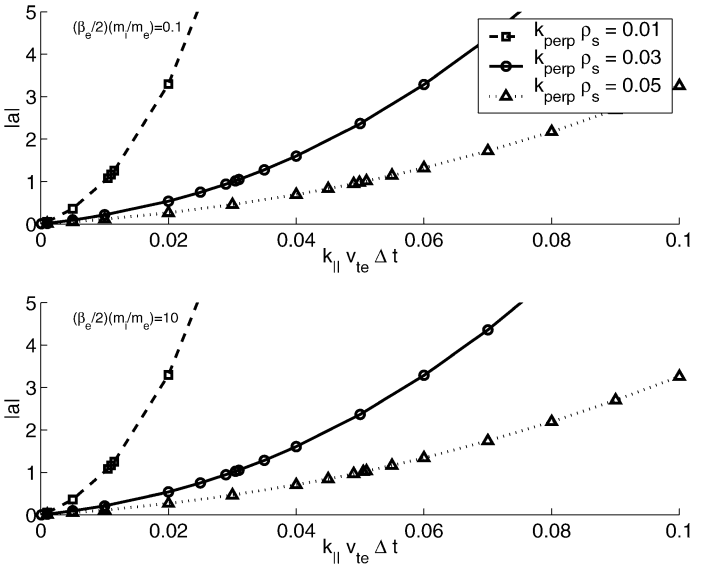

Fig. 7. Amplification factor vs. normalized time step for the computational mode using a partially implicit Adams–Bashforth algorithm.

the electric field terms and found that it had an even smaller stability limit on the time step than the second-order Adams–Bashforth method.

For comparison, we also consider the stability of a fully explicit Adams–Bashforth algorithm. For this case, we treat the parallel derivative operator term as well as the field terms with a second-order Adams–Bashforth scheme. Using our usual ansatz, we find that

$$\hat{g}_e = \frac{Z_e e}{T_{0e}} F_{Me}\left(\hat{\Phi} - \frac{v_\parallel}{c}\hat{A}_\parallel\right)\frac{k_\parallel v_\parallel}{\hat{\tilde{\omega}} - k_\parallel v_\parallel}, \tag{34}$$

where here the effective frequency is defined such that $\frac{a(a-1)}{3a-1} = -i\hat{\tilde{\omega}}\Delta t/2$.

The result of Eq. (34) has the same form as the exact time continuous result (i.e. the equivalent of Eq. (4) for our starting equations) and is thus surprisingly not subject to the same stability restrictions $|k_\parallel v_{te}\Delta t| \lesssim |k_\perp \rho_s|$ as the ADI and partially-implicit Adams–Bashforth algorithms for this problem.

As is well known, a second-order Adams–Bashforth algorithm does induce a small amount of artificial growth. The amplification factor for this case is given by

$$a = \frac{1}{2}\left[1 - \frac{3}{2}i\hat{\omega}\Delta t \pm \sqrt{1 - i\hat{\tilde{\omega}}\Delta t - \frac{9}{4}(\hat{\tilde{\omega}}\Delta t)^2}\right] \tag{35}$$

as shown in Fig. 8 for the low and high $(\beta_e/2)(m_i/m_e)$ cases. We show both the physical Alfvén mode, for which $|a| \to 1$ in the limit of $\Delta t \to 0$, and the unphysical computational mode, for which $|a| \to 0$ in the limit of $\Delta t \to 0$. These levels of artificial growth would be quite adequate for many gyrokinetic turbulence simulations. One just needs to keep the time step sufficiently small so that this artificial amplification is small compared to physical dissipation mechanisms, or in turbulent systems, small compared to the rate at which nonlinear interactions take energy out of undamped modes and transfer energy to damped modes. Overall, this analysis suggests that even a second-order purely explicit Adams–Bashforth algorithm can be better at long wavelengths than the ADI or partially implicit Adams–Bashforth algorithms explored here.

By going to an even higher-order explicit algorithm, one can completely eliminate artificial growth over a range of time step, since the stability boundaries in the complex $\omega\Delta t$ plane are well known [24,25]. Often a third-order

