Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 4518–4542

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An iterative semi-implicit scheme with robust damping

N.F. Loureiro [a,b,*], G.W. Hammett [b]

[a] CMPD, University of Maryland, College Park, MD 20742, USA
[b] Princeton Plasma Physics Laboratory, Princeton University, Princeton, NJ 08543, USA

Received 22 June 2007; received in revised form 2 January 2008; accepted 10 January 2008
Available online 26 January 2008

Abstract

An efficient, iterative semi-implicit (SI) numerical method for the time integration of stiff wave systems is presented. Physics-based assumptions are used to derive a convergent iterative formulation of the SI scheme which enables the monitoring and control of the error introduced by the SI operator. This iteration essentially turns a semi-implicit method into a fully implicit method. Accuracy, rather than stability, determines the timestep. The scheme is second order accurate and shown to be equivalent to a simple preconditioning method. We show how the diffusion operators can be handled so as to yield the property of robust damping, i.e. dissipating the solution at all values of the parameter $\mathcal{D}\Delta t$, where $\mathcal{D}$ is a diffusion operator and Δt the timestep. The overall scheme remains second order accurate even if the advection and diffusion operators do not commute. In the limit of no physical dissipation, and for a linear test wave problem, the method is shown to be symplectic. The method is tested on the problem of Kinetic Alfvén wave mediated magnetic reconnection. A Fourier (pseudo-spectral) representation is used. A two-field gyrofluid model is used and an efficacious k-space SI operator for this problem is demonstrated. CPU speed-up factors over a CFL-limited explicit algorithm ranging from ~20 to several hundreds are obtained, while accurately capturing the results of an explicit integration. Possible extension of these results to a real-space (grid) discretization is discussed.

PACS: 52.30Ex; 52.35Vd; 52.35Py; 52.35Hr; 52.65Tt

Keywords: Semi-implicit methods; Robust damping; Magnetic reconnection; Gyrofluid; Dispersive plasma waves; Symplectic methods

1. Introduction

One ubiquitous property of the dynamics of fluids and plasmas is their inherent multiscale character, both spatial and temporal [1,2]. This translates into a stiffness of the partial differential equations (PDEs) that are used to describe the physical systems of interest. The numerical solution of stiff systems of coupled PDEs is notoriously difficult, presenting a serious challenge to present day computational resources.

[*] Corresponding author. Present address: EURATOM/UKAEA Fusion Association, Culham Science Centre, OX14 3DB, U.K. Tel.: +44 (0) 1235 466541.

E-mail addresses: nuno.loureiro@ukaea.org.uk (N.F. Loureiro), hammett@pppl.gov (G.W. Hammett).

URLs: http://www.cscamm.umd.edu/cmpd/people/loureiro/ (N.F. Loureiro), http://w3.pppl.gov/~hammett/ (G.W. Hammett).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.015

Wave-induced temporal stiffness occurs when the characteristic time scale of interest is much slower than that of the fastest waves present in the system. Fortunately, however, it is often the case that it is not necessary to accurately capture these waves in order to obtain accurate solutions of the slowly evolving physics [3,4]. Examples are the gravity wave in weather modeling [4,5], or the Alfvén and whistler waves in studies of some instabilities in plasmas [6–12]. When that occurs, implicit and semi-implicit (SI) numerical methods have been employed as very powerful tools to potentially allow the stable and accurate time integration of the PDEs on timescales which largely exceed those imposed by these high frequency, fast waves (see, for example, the paper by Knoll et al. [13] and references therein). More conventional and easy to implement explicit numerical schemes are limited by the well-known Courant–Friedrichs–Lewy (CFL) condition and thus, in order to be stable, require integration timesteps of the order of the inverse of the maximum frequency present – a very severe, and generally unfeasible, requirement.

There are advantages and disadvantages to both purely implicit and SI numerical methods. Implicit and SI methods can both use timesteps much larger than the CFL limit. Implicit methods are more robust and accurate but can be harder to implement, in some cases requiring more extensive modifications to the structure of an existing code. Direct solution of an implicit scheme can require the storage and inversion of impractically large matrices (though direct sparse matrix solvers with reordering can be effective for some problems that are not too large). Thus, implicit methods often employ preconditioned iterative methods (such as Krylov methods like Conjugate-Gradients or GMRES), sometimes in combination with multigrid or FFT solvers (see for example Ref. [14,13], which also discuss non-linear problems using Jacobian–Free Newton–Krylov solvers). To obtain a net speed-up, it is often crucial to have a sufficiently good preconditioner that can still be efficiently inverted. While there are many options [14,13,15,16], finding such a preconditioner is problem-dependent and can be a non-trivial task. If no preconditioner is used, a Krylov iterative solver applied directly to a hyperbolic wave/advection problem will usually give at most an order unity speed-up in net CPU time, because the larger timestep allowed by an implicit method is largely offset by the increased number of iterations required per timestep. However, when a good preconditioner exists, or a multigrid or FFT solver can be used, then significant savings in CPU time are possible.

SI methods are usually simpler to implement and work well for some problems. However, they have to be used with some care, because in some parameter regimes there can be splitting errors (due to the different treatment of the operators involved) and/or uncontrolled approximations to the equations that are made in their derivation (such as linearization of the fast wave physics). The reader is referred to the work of Knoll and Keyes [14] and Knoll et al. [13] for an overview of SI methods in a variety of applications and their comparison to the JFNK approach to fully implicit methods. In this paper, we revisit SI methods as they are traditionally formulated in the field of plasma physics [17,18,6,19]. Using physics-based assumptions, we are able to consistently derive an iterative formulation of SI schemes, which lends itself to an efficient way of monitoring and controlling the error introduced by the SI operator. Accuracy, rather than stability, determines the timestep. It is shown that this iterative extension of an SI scheme leads to essentially a fully implicit method with the SI operator providing a preconditioner. We also demonstrate how the diffusive terms can be treated so as to yield the property of robust damping, i.e. the consistent damping of all the modes present in the simulation domain in the limit $\Delta t \to \infty$. The SI method presented here is second order accurate even if the diffusion and non-linear advection operators do not commute.

The method is then applied to the problem of gyrofluid magnetic reconnection. This is a very stiff application due to the presence of a high frequency dispersive wave, the Kinetic Alfvén wave (KAW). A Fourier-space, pseudo-spectral, discretization of the equations is used. Compared to explicit integrations schemes, our SI scheme yields CPU speed-up factors of $\sim$20 to several hundreds while accurately reproducing the results of an explicit integration.

Before proceeding further, we caution the reader that, although our derivation of the SI algorithm presented in Section 2.1 is general, we have only tested it in a pseudo-spectral implementation, and only for a two-field problem that involves a single wave family (a single dispersion relation $\pm\omega$ vs. $\boldsymbol{k}$). The efficiency of this method relies heavily on the effectiveness of the SI operator used, which may be complicated for problems with strong spatial inhomogeneities, strong anisotropies, or for multi-field problems with multiple waves. Also, our treatment of diffusion terms with a robust damping algorithm has been tested only in a Fourier representation, although we discuss, in the conclusions section, how a similar result might be achieved by grid codes. While we believe this SI algorithm could be useful in real-space codes at least for certain types of problems, the reader interested in other applications should proceed with caution. Finally, we note that although our focus is in the plasma physics field, the derivation of the iterative SI method presented is general and should apply well to other stiff wave physics phenomena that can be described by means of advection–diffusion PDEs.

2. Numerical scheme

2.1. Derivation of the method

As mentioned above, we are interested in dealing with coupled sets of two-dimensional advection–diffusion partial differential equations. Let us consider the following set as our general model:

$$\frac{\partial\psi}{\partial t} = \mathcal{F}(\phi,\psi) + (\eta\nabla^2 - \eta_{\mathrm{H}}\nabla^4)\psi, \tag{1}$$

$$\frac{\partial\phi}{\partial t} = \mathcal{G}(\phi,\psi) + (\nu\nabla^2 - \nu_{\mathrm{H}}\nabla^4)\phi. \tag{2}$$

Here, ϕ, ψ are arbitrary scalar fields, the operators $\mathcal{F}$, $\mathcal{G}$ represent the non-linear advection-like operators, η and η_{H} are the resistivity and hyper-resistivity and ν and ν_{H} the viscosity and hyper-viscosity, respectively. In Fourier-space, we can define the diffusion operators as

$$\mathcal{D}_\eta = \eta k_\perp^2 + \eta_{\mathrm{H}} k_\perp^4, \quad \mathcal{D}_\nu = \nu k_\perp^2 + \nu_{\mathrm{H}} k_\perp^4 \tag{3}$$

and thus rewrite the above set (in Fourier-space), as

$$\frac{\partial\psi_k}{\partial t} = \overline{\mathcal{F}}(\phi,\psi) - \mathcal{D}_\eta\psi_k, \tag{4}$$

$$\frac{\partial\phi_k}{\partial t} = \overline{\mathcal{G}}(\phi,\psi) - \mathcal{D}_\nu\phi_k. \tag{5}$$

Here, the k subscripts denote variables in Fourier-space and $\overline{\mathcal{F}}$, $\overline{\mathcal{G}}$ represent the Fourier transforms of the non-linear operators $\mathcal{F}$, $\mathcal{G}$. For notational simplicity, we will drop both the subscripts and the over bars: it is understood that we are working in Fourier-space. We note also that above we have chosen a fourth order operator to represent the hyperviscosity, but the following derivation holds irrespective of the order chosen for this operator.

The diffusion terms can be integrated analytically by means of the following variable transformations:

$$\psi = \mathrm{e}^{-\mathcal{D}_\eta t}\tilde{\psi}, \quad \phi = \mathrm{e}^{-\mathcal{D}_\nu t}\tilde{\phi}. \tag{6}$$

Eqs. (4) and (5) thus become:

$$\frac{\partial\tilde{\psi}}{\partial t} = \mathrm{e}^{\mathcal{D}_\eta t}\mathcal{F}(\phi,\psi), \tag{7}$$

$$\frac{\partial\tilde{\phi}}{\partial t} = \mathrm{e}^{\mathcal{D}_\nu t}\mathcal{G}(\phi,\psi). \tag{8}$$

We now apply a Crank–Nicolson discretization scheme (spatial indices omitted for simplicity):

$$\frac{\tilde{\psi}^{n+1} - \tilde{\psi}^n}{\Delta t} = \frac{1}{2}\mathcal{F}(\phi^n,\psi^n) + \frac{1}{2}\mathrm{e}^{\mathcal{D}_\eta\Delta t}\mathcal{F}(\phi^{n+1},\psi^{n+1}), \tag{9}$$

$$\frac{\tilde{\phi}^{n+1} - \tilde{\phi}^n}{\Delta t} = \frac{1}{2}\mathcal{G}(\phi^n,\psi^n) + \frac{1}{2}\mathrm{e}^{\mathcal{D}_\nu\Delta t}\mathcal{G}(\phi^{n+1},\psi^{n+1}). \tag{10}$$

(Without loss of generality, we have defined $t_n = 0$, so $t_{n+1} = \Delta t$.)

Note that the handling of the diffusion terms through the variable substitutions of (6) is a trivial step for spectral and pseudo-spectral codes, but more complicated in real-space codes. However, the derivation of the semi-implicit method that follows holds irrespective of how the diffusion terms are handled – this is only important for the robust damping property, to be discussed later on in Section 2.6. In Section 5, we briefly discuss how a similar treatment of the diffusive terms can be achieved in real-space codes.

Usual implicit schemes would now invert Eqs. (9) and (10) to determine the unknowns ψ^{n+1}, ϕ^{n+1} in terms of the known quantities ψ^n, ϕ^n. This may be done by, for example, an iterative Jacobian–Free Newton–Krylov solver. A Newton method turns this non-linear root problem into a series of linear problems. These linear problems can often involve very large sparse matrices. If a direct inversion was done, the very large timestep enhancements that can be obtained relative to an explicit scheme might be severely offset by the increased CPU time per step. Jacobian–Free Krylov iteration algorithms can be used to solve the resulting linear problems without explicitly forming the matrices. If these iterative methods converge quickly enough (which for hyperbolic problems may require very effective preconditioners), then the implicit method can provide a significant improvement in CPU time relative to an explicit algorithm.

The aim of our method, like more complicated iterative methods, is to circumvent the need to directly invert a large linear problem. We will use a simple iterative method, with a semi-implicit operator providing an effective physics-based preconditioner. The first step is to reformulate Eqs. (9) and (10) in an iterative way. To this effect, we begin by rewriting these equations as

$$\frac{\tilde{\psi}^{n+1,p+1}-\tilde{\psi}^{n}}{\Delta t}=\frac{1}{2}\mathcal{F}(\phi^{n},\psi^{n})+\frac{1}{2}\mathrm{e}^{\mathcal{D}_{\eta}\Delta t}\mathcal{F}(\phi^{n+1,p+1},\psi^{n+1,p+1}), \tag{11}$$

$$\frac{\tilde{\phi}^{n+1,p+1}-\tilde{\phi}^{n}}{\Delta t}=\frac{1}{2}\mathcal{G}(\phi^{n},\psi^{n})+\frac{1}{2}\mathrm{e}^{\mathcal{D}_{\nu}\Delta t}\mathcal{G}(\phi^{n+1,p+1},\psi^{n+1,p+1}). \tag{12}$$

Next, we introduce one simple, but crucial, physics-based assumption: that the stiff waves arise through the coupling of the equations and are well described by the assumptions

$$\mathcal{F}(\psi^{n+1,p+1},\phi^{n+1,p+1})\approx\mathcal{F}(\psi^{n+1,p},\phi^{n+1,p+1})\approx\mathcal{F}(\psi^{n+1,p},\phi^{n+1,p})+\frac{\delta\mathcal{F}}{\delta\phi}(\phi^{n+1,p+1}-\phi^{n+1,p}), \tag{13}$$

$$\mathcal{G}(\psi^{n+1,p+1},\phi^{n+1,p+1})\approx\mathcal{G}(\psi^{n+1,p+1},\phi^{n+1,p})\approx\mathcal{G}(\psi^{n+1,p},\phi^{n+1,p})+\frac{\delta\mathcal{G}}{\delta\psi}(\psi^{n+1,p+1}-\psi^{n+1,p}). \tag{14}$$

In other words, we are assuming that the dependence of $\mathcal{F}$ on ψ is relatively weak (i.e. the fast timescales arise via the dependence of $\mathcal{F}$ on ϕ, not on ψ) and can be approximated from the previous iteration p, while it depends more sensitively on ϕ and so we need to keep its dependence on the present $p+1$ iteration value by using a Taylor-series (and *vice-versa* for $\mathcal{G}$). This is certainly true in the linear limit for our model, as shown by Eqs. (63) and (64). Most wave phenomena involve oscillations between two different variables (for example, pressure and velocity for sound waves, or position and velocity for a pendulum), so that approximations like the above are often justified. A key step when extending this approach to more complicated systems of equations, which might contain multiple waves, is to identify the key terms involved in the fast wave dynamics.

[We have written the Jacobians in the above Taylor-series expansions as functional derivatives, such as $\delta\mathcal{F}/\delta\phi$, since $\mathcal{F}$ may be a non-local functional that operates on ϕ. After spatial discretization, $\delta\mathcal{F}/\delta\phi$ will in general be a large sparse matrix. However, in our final algorithm all we have to evaluate is an approximation to $(\delta\mathcal{F}/\delta\phi)(\delta\mathcal{G}/\delta\psi)$. This approximation will be local in Fourier space, and so for simplicity we will replace the functional derivative notation with the standard notation $\partial\mathcal{F}/\partial\phi$. Note that in our final algorithm we never actually need to explicitly evaluate the Jacobian, just the action of a Jacobian on a vector, and so this formulation is "Jacobian-free".]

With this assumption, the non-linear future time term in Eq. (9) can be simplified by means of the Taylor expansion, i.e.

$$\frac{\tilde{\psi}^{n+1,p+1}-\tilde{\psi}^{n}}{\Delta t}=\frac{1}{2}\mathcal{F}(\phi^{n},\psi^{n})+\frac{1}{2}\mathrm{e}^{\mathcal{D}_{\eta}\Delta t}\left[\mathcal{F}(\phi^{n+1,p},\psi^{n+1,p})+\frac{\partial\mathcal{F}}{\partial\phi}(\phi^{n+1,p+1}-\phi^{n+1,p})\right]. \tag{15}$$

An expression for the last term on the RHS of this equation is straightforwardly obtained (for $p>0$) by taking the difference between Eq. (12) evaluated for two successive values of p:

$$\phi^{n+1,p+1} - \phi^{n+1,p} = \frac{\Delta t}{2}\frac{\partial \mathcal{G}}{\partial \psi}(\psi^{n+1,p+1} - \psi^{n+1,p}), \tag{16}$$

where again we have used the approximations of Eq. (13). Undoing the variable substitution of Eqs. (6) and (15) thus becomes:

$$\psi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,p},\psi^{n+1,p}) + \frac{\Delta t^2}{4}\frac{\partial \mathcal{F}}{\partial \phi}\frac{\partial \mathcal{G}}{\partial \psi}(\psi^{n+1,p+1} - \psi^{n+1,p}). \tag{17}$$

At this point, we note that if the above equation is iterated until convergence, i.e. $|\psi^{n+1,p+1} - \psi^{n+1,p}| < \epsilon$, where ϵ is the prescribed accuracy, the last term becomes negligible. Therefore, it is not important to retain its exact functional form as long as convergence is reached at each timestep. We can thus simplify Eq. (17) by replacing the last term on the RHS with an analytically invertible approximation. We use:

$$\frac{\partial \mathcal{F}}{\partial \phi}\frac{\partial \mathcal{G}}{\partial \psi} = -\hat{\omega}^2, \tag{18}$$

where $\hat{\omega}$ is given by the dispersion relation, or an approximation to it. [Specific forms for the operator $\hat{\omega}^2$ will be discussed later, for now it should just be thought of as an operator that is relatively easy to invert.]