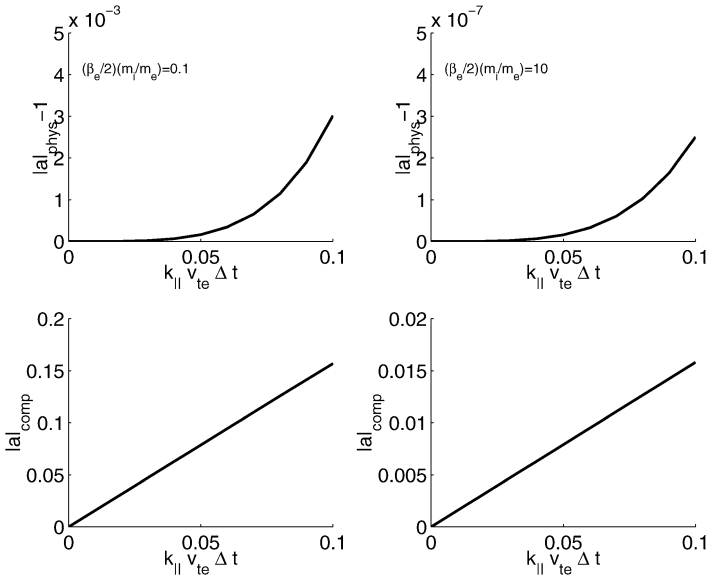

Fig. 8. Amplification factor vs. normalized time step for the physical mode and the computational mode at $k_\perp \rho_s = 0.03$ using a fully explicit Adams–Bashforth algorithm.

Adams–Bashforth (AB3) or a fourth-order Runge–Kutta (RK4) algorithm is chosen to provide a good balance between maximum stable time step and computational cost. On the real ω axis, the maximum stable time step for AB3 is $|\omega|\Delta t = 0.72$, while for purely damped modes the maximum stable time step for AB3 is $|\omega|\Delta t = 0.55$. The stability limit of a fourth-order Runge–Kutta algorithm is comparable (after dividing by 4 to account for the 4 intermediate steps that make up a full step of a fourth-order Runge–Kutta algorithm), equivalent to $|\omega|\Delta t = 0.71$ for real ω, and $|\omega|\Delta t = 0.70$ for purely damped modes. The dispersion relation that follows from the third-order Lorentzian in Eq. (18) has 3 roots. For $(\beta_e/2)(m_i/m_e) = 0.1$ and $k_\perp \rho_s = 0.03$, these roots are $\omega = \pm 3.2 k_\parallel v_{te}$ (the Alfvén waves) and $\omega = -5.2 i k_\parallel v_{te}$ (a heavily damped entropy mode related to Landau damping). This heavily damped mode would set a stability limit for an RK4 algorithm of $k_\parallel v_{te} \Delta t = 0.14$, 2.2 times better than the stability limit of the ADI algorithm. For lower $k_\perp$ modes, a fully explicit RK4 or AB3 algorithm would be even better in comparison to the ADI and partially implicit algorithms studied here.

6. Discussion and summary

Though ADI algorithms applied to many problems have the nice property that they are absolutely stable for arbitrarily large time step, and though Kupfer's ADI algorithm has previously worked for a related plasma kinetic problem, we have found that the implementation in a gyrokinetic problem yields a severe time step restriction. It is somewhat surprising not only that the ADI algorithm for this problem has a stability limit, but also that this stability limit is so short, even worse than some fully explicit algorithms.

Specifically, for a test problem of a shear kinetic Alfvén wave at small $k_\perp \rho_i$, the ADI algorithm is numerically unstable for $\Delta t/2 > |k_\perp \rho_s/(k_\parallel v_{te})|$ in both the low $(\beta_e/2)(m_i/m_e)$ regime and the high $(\beta_e/2)(m_i/m_e)$ regime, where the Alfvén wave is slower than the electron thermal speed and all of the important dynamics would seem to be well resolved. Furthermore, this stability problem is not unique to the ADI algorithm, as a partially implicit Adams–Bashforth scheme yielded a restriction on the time step twice as low as the ADI algorithm for this problem. A simple set of gyrofluid model equations was constructed to illustrate the source of the problems. In the simple

electrostatic limit, one of operators used in the alternating implicit steps was nilpotent and was not diagonalizable, so that what appeared to be an implicit step was actually equivalent to an explicit step.

Eventually, the best approach for kinetic edge microturbulence simulations might be a fully implicit algorithm for the linear terms, perhaps employing preconditioned Krylov solvers from an advanced package such as PETSc [26] or SUNDIALS [27]. A key to successful use of such iterative methods is a good preconditioner. As part of the preconditioning, one might use precomputed plasma response matrices as used in the linearly fully implicit GS2 algorithm [7] or a similar approach used in GYRO [10]. As a starting point short of these more complicated implicit methods, one might use a fully explicit fourth-order Runge–Kutta algorithm. Though we have found severe stability restrictions for a standard ADI algorithm and one form of a semi-implicit Adams–Bashforth algorithm, perhaps there is some other variant of a semi-implicit algorithm for parallel dynamics that could be more successful.

Acknowledgements

We thank Gary Kerbel, Bruce Cohen, and Bill Nevins for useful discussions regarding this problem. This research was supported by U.S. DOE contract # DE-AC02-76CH03073, and EAB was supported in part by a fellowship through the PICASso program at Princeton, funded under NSF IGERT grant # DGE-9972930.