The steps we have just taken are similar to those for some previous SI derivations and share some key advantages for high frequency waves, since it is only the combination $(\partial\mathcal{F}/\partial\phi)(\partial\mathcal{G}/\partial\psi)$ that has to be approximated by the semi-implicit operator, and not $(\partial\mathcal{F}/\partial\phi)$ and $(\partial\mathcal{G}/\partial\psi)$ separately. As discussed in the SI literature, this is much simpler because one does not have to worry about getting the sign of the approximation $\hat{\omega}$ correct (i.e. one does not have to worry about the direction of propagation of the waves). Chacón and Knoll [11,14,13] have shown how this kind of physics-based semi-implicit operator can be related to a Schur complement preconditioner, turning an originally hyperbolic problem into a diagonally dominant parabolic problem that can be efficiently solved by preconditioned Krylov (if the preconditioner is sufficiently effective) and/or multigrid iterative techniques. In some cases, the resulting problem can also be efficiently solved with FFTs, which we use here.

So far, we have:

$$\psi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,p},\psi^{n+1,p}) - \frac{\hat{\omega}^2\Delta t^2}{4}(\psi^{n+1,p+1} - \psi^{n+1,p}), \tag{19}$$

$$\phi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\nu \Delta t}\phi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\nu \Delta t}\mathcal{G}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{G}(\phi^{n+1,p},\psi^{n+1,p+1}). \tag{20}$$

Eqs. (19) and (20) constitute the basis of our iterative semi-implicit method. Since this iteration converges (as we will later show) to the Crank–Nicolson (CN) solution for an arbitrary initial condition (within some domain), it is tempting to just use these equations with the initial guess $\psi^{n+1,0} = \psi^n$, $\phi^{n+1,0} = \phi^n$. However, one finds that if one does this for an undamped wave test problem, the amplitude of the solution converges to the CN solution from above, so the algorithm is numerically unstable for any finite number of iterations. One can demonstrate this by considering the test problem $\mathcal{F} = \omega\phi$, $\mathcal{G} = -\omega\psi$, $\mathcal{D}_\eta = \mathcal{D}_\nu = 0$, finding that the squared amplification factor for the first iteration gives $|A|^2 = ((\psi^{n+1,1})^2 + (\phi^{n+1,1})^2)/((\psi^n)^2 + (\phi^n)^2) = 1 + 2\omega^2\Delta t^2/(4 + \hat{\omega}^2\Delta t^2)$. For high frequency modes that we are trying to treat implicitly, this becomes $|A|^2 \approx 1 + 2\omega^2/\hat{\omega}^2$, which is significantly larger than unity. While further iterations would converge towards the Crank–Nicolson solution, which has $|A|^2 = 1$, this significant numerical amplification would still be problematic, and would require that the semi-implicit operator be an extremely good approximation to the real dynamics in order to converge quickly. Furthermore, it is somewhat bothersome that this algorithm would give such a large numerical amplification, since it is supposed to be an approximation to the stable Crank–Nicolson algorithm given by Eqs. (9) and (10). However, note Eqs. (19) and (20) were derived using Eq. (16), which is valid only for $p > 0$, so an expression for the initial $p = 0$ step of the iteration requires further consideration. It turns out that a more careful treatment of the initial step will also make the final algorithm stable. Using Eq. (14) in Eq. (20) for $p = 0$, we have:

$$\phi^{n+1,1} = \mathrm{e}^{-\mathcal{D}_\nu \Delta t}\phi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\nu \Delta t}\mathcal{G}(\phi^n,\psi^n) + \frac{\Delta t}{2}\left[\mathcal{G}(\phi^{n+1,0},\psi^{n+1,0}) + \frac{\partial \mathcal{G}}{\partial \psi}(\psi^{n+1,1} - \psi^{n+1,0})\right]. \tag{21}$$

Let us define the quantity $\phi^{n+1,*}$ as

$$\phi^{n+1,*} = \mathrm{e}^{-\mathcal{D}_\nu \Delta t}\phi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\nu \Delta t}\mathcal{G}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{G}(\phi^{n+1,0},\psi^{n+1,0}), \tag{22}$$

which reduces Eq. (21) to:

$$\phi^{n+1,1} = \phi^{n+1,*} + \frac{\Delta t}{2}\frac{\partial \mathcal{G}}{\partial \psi}(\psi^{n+1,1} - \psi^{n+1,0}). \tag{23}$$

Substituting this into Eq. (15) evaluated at $p = 0$, and using Eq. (14), yields

$$\psi^{n+1,1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,*},\psi^{n+1,0}) + \frac{\Delta t^2}{4}\frac{\partial \mathcal{F}}{\partial \phi}\frac{\partial \mathcal{G}}{\partial \psi}(\psi^{n+1,1} - \psi^{n+1,0}), \tag{24}$$

where again the last term on the RHS can be simplified making use of approximation (18). Eq. (24) followed by Eq. (20) (evaluated for $p = 0$) would give a first order accurate prediction of $(\psi^{n+1,1},\phi^{n+1,1})$. This could then be iterated once with Eqs. (19) and (20) to obtain a second order accurate calculation for $(\psi^{n+1,2},\phi^{n+1,2})$. However, a small modification to the prediction step can give a second order accurate results for $(\psi^{n+1,1},\phi^{n+1,1})$. This is done by replacing $\psi^{n+1,0}$ in the non-linear term of Eq. (24) with a first order accurate approximation, $\psi^{n+1,*}$, so that Eq. (24) becomes

$$\psi^{n+1,1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,*},\psi^{n+1,*}) - \frac{\hat{\omega}^2\Delta t^2}{4}(\psi^{n+1,1} - \psi^{n+1,0}), \tag{25}$$

where $\psi^{n+1,*}$ is defined as

$$\psi^{n+1,*} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,0},\psi^{n+1,0}). \tag{26}$$

Note that the replacement of $\mathcal{F}(\phi^{n+1,*},\psi^{n+1,0})$ with $\mathcal{F}(\phi^{n+1,*},\psi^{n+1,1})$ is consistent with the assumption of Eq. (13) that the dependence of $\mathcal{F}$ on ψ is relatively weak, but is useful to ensure second order accuracy of the calculation of $\psi^{n+1,1}$. Similarly, when calculating $\phi^{n+1,0}$ from Eq. (20) for $p = 0$, the term $\mathcal{G}(\phi^{n+1,0},\psi^{n+1,1})$ is replaced by $\mathcal{G}(\phi^{n+1,*},\psi^{n+1,1})$. This gives a second order accurate result for $\phi^{n+1,1}$. The choice $\psi^{n+1,0} = \psi^n$, $\phi^{n+1,0} = \phi^n$ closes the specification of our semi-implicit scheme.

The final equations for the iterative SI algorithm can be written in a compact form as follows:

$$\phi^{n+1,*} = \mathrm{e}^{-\mathcal{D}_\nu \Delta t}\phi^n + \frac{\Delta t}{2}(1 + \mathrm{e}^{-\mathcal{D}_\nu \Delta t})\mathcal{G}(\phi^n,\psi^n), \tag{27}$$

$$\psi^{n+1,*} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}(1 + \mathrm{e}^{-\mathcal{D}_\eta \Delta t})\mathcal{F}(\phi^n,\psi^n), \tag{28}$$

$$\psi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,p},\psi^{n+1,p}) - \frac{\hat{\omega}^2\Delta t^2}{4}(\psi^{n+1,p+1} - \bar{\psi}^{n+1,p}), \tag{29}$$

$$\phi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\nu \Delta t}\phi^n + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\nu \Delta t}\mathcal{G}(\phi^n,\psi^n) + \frac{\Delta t}{2}\mathcal{G}(\phi^{n+1,p},\psi^{n+1,p+1}). \tag{30}$$

where now $\phi^{n+1,0} = \phi^{n+1,*}$, $\psi^{n+1,0} = \psi^{n+1,*}$ and $\bar{\psi}^{n+1,0} = \psi^n$, $\bar{\psi}^{n+1,p} = \psi^{n+1,p}$ for $p > 0$.

We mentioned above that these equations ought to be iterated until convergence, at which point the last term on the RHS of Eq. (29) is negligible and these equations become equivalent to the fully implicit Eqs. (9) and (10). The relative importance of this term can be quantified by

$$\mathcal{E}_{ji}^{p+1} = \left|\frac{\mathcal{L}\left(\psi_{ji}^{n+1,p+1} - \bar{\psi}_{ji}^{n+1,p}\right)}{\sqrt{\frac{1}{N}\sum_{ji}\left|\psi_{ji}^{n+1,p+1} - \psi_{ji}^{n}\right|^2}}\right|, \tag{31}$$

where $\mathcal{L} = \hat{\omega}^2\Delta t^2/4$, j, i are the Fourier (or real) space grid point indexes and N is the total number of grid points. The timestep of the integration and/or the number of p iterations is thus determined by the requirement:

$$\max\left|\mathcal{E}_{ji}^{p+1}\right| < \epsilon, \tag{32}$$

where ϵ is the user prescribed accuracy. We emphasize that this error is a measure of the convergence of the iteration scheme to the Crank–Nicolson difference equation. It is not a direct control of the overall error in the integration (which, when converged, will be that of a CN scheme). For convenience, we also define the total number of iterations of Eqs. (29) and (30) as $p_{\max}$. That is, $p_{\max}$ is the number of times the semi-implicit operator is inverted in corrector steps. $p_{\max} = 1$ evaluates Eqs. (29) and (30) once, for $p = 0$, to determine $(\psi^{n+1,1}, \phi^{n+1,1})$. $p_{\max} = 2$ evaluates Eqs. (29) and (30) twice, for $p = 0$, 1, and calculates up to $(\psi^{n+1,2}, \phi^{n+1,2})$.

There are several variations of SI methods and their derivations in the literature, some using a more heuristic approach of adding and subtracting an operator to the RHS of one of the equations, and some taking a more systematic approach to deriving an SI operator. Also, some SI algorithms have used a leap-frog-like staggered time grid. Our final result is closest to certain SI methods using non-staggered grids. For example, our Eqs. (27)–(30) for $p_{\max} = 1$ are equivalent to Eqs. (11)–(14) of [18]. The formulation presented here makes it clear how to extend other semi-implicit methods to use additional iterations if desired (in this regard, note that it is important that the definition of $\bar{\psi}^{n+1,p}$ is different on the first $p = 0$ corrector step than in later iterations). These additional iterations provide a simple way to turn a semi-implicit method into a fully implicit method, and further demonstrate the relation between SI ideas and preconditioning for iterative methods, as discussed in Section 2.4.

2.2. Linear stability analysis

We will now show under which conditions the method described by Eqs. (27)–(30) is unconditionally stable with respect to the timestep Δt. For simplicity, we neglect the diffusive terms (since these terms are treated implicitly, they have a stabilizing effect). In line with the assumptions of (13), we assume that, in the linear regime, the following is true:

$$\begin{aligned}\mathcal{F}(\phi,\psi) &\approx f\phi,\\ \mathcal{G}(\phi,\psi) &\approx g\psi.\end{aligned} \tag{33}$$

Taking the simplest case of $p_{\max} = 1$ (i.e. a predictor–corrector scheme) Eqs. (27)–(30) can be written in matrix form as

$$\begin{bmatrix} 1 & -\Delta t/2g \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \phi^{n+1,1} \\ \psi^{n+1,1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t/2g \\ \frac{\Delta t f}{1+\mathcal{L}} & 1+\frac{\Delta t^2 fg}{2(1+\mathcal{L})} \end{bmatrix}\begin{bmatrix} \phi^{n} \\ \psi^{n} \end{bmatrix}, \tag{34}$$

where $\mathcal{L} = \hat{\omega}^2(\Delta t)^2/4$. Inversion of the matrix on the LHS yields the equation:

$$\begin{bmatrix} \phi^{n+1,1} \\ \psi^{n+1,1} \end{bmatrix} = A\begin{bmatrix} \phi^{n} \\ \psi^{n} \end{bmatrix}, \tag{35}$$

where A is the amplification matrix:

$$A = \begin{bmatrix} 1+\frac{fg\Delta t^2}{2\mathcal{L}+2} & \frac{\Delta t g(fg\Delta t^2+4\mathcal{L}+4)}{4(\mathcal{L}+1)} \\ \frac{\Delta t f}{\mathcal{L}+1} & 1+\frac{fg\Delta t^2}{2\mathcal{L}+2} \end{bmatrix}. \tag{36}$$

The method is unconditionally stable when the eigenvalues $\lambda_\pm$ of the matrix A are such that $|\lambda_\pm| \leqslant 1$. Using $fg = -\omega^2$, we obtain:

$$\lambda_\pm = \frac{4 + (\hat{\omega}^2 - 2\omega^2)\Delta t^2 \pm 2i\omega\Delta t\sqrt{4 + (\hat{\omega}^2 - \omega^2)\Delta t^2}}{4 + \Delta t^2\hat{\omega}^2} \tag{37}$$

It is straightforward to show that this algorithm is symplectic, $|\lambda_\pm| = 1$, if the argument of the square root is positive. Thus, we have unconditional stability for arbitrarily large $\omega\Delta t$ if $\hat{\omega}^2\Delta t^2 > \omega^2\Delta t^2 - 4$, or in the relevant limit of $\omega\Delta t \gg 1$, if $\hat{\omega}^2 > \omega^2$. Since the magnitude of the error at each iteration, expression (31) is proportional to $\hat{\omega}^2$, it is clear that setting $\hat{\omega}^2 \approx \omega^2$ (but $\hat{\omega}^2 > \omega^2$) is the best choice for stability and accuracy. (Of course, non-linearly ω^2 is not known, so one wishes to set $\hat{\omega}^2$ to be an approximate upper bound on the true operator ω^2.) Here we have demonstrated that the first corrector step, $p_{\max} = 1$, gives a solution that is symplectic (at least for this linear wave test problem). Additional iterations would eventually converge to the Crank–Nicolson solution, which is also symplectic. We have also proven that the algorithm is symplectic for arbitrary $p_{\max}$, as discussed at the end of the next section.

Finally, note that a Taylor-series expansion of expressions (37) yields:

$$\lambda_{\pm} = 1 \pm \mathrm{i}\omega\Delta t - \frac{\omega^2 \Delta t^2}{2} \pm \mathcal{O}(\Delta t^3), \tag{38}$$

consistent with the fact that the first order accurate predictor step of Eqs. (27) and (28) ensures that the corrector iterations are second order accurate.

2.3. Linear convergence rate

To determine the convergence rate of the iterative scheme presented in Eqs. (27)–(30), we set:

$$\phi^{n+1,p+1} = \phi^{n+1} + \delta\phi^{n+1,p+1}, \tag{39}$$

$$\psi^{n+1,p+1} = \psi^{n+1} + \delta\psi^{n+1,p+1}, \tag{40}$$

where the first term on the RHS is the converged solution and $\delta\phi^{n+1,p+1}$, $\delta\psi^{n+1,p+1}$ are the error at each iteration. Again using the linear approximations of Eq. (33), it is straightforward to obtain the equation:

$$\begin{bmatrix} \delta\phi^{n+1,p+1} \\ \delta\psi^{n+1,p+1} \end{bmatrix} = \begin{bmatrix} \frac{\Delta t^2 fg}{4(\mathcal{L}+1)} & \frac{\Delta t g \mathcal{L}}{2(\mathcal{L}+1)} \\ \frac{\Delta t f}{2(\mathcal{L}+1)} & \frac{\mathcal{L}}{\mathcal{L}+1} \end{bmatrix} \begin{bmatrix} \delta\phi^{n+1,p} \\ \delta\psi^{n+1,p} \end{bmatrix}. \tag{41}$$

The matrix on the RHS has the eigenvalues:

$$\lambda_1 = 0, \quad \lambda_2 = \frac{\Delta t^2(\hat{\omega}^2 - \omega^2)}{4 + \hat{\omega}^2 \Delta t^2}. \tag{42}$$

Clearly, λ_2 is the eigenvalue of interest. In the limit of $\hat{\omega}\Delta t \gg 1$, this expression simplifies to:

$$\lambda_2 = 1 - \frac{\omega^2}{\hat{\omega}^2}. \tag{43}$$

We showed in the previous section that $\hat{\omega}^2 \approx \omega^2$ is the best choice in terms of achieving stability while minimizing the error (31) at each iteration. Expression (43) shows that this is also the best choice in order to maximize the convergence rate. However, the iteration will converge as long as the semi-implicit operator is large enough. The stability condition for the first iteration, $\hat{\omega}^2\Delta t^2 > \omega^2\Delta t^2 - 4$ from the previous section, is also a sufficient condition for convergence.