Appendix A

Eqs. (6)–(8) provide a relatively simple problem with 1 spatial and 1 velocity coordinate that could provide a very useful testbed for studying alternative gyrokinetic algorithms. With appropriate normalizations, these equations can be written as the following set of integro-differential equations:

$$\frac{\partial g_e}{\partial t} + v_\parallel \frac{\partial g_e}{\partial z} = v_\parallel F_{Me}\left(\frac{\partial \Phi}{\partial z} - v_\parallel \frac{\partial A_\parallel}{\partial z}\right), \tag{A.1}$$

$$k_\perp^2 \Phi = -\int \mathrm{d}v_\parallel\, g_e, \tag{A.2}$$

$$\left(k_\perp^2 + \hat{\beta}\right) A_\parallel = -\hat{\beta} \int \mathrm{d}v_\parallel\, v_\parallel g_e, \tag{A.3}$$

where $k_\perp^2$ is normalized to $Z_i \rho_s^2$ and $\hat{\beta} \equiv (\beta_e/2)\frac{m_i}{m_e}$. The electron thermal velocity v_{te} has been normalized to unity, so that the Maxwellian equilibrium is $F_{Me} = \exp(-v_\parallel^2/2)/\sqrt{2\pi}$. Because $v_A^2 = v_{te}^2/\hat{\beta}$, Eq. (5) becomes

$$\omega^2 = \frac{k_\parallel^2/\hat{\beta}}{1 + k_\perp^2/\hat{\beta}}. \tag{A.4}$$

A more complete general comprehensive test of the numerical stability of an algorithm for this problem would include a typical range of parameters, particularly, $k_\perp = 0.01$–10 and $\hat{\beta} = 0.1$–10, and perhaps also the addition of collisions in Eq. (A.1) to test the collisional component of the algorithm.

References

[1] Y. Chen, S.E. Parker, Phys. Plasmas 8 (2001) 441.
[2] B.I. Cohen, A.M. Dimits, W.M. Nevins, et al., Phys. Plasmas 9 (2002) 251.
[3] B.I. Cohen, A.M. Dimits, W.M. Nevins, et al., Phys. Plasmas 9 (2002) 1915.
[4] Y. Chen, S.E. Parker, J. Comput. Phys. 189 (2003) 463.

[5] T. Dannert, F. Jenko, Comput. Phys. Comm. 163 (2004) 67.
[6] J. Candy, R. Waltz, J. Comput. Phys. 186 (2003) 545.
[7] M. Kotschenreuther, G. Rewoldt, W.M. Tang, Comput. Phys. Comm. 125 (1995) 88.
[8] F. Jenko, W. Dorland, M. Kotschenreuther, B.N. Rogers, Phys. Plasmas 7 (2000) 1904.
[9] F. Jenko, Comput. Phys. Comm. 125 (2000) 196.
[10] J. Candy, R. Waltz, Phys. Rev. Lett. 91 (2003) 045001.
[11] W. Dorland, F. Jenko, M. Kotschenreuther, B.N. Rogers, Phys. Rev. Lett. 85 (2000) 5579.
[12] K. Kupfer, R.W. Harvey, O. Sauter, et al., Phys. Plasmas 3 (1996) 3644.
[13] W. Hundsdorfer, Math. Comput. 67 (1998) 183.
[14] D.W. Peaceman, H.H. Rachford Jr., J. SIAM 3 (1955) 28.
[15] J. Douglas, H.H. Rachford Jr., Trans. Amer. Math. Soc. 82 (1956) 421.
[16] E.A. Frieman, L. Chen, Phys. Fluids 25 (1982) 502.
[17] T.M. Antonsen, B. Lane, Phys. Fluids 23 (1980) 1205.
[18] P.J. Catto, W.M. Tang, D.E. Baldwin, Plasma Phys. 23 (1981) 639.
[19] P. Snyder, Ph.D. Thesis, Princeton University, 1999.
[20] T.S. Hahm, L. Chen, Phys. Fluids 28 (1985) 3061.
[21] W.W. Lee, J. Comput. Phys. 72 (1987) 243.
[22] Maple is a Symbolic Computational System from http://www.maplesoft.com.
[23] G.W. Hammett, W. Dorland, F. Perkins, Phys. Fluids B 4 (1992) 2052.
[24] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, 1971.
[25] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer, New York, 1999.
[26] S. Balay, K. Buschelman, V. Eijkhout, et al., PETSc Users Manual, ANL-95/11, Revision 2.1.5, 2004.
[27] A.C. Hindmarsh, P.N. Brown, K.E. Grant, et al., SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers, UCRL-JP20037, 2003.