In the previous section, we demonstrated that the algorithm for $p_{\max} = 1$ is symplectic. We have also shown this is true for arbitrary $p_{\max}$, and here we summarize the main steps. First write Eq. (41), which holds for $p_{\max} \geqslant 1$, in the vector-matrix form $\delta\boldsymbol{y}^{n+1,p+1} = \mathbf{B}\delta\boldsymbol{y}^{n+1,p} = (\mathbf{B})^p\delta\boldsymbol{y}^{n+1,1}$, where $(\mathbf{B})^p$ denotes the matrix $\mathbf{B}$ raised to the pth power (as opposed to an index superscript). Expressing the converged Crank–Nicolson solution as $\boldsymbol{y}^{n+1} = \mathbf{C}\boldsymbol{y}^n$ (where the matrix $\mathbf{C}$ can be easily calculated), and the $p_{\max} = 1$ solution as $\boldsymbol{y}^{n+1,1} = \mathbf{A}\boldsymbol{y}^n$ (where $\mathbf{A}$ is given by Eq. (36)), this can be written as $\boldsymbol{y}^{n+1,p+1} = [\mathbf{C} + (\mathbf{B})^p(\mathbf{A} - \mathbf{C})]\boldsymbol{y}^n \doteq \mathbf{A}_p\boldsymbol{y}^n$. The eigenvalues of the matrix $\mathbf{A}_p$ can then be calculated using a symbolic mathematics package for simplicity (we used Maple [21] for this calculation). To reduce the complexity of intermediate equations, it is useful to express $(\mathbf{B})^p = \mathbf{E}(\mathbf{D})^p\mathbf{E}^{-1}$, where the columns of $\mathbf{E}$ are the eigenvectors of $\mathbf{B}$ and the diagonal matrix $(\mathbf{D})^p = (0,0;0,(\lambda_2)^p)$ (without yet substituting the actual value of the eigenvalue λ_2 from Eq. (42)). One can then show that the eigenvalues of $\mathbf{A}_p$ can be written in the form $\lambda_{p,\pm} = c_1 \pm \mathrm{i}\sqrt{c_2}$, where c_1 is real and a sufficient condition that c_2 be positive is again the stability criterion from the previous section, $\hat{\omega}^2\Delta t^2 > \omega^2\Delta t^2 - 4$. The terms c_1 and c_2 satisfy $c_1^2 + c_2 = 1$, thus showing that this algorithm is symplectic for an arbitrary number of iterations $p_{\max}$.

2.4. On the relation between the iterative semi-implicit method and a preconditioned Jacobian–Free Newton iterative method

To investigate some of the properties of this algorithm, let us for simplicity consider now a more compact vector form provided by the following equation:

$$\frac{\partial\psi}{\partial t} = \mathcal{F}(\psi) - \mathcal{D}_\eta\psi, \tag{44}$$

where $\psi = \psi_{\mathrm{i}}(\boldsymbol{x}, t)$ may be a vector field (with components $i = 1,\ldots,N$), $\mathcal{F}$ is a non-linear operator and $\mathcal{D}_\eta$ is a linear operator (which represents dissipation in our problems). Introducing the variable transformation $\psi(t) = \mathrm{e}^{-\mathcal{D}_\eta t}\tilde{\psi}(t)$, this becomes

$$\frac{\partial\tilde{\psi}}{\partial t} = \mathrm{e}^{\mathcal{D}_\eta t}\mathcal{F}(\psi) \doteq \widetilde{\mathcal{F}}(\tilde{\psi}). \tag{45}$$

(A definition is denoted by $\doteq$.) For simplicity, we will drop the tildes over variables in the rest of this subsection. It is not hard to see that, in this case, the numerical scheme of Eqs. (29) and (30) (using Eq. (13) to approximate $\mathcal{F}$) is equivalent, for $p > 1$, to:

$$\psi^{n+1,p+1} = \psi^n + \frac{\Delta t}{2}[\mathcal{F}(\psi^n) + \mathcal{F}(\psi^{n+1,p}) + \widehat{\mathcal{F}}(\psi^{n+1,p+1} - \psi^{n+1,p})], \tag{46}$$

where $\widehat{\mathcal{F}}$ is the semi-implicit operator, and therefore, chosen to be invertible. Eq. (46) can thus be easily manipulated to yield:

$$\psi^{n+1,p+1} = \left(1 - \frac{\Delta t}{2}\widehat{\mathcal{F}}\right)^{-1}\left[\psi^n + \frac{\Delta t}{2}\mathcal{F}(\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\psi^{n+1,p}) - \frac{\Delta t}{2}\widehat{\mathcal{F}}(\psi^{n+1,p})\right]. \tag{47}$$

We now show how the above equation can be obtained with a Jacobian–Free Newton algorithm coupled with a preconditioned simple functional iteration. A Crank–Nicolson discretization of Eq. (45) leads to:

$$\psi^{n+1} - \frac{\Delta t}{2}\mathcal{F}(\psi^{n+1}) = \psi^n + \frac{\Delta t}{2}\mathcal{F}(\psi^n). \tag{48}$$

As mentioned before, directly inverting the LHS of this equation is often hard and impractical. A common and useful approach is to resort to preconditioned iterative methods. For non-linear cases like this, it is useful to use an outer Newton iteration for the non-linear part of the problem, and an inner iteration with preconditioning for the resulting linear part of the problem. That is, introducing a Newton iteration count p, the above equation becomes

$$\psi^{n+1,p+1} - \frac{\Delta t}{2}\left[\mathcal{F}(\psi^{n+1,p}) + \frac{\delta\mathcal{F}}{\delta\psi}(\psi^{n+1,p+1} - \psi^{n+1,p})\right] = \psi^n + \frac{\Delta t}{2}\mathcal{F}(\psi^n). \tag{49}$$

Because $\delta\mathcal{F}/\delta\psi$ is a complicated, large sparse matrix, it is still impractical to directly solve this problem. If we have an approximation to $\delta\mathcal{F}/\delta\psi$, denoted by $\widehat{\mathcal{F}}$, that is not too hard to invert, we can use this as a preconditioner, and write Eq. (49) as

$$\left(1 - \frac{\Delta t}{2}\widehat{\mathcal{F}}\right)^{-1}\left(1 - \frac{\Delta t}{2}\frac{\delta\mathcal{F}}{\delta\psi}\right)\psi^{n+1,p+1} = \left(1 - \frac{\Delta t}{2}\widehat{\mathcal{F}}\right)^{-1}\left[\psi^n + \frac{\Delta t}{2}\mathcal{F}(\psi^n) + \frac{\Delta t}{2}\mathcal{F}(\psi^{n+1,p}) - \frac{\Delta t}{2}\frac{\delta\mathcal{F}}{\delta\psi}\psi^{n+1,p}\right]. \tag{50}$$

This is of the form $\widehat{A}^{-1}A\psi^{n+1,p+1} = b^p$. If the preconditioner is sufficiently good, then $\widehat{A}^{-1}A$ is close enough to the identity matrix that a simple functional iteration should be sufficient. Using an index q to label these iterations within each Newton step, this would become $\psi^{n+1,p+1,q+1} = b^p + (1 - \widehat{A}^{-1}A)\psi^{n+1,p+1,q}$, with the initial condition $\psi^{n+1,p+1,0} = \psi^{n+1,p}$. This should converge as long as the magnitude of the eigenvalues of $(1 - \widehat{A}^{-1}A) = \widehat{A}^{-1}(\widehat{A} - A)$ are all less than unity. If we do only one inner functional iteration ($q = 0$) per outer Newton iteration, then this exactly reproduces Eq. (47). [Alternatively, replacing $\delta\mathcal{F}/\delta\psi$ in Eq. (49) with the approximate Jacobian $\widehat{\mathcal{F}}$ leads to Eq. (47).] The iterative semi-implicit algorithm presented in this paper is essentially this, with a particular physics-based form for $\widehat{\mathcal{F}}$ appropriate for high frequency waves arising from coupling between differential equations, and a modification of the first step to give a symplectic second order accurate result after one corrector step.

More advanced iterative methods could be used, such as Krylov methods like GMRES or restarted Loose GMRES [22], and will accelerate convergence for some types of problems. (Also, because of the various approximations made, our simplified algorithm will not have the asymptotic quadratic convergence of a full Newton algorithm on the non-linear part of the problem.) However, we have found that simple functional iteration works quite well for our problems, converging in just an iteration or two. In part, this is because there is very little energy contained in the high frequency waves in these problems. In principle, convergence might be slow if there were a significant component of the error vector in these high frequency waves, for which $\widehat{A}^{-1}(\widehat{A}-A)$ has the largest eigenvalues. But there is very little physical coupling between the low frequency dynamics of interest and the high frequency waves, so if they are treated in a numerically stable way, they will have been damped away on previous timesteps and do not require many iterations on subsequent timesteps.

Although we use a simple functional iteration, note that the optimal method for inverting the physics-based preconditioner $\left(1-\frac{\Delta t}{2}\widehat{\mathcal{F}}\right)$ itself might in some cases be an iterative algorithm (more on this in Section 3.2), or in other cases with FFTs or a direct sparse matrix solver. For our application, we are already using FFTs for a pseudo-spectral evaluation of the non-linear terms, so it is easy to implement and very efficient to directly invert our physics-based preconditioner in wave-number space.

2.5. Second order operator splitting

In the derivation of the numerical scheme of Section 2.1, the non-linear advection operators and the diffusion operators are treated differently. Operator splitting methods can give rise to splitting errors in numerical schemes, resulting in an overall order of the scheme which is lower than that of any of the individual methods applied to treat each of the operators. In this section, we show that our scheme is second order accurate even if these operators do not commute. To demonstrate this point, it is sufficient to consider the paradigm posed by the compact vector form of Eq. (44). Crank–Nicolson discretization of Eq. (45) yields:

$$\tilde{\psi}^{n+1}-\tilde{\psi}^{n}=\frac{\Delta t}{2}\mathrm{e}^{\mathcal{D}_\eta(t+\Delta t)}\mathcal{F}(\psi^{n+1})+\frac{\Delta t}{2}\mathrm{e}^{\mathcal{D}_\eta t}\mathcal{F}(\psi^{n}). \tag{51}$$

Undoing the variable transformation and rearranging gives

$$\psi^{n+1}-\frac{\Delta t}{2}\mathcal{F}(\psi^{n+1})=\mathrm{e}^{-\mathcal{D}_\eta\Delta t}\left(\psi^{n}+\frac{\Delta t}{2}\mathcal{F}(\psi^{n})\right). \tag{52}$$

Through second order accuracy, the LHS can be expanded as $\mathrm{LHS}=\psi^{n+1}-(\Delta t/2)[\mathcal{F}(\psi^n)+(\partial\mathcal{F}/\partial\psi)(\psi^{n+1}-\psi^n)]$. Then the above equation becomes

$$\left(1-\frac{\Delta t}{2}\frac{\partial\mathcal{F}}{\partial\psi}\right)\psi^{n+1}=\mathrm{e}^{-\mathcal{D}_\eta\Delta t}\left(\psi^{n}+\frac{\Delta t}{2}\mathcal{F}(\psi^{n})\right)+\frac{\Delta t}{2}\mathcal{F}(\psi^{n})-\frac{\Delta t}{2}\frac{\partial\mathcal{F}}{\partial\psi}\psi^{n}. \tag{53}$$

From a second order Taylor-series expansion, $\psi(t+\Delta t)=\psi(t)+\Delta t\psi'+\frac{1}{2}\Delta t^2\psi''$, we see that the solution to Eq. (44) through second order accuracy should be:

$$\psi^{n+1}=\psi^{n}+\Delta t\mathcal{F}(\psi^{n})-\Delta t\mathcal{D}_\eta\psi^{n}+\frac{\Delta t^2}{2}\left[\frac{\partial\mathcal{F}}{\partial\psi}\mathcal{F}(\psi^{n})-\frac{\partial\mathcal{F}}{\partial\psi}\mathcal{D}_\eta\psi^{n}-\mathcal{D}_\eta F(\psi^{n})+\mathcal{D}_\eta^2\psi^{n}\right]. \tag{54}$$

Inverting the operator on the LHS of Eq. (53) and expanding, one can show that Eq. (53) agrees with Eq. (54) through second order. Thus we have demonstrated that, at least in the limit of $\Delta t\,\partial\mathcal{F}/\partial\psi\ll 1$, our splitting scheme is second order accurate even if the advection ($\mathcal{F}$) and the diffusion ($\mathcal{D}_\eta$) operators do not commute. Numerical demonstration of the second order convergence of the algorithm for our test problem will be presented in Section 4, where it will be seen that the algorithm remains second order accurate even at values of Δt considerably larger than those required by an explicit integration.

2.6. Robust damping

Let us consider a simple test problem based on a linear scalar version of Eq. (44) with $\mathcal{F} = -\mathrm{i}\omega$ and $\mathcal{D}_\eta$ a real number,

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = -(\mathrm{i}\omega + \mathcal{D}_\eta)\psi. \tag{55}$$

The analytic solution is straightforward:

$$\psi = \psi_0 \mathrm{e}^{-(\mathrm{i}\omega+\mathcal{D}_\eta)t}, \tag{56}$$

where ψ_0 is the initial condition. A plot of the magnitude of this solution is shown in Fig. 1, for the parameters $\omega = 10$, $\psi_0 = 1$ and $\mathcal{D}_\eta = 1$ (full black line). Also shown are the numerical solutions of Eq. (55) with three different numerical methods: in red (dashed line), the solution yielded by Eq. (52) (i.e. the method described in this paper) $\psi^{n+1} = (1 + \mathrm{i}\omega\Delta t/2)^{-1}\mathrm{e}^{-\mathcal{D}_\eta\Delta t}(1 - \mathrm{i}\omega\Delta t/2)\psi^n$; in green (dotted line) the result obtained by the analytic integration of the diffusive exponential, as described in Appendix A and, in blue (dash-dotted line), the solution given by a Crank–Nicolson discretization of the RHS. As seen, for $\omega\Delta t < 1$, all three methods reproduce the exact solution. However, only the discretization of Eq. (52) is capable of capturing the behavior of the exact solution for *any* value of Δt, as is desirable for a robust treatment of damping with implicit integration, where $\mathcal{D}_\eta\Delta t \gg 1$. One might try to improve the CN approach by using CN for the wave term and backwards Euler for the damping term, i.e. discretize Eq. (55) as $(\psi^{n+1} - \psi^n)/\Delta t = -\mathrm{i}\omega(\psi^{n+1} + \psi^n)/2 - \mathcal{D}_\eta\psi^{n+1}$. However, in the limit $\Delta t \to \infty$ this gives $\psi^{n+1}/\psi^n = -\omega/(\omega - \mathrm{i}2\mathcal{D}_\eta)$, which for $\mathcal{D}_\eta \ll \omega$ gives very little damping, even though $\mathcal{D}_\eta\Delta t \gg 1$.

The algorithm presented here preserves the property of robust damping, which mathematically is termed L-stability. An algorithm that is stable for the Eq. (55) for arbitrary Δt (for $\mathcal{D}_\eta \geqslant 0$) is called A-stable. An algorithm is L-stable if it is A-stable and has the property of the exact solution that $\psi^{n+1}/\psi^n \to 0$ as $\Delta t \to \infty$. It is well-known that Crank–Nicolson is A-stable but not L-stable. In physical terms, while Crank–Nicolson has the nice property of giving no artificial damping of pure wave problems (i.e. it is symplectic for $\mathcal{D}_\eta = 0$), it has a difficulty in not giving significant damping for problems that should be strongly damped in the large Δt limit. Examples of L-stable algorithms include the second order backward differentiation formula (BDF2), and certain Rosenbrock or implicit–explicit Runge–Kutta algorithms. The iterated semi-implicit method presented here is based on a kind of integrating-factor variation of Crank–Nicolson that makes it L-stable. (An algorithm based on the modified Crank–Nicolson algorithm [23,24] could be another way to improve the treatment of damping.) Alternatively, one could develop an iterative semi-implicit algorithm starting from BDF2 instead of CN. This would be automatically L-stable and would treat the damping and other terms in a balanced way that might have certain advantages for some applications. On the other hand, BDF2 introduces a weak numerical damping of resolved waves, $|\psi^{n+1}/\psi^n| \approx (1 - (\omega\Delta t)^4/4 + \mathcal{O}(\Delta t^6))$, while the algorithm presented here has the advantage of retaining the CN property of being

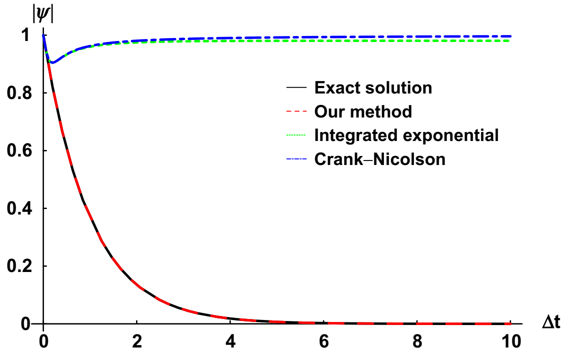

Fig. 1. Plot of the solution of Eq. (55) as obtained analytically, expression (56), and with three different numerical methods. Input parameters are $\omega = 10$, $\mathcal{D}_\eta = 1$ and $\psi_0 = 1$.

dissipation free in the limit $\mathcal{D}_\eta = 0$. Future work could investigate the relative advantages of these different approaches.

3. Application – gyrofluid magnetic reconnection

The field of plasma physics is particularly rich in stiff wave problems. One such application is magnetic reconnection [25,26], a phenomenon in which magnetic field lines that are being convected with the plasma flows suddenly break and reconnect in a different configuration. Magnetic energy is released in this process, giving rise to high velocity plasma flows, energetic particles and plasma heating. Reconnection is the cause of solar flares [27,28]; it is also manifest in the interaction between the solar wind and the Earth's magnetic field [29] in the magnetopause [30] and magnetotail [31]. In fusion devices it plays a crucial role in the development of the sawtooth instability, which can be detrimental to the plasma confinement [32].

In the absence of external forcing, magnetic reconnection arises as a result of a well-known plasma instability called the tearing mode [33]. Regardless of the physical framework [i.e. resistive magnetohydrodynamics (MHD), Hall-MHD,[1] kinetic, etc.], this instability is particularly hard to simulate numerically (see, e.g. [2,11]). The reason is essentially one: the intrinsically multiscale character, both temporal and spatial, of magnetic reconnection. With respect to its time evolution, the problem arises because the typical growth rate γ of the tearing instability is much smaller than the frequency of the fastest waves present in the system (e.g. Alfvén waves, Kinetic Alfvén waves, whistler, etc.); spatially, this problem requires the simulation of at least two, but often more, spatially distinct scales: the large scale equilibrium and the short scale dissipation layer. The separation between these two scales is a function of the magnitude of the field-line breaking terms (e.g. resistivity, electron inertia) which, in order to address a physically meaningful region of parameter space, have to have extremely small magnitudes, and thus become important only in very narrow regions in the plasma where strong gradients arise. It follows from these conditions that numerical simulations of magnetic reconnection require extremely high spatial resolutions and either an implicit time integration scheme or, if an explicit scheme is adopted, timesteps which are much smaller than $1/\gamma$.

In many plasmas of interest, such as those in fusion devices, or at the magnetotail or solar flares, it has long been suspected that an MHD description is too simple to fully explain the complexity of the observations. The discrepancy is obvious in the observed and calculated reconnection rates which, in the MHD framework, differ by several orders of magnitude. Interest has diverged to non-MHD effects which might cause a speed-up of this process [34–36]. The potential candidates differ depending on the specific geometry and plasma parameters. For example, in most present fusion plasmas, the ion Larmor radius exceeds, or at least is comparable to, the width of the dissipation region. Thus, finite ion Larmor radius (FLR) effects cannot be neglected and are, in fact, known to produce a speed-up of the reconnection rate [34,37]. From the numerical point of view, the FLR terms bring about one further complication: their inclusion in the equations fundamentally changes the dispersion relation of the shear Alfvén wave, essentially generalizing it to the so-called Kinetic Alfvén wave (KAW), which has a high frequency dispersive character, i.e. $\omega \sim k_\perp^2$, where ω is the wave frequency and $k_\perp$ the wavenumber (the counterpart to the FLR effects in the absence of a strong magnetic field is the Hall term in Ohm's law, which also introduces a high frequency dispersive wave, the whistler – see [38] for a detailed discussion of the relative roles of these waves). Explicit numerical integration schemes are forced to resolve the highest frequencies present in the system, i.e. they must integrate on a timestep $\Delta t \sim 1/\omega_{\max}$. Therefore, when dispersive waves are present, $\Delta t \sim 1/k_{\perp,\max}^2$. Since $k_{\perp,\max} \sim 1/\Delta x$, where Δx is the grid-spacing, the timestep becomes impractically small as higher resolutions are required to resolve the extremely thin reconnection layer. Reconnection studies with FLR or Hall terms typically use hyper-diffusion (or hyper-resistivity) to limit how thin the reconnection layer becomes, which helps limit how small the grid spacing Δx needs to be, and thus allows the timestep Δt to be larger than it would be otherwise. There can be a certain amount of physical hyper-resistivity in the Ohm's law due to electron viscosity, and the macroscopic physical results are found to be independent of the choice of hyper-diffusion over a range of small hyper-diffusion, but if the hyper-diffusion is too large it can affect some of the physical results of interest. For typical values of hyper-

[1] Hall-MHD consists in extending the usual MHD Ohm's law to include the so-called Hall term, $\boldsymbol{J} \times \boldsymbol{B}$.

diffusion used in simulations, the required grid resolution is still fairly fine and the resulting explicit timestep limitation is still quite severe, so one is motivated to search for an implicit or semi-implicit algorithm to bypass this severe timestep limitation.

3.1. Equations

To study FLR effects in magnetic reconnection we chose a two-field gyrofluid model, described by the following normalized equations:

$$\frac{\partial n_\mathrm{e}}{\partial t}+[\phi,n_\mathrm{e}]=[\psi,\nabla_\perp^2\psi]+\nu\nabla_\perp^2 n_\mathrm{e}-\nu_\mathrm{H}\nabla_\perp^4 n_\mathrm{e}, \tag{57}$$

$$\frac{\partial\psi}{\partial t}+[\phi,\psi]=\rho_\mathrm{s}^2[n_\mathrm{e},\psi]+\eta\nabla_\perp^2(\psi-\psi_\mathrm{eq})-\eta_\mathrm{H}\nabla_\perp^4(\psi-\psi_\mathrm{eq}), \tag{58}$$

$$n_\mathrm{e}=\frac{1}{\rho_\mathrm{i}^2}[\widehat{\Gamma}_0(b)-1]\phi. \tag{59}$$

Here n_e is the perturbed electron density, ϕ represents the electrostatic potential, ψ is the magnetic flux, related to the in-plane magnetic field by $\boldsymbol{B}_\perp=\boldsymbol{e}_z\times\nabla\psi$, $[P,Q]=\partial_xP\partial_yQ-\partial_yP\partial_xQ$ is the Poisson bracket and ρ_i and ρ_s are the ion and the ion-sound Larmor radius, respectively. Auxiliary quantities are the parallel current density, $j_\parallel=\nabla_\perp^2\psi$, where $\nabla_\perp^2=\partial_x^2+\partial_y^2$, and the flow velocity, $\boldsymbol{v}_\perp=\boldsymbol{e}_z\times\nabla\phi$. Dissipation is provided by the viscosity ν and the resistivity η. We also add, for physical reasons and numerical convenience, phenomenological hyper-diffusion coefficients, ν_H and η_H [39,40]. In Eq. (58), the equilibrium is subtracted from the non-ideal terms on the RHS in order to impose an ideal equilibrium [this is equivalent to imposing an external electric field, $E_\mathrm{ext}=\eta\nabla_\perp^2-\eta_\mathrm{H}\nabla_\perp^4\psi_\mathrm{eq}]$ – see Appendix B for how to include this term in the numerical scheme of Eqs. (27)–(30). Time is normalized to the Alfvén time, lengths to the equilibrium scale length.

These equations are reminiscent of the more complete models of Snyder and Hammett [41] and Schep et al. [42]. Eq. (59) is known in the literature as the gyrokinetic Poisson equation [43]. The integral operator $\widehat{\Gamma}_0$ expresses the average of the electrostatic potential over rings of radius ρ_i. In Fourier-space, this operator simply becomes

$$\Gamma_0(b)=\mathrm{e}^{-b}I_0(b), \tag{60}$$

where $b=k_\perp^2\rho_\mathrm{i}^2$ and I_0 is the modified Bessel function of zeroth order. In other words, within the gyrokinetic work frame, the particle is allowed to experience different values of the electrostatic potential as it orbits the magnetic guide field. The purpose of the Γ_0 operator is to average ϕ over such orbits and over the Maxwellian velocity distribution. When implemented numerically in real-space, a Padé approximant of this operator is often used [44]:

$$\Gamma_0(b)-1\approx-\frac{b}{1+b}, \tag{61}$$

which converts Eq. (59) into:

$$(1-\rho_\mathrm{i}^2\nabla_\perp^2)n_\mathrm{e}=\nabla_\perp^2\phi. \tag{62}$$

In our case, Eq. (59) is solved in Fourier-space and thus no approximation to $\Gamma_0(b)$ is required. In the limit $k_\perp\rho_\mathrm{i}\to 0$, the above set of equations reduces simply to the reduced MHD (RMHD) model of Strauss [45]. Sets of fluid equations similar to the above, or their collisionless version, which further include electron inertia, have been in use for quite some time to study magnetic reconnection (e.g. [37]). The particular form presented, however, differs from most in that it also includes the effect of finite ion temperature (i.e. $\rho_\mathrm{i}\neq 0$) and is unique, so far, in magnetic reconnection studies, in keeping the full form of the gyrokinetic Poisson law, Eq. (59), instead of its Padé approximant, Eq. (62) (see, e.g. [46]). For simplicity, let us for now ignore the diffusive and hyper-diffusive terms and consider an equilibrium described by $\boldsymbol{B}_{\perp,\mathrm{eq}}=B_0\boldsymbol{e}_y$, with B_0 a constant, $n_{\mathrm{e,eq}}=0$ and $\phi_\mathrm{eq}=0$. We want to linearise Eqs. (57)–(59) assuming that all the fields can be written as $\chi=\chi_\mathrm{eq}+\chi^1(x,y)$, where χ^1 represents small perturbations to the equilibrium of the form $\chi^1(x,y)=\chi_1(x)\mathrm{e}^{\mathrm{i}k_yy}$. Dropping the subscripts, Eqs. (57) and (58) become:

$$\frac{\partial n_e}{\partial t} = f\psi, \tag{63}$$

$$\frac{\partial \psi}{\partial t} = g n_e, \tag{64}$$

where

$$f = -\mathrm{i}k_y B_0 k_\perp^2, \tag{65}$$

$$g = -\mathrm{i}k_y B_0 \left(\rho_s^2 - \frac{\rho_i^2}{\Gamma_0(b) - 1}\right). \tag{66}$$

Combining these two equations, we have:

$$\frac{\partial^2 \psi}{\partial t^2} = -\omega^2 \psi, \tag{67}$$

and similarly for n_e, where ω^2 is given by

$$\omega^2 = -fg = k_\perp^2 \left(\rho_s^2 - \frac{\rho_i^2}{\Gamma_0(b) - 1}\right) k_y^2 B_0^2. \tag{68}$$

This is the general dispersion relation for the Kinetic Alfvén wave (KAW). In the limit of $k_\perp \rho_i \ll 1$ this relation reduces to:

$$\omega^2 = \left[1 + k_\perp^2 \rho_i^2 \left(\frac{3}{4} + \frac{1}{\tau}\right)\right] k_y^2 B_0^2, \tag{69}$$

where $\tau = T_i/T_e$. The first term on the RHS constitutes what is usually referred to as the shear Alfvén wave, and the remaining $k_\perp^2 \rho_i^2$ terms are the FLR corrections. [Note that $|\mathbf{k} \cdot \mathbf{B}_{eq}|^2 = k_y^2 B_{eq,y}^2 = k_y^2 B_0^2$ because $k_z = 0$ in this 2D problem and the guide field $B_{eq,z}$ only enters through the gyroradii ρ_i and ρ_s.] The opposite limit of $k_\perp \rho_i \gg 1$ gives:

$$\omega^2 = k_\perp^2 \rho_\tau^2 k_y^2 B_0^2, \tag{70}$$

where $\rho_\tau^2 = \rho_s^2 + \rho_i^2$. The numerical difficulties faced in the explicit integration of the set of Eqs. (57)–(59) at $k_\perp \rho_i > 1$ are obvious from expression (70), where we basically find that $\omega \propto k_\perp k_y \propto k^2$.

3.2. Semi-implicit operator

Based on the linear dispersion relation, Eq. (68), a plausible first choice for the SI operator $\hat{\omega}^2$ could be:

$$\hat{\omega}^2 = k_\perp^2 \left(\rho_s^2 - \frac{\rho_i^2}{\Gamma_0(b) - 1}\right) k_y^2 a_0^2 B_0^2, \tag{71}$$

where a_0 is a dimensionless constant, set to a value that ensures the stability of the algorithm and, to account for a spatially varying equilibrium magnetic field, $B_{eq} = B_{eq}(x)$, B_0 can be chosen to be its maximum value. However, in the non-linear regime, the magnetic field will no longer be a function of x alone, but rather an arbitrary function of x and y. This suggests a qualitative generalization of this expression in order to also apply to the non-linear regime:

$$\hat{\omega}^2 = k_\perp^4 \left(\rho_s^2 - \frac{\rho_i^2}{\Gamma_0(b) - 1}\right) a_0^2 B_{\perp,\max}^2, \tag{72}$$

where we simply have made the transformations $k_y \to k_\perp$ and $B_0 \to B_{\perp,\max}$, where we define $B_{\perp,\max} = \max\left(\sqrt{B_x^2 + B_y^2}\right)$ is the maximum magnitude of total in-plane magnetic field, updated at each new timestep.

Alternatively, one could try developing an SI operator that included the spatial dependence of the magnetic field, but it would no longer be analytically invertible in k, and would require alternative solvers that are not available in our code at present. (For example, some rational function approximations for an SI operator that included spatial variation of B_0 might be efficiently invertible in real-space using a multigrid solver. See Ref. [11] for some other possible approaches to preconditioners/SI operators for problems with strong inhomogeneities, anisotropies and multiple waves.) Of course, for problems in which the magnetic field has spatial variation, as in the application of the next section, Eq. (68) is no longer an accurate dispersion relation for the system (i.e. it is no longer an eigenvalue of the linear problem, and Fourier modes are no long eigenvectors). Nevertheless, as we shall see in Section 4, the SI operator based on a constant B_0 is found to be fairly effective for the case considered there. (In part this may be because stability of the SI algorithm only needs the approximation $\hat{\omega}^2$ to be sufficiently large. While an SI operator that is larger than necessary may require more iterations to converge from an arbitrary initial guess, this can be offset if the initial predictor–corrector step is sufficiently accurate.) In places where $k_\perp\rho_{\rm i} \gg 1$, this operator simply reduces to the spectral equivalent of the real-space $\nabla_\perp^4$ operator proposed by Harned and Mikic [7] for the whistler wave. Note, however, that whereas $k_\perp\rho_{\rm i}$ will certainly be very large in the vicinity of the X-point, it will also be negligibly small far away. It is therefore very useful to retain the full form of the Γ_0 operator in the expression for $\hat{\omega}$. Because our code works in Fourier-space, this operator is trivial to implement without resorting to any approximations. Since it does not require any convolution, the computational cost of evaluating expression (72) at each timestep is negligible. In the conclusions, we address the issue of how it might be generalized to a real-space grid implementation.

An important property of this SI operator is its ability to act on each value of $k_\perp$ individually. For instance, the magnitude of $\hat{\omega}$ is small for the lower values of $k_\perp$, meaning that the large scale features are left relatively unaffected. Also, it is well behaved in the limit of $k_\perp\rho_{\rm i} \ll 1$, where it reduces to:

$$\hat{\omega} = \left[1 + k_\perp^2\rho_{\rm i}^2\left(\frac{3}{4}+\frac{1}{\tau}\right)\right]k_\perp^2 a_0^2 B_{\perp,\max}^2, \tag{73}$$

a form which makes transparent the operator's ability to equally stabilize the shear Alfvén wave.

Finally, we note that the substitution $k_y \to k_\perp$ in the expression for the SI operator implies that during the linear and early non-linear regimes the parameter a_0 can be set to values which are smaller than 1. This is important since, as mentioned, we want the operator $\hat{\omega}$ to be sufficiently large to satisfy the stability condition $\hat{\omega}^2 > \omega^2$ (and ideally, $\hat{\omega}^2$ as close as possible to ω^2 for all wavenumbers), but not so large that convergence is slow and the error defined by expression (32) (which is proportional to a_0^2) is larger than it need be.

3.3. Semi-implicit error control and timestep determination loop

An important part of the algorithm is the semi-implicit error control and timestep determination loop. The basic idea behind it is to enforce the limit on the semi-implicit error prescribed by the user,

$$\max\left|\mathcal{E}_{ji}^p\right| \leqslant \mathcal{E}^{\max}, \tag{74}$$

while maximizing the timestep that can be taken. This is done the following way. The discretized set of Eqs. (27)–(30) is iterated until $p = p_{\max}$. Then, the following conditions determine the next step:

$$\begin{aligned}
&(\mathrm{a})\quad \mathcal{E} < 0.8\mathcal{E}^{\max}: \quad \Delta t^{n+1} = C^{+}\Delta t^{n},\\
&(\mathrm{b})\quad 0.8 < \mathcal{E} < \mathcal{E}^{\max}: \quad \Delta t^{n+1} = \Delta t^{n},\\
&(\mathrm{c})\quad \mathcal{E} > \mathcal{E}^{\max}: \quad \text{loop is repeated with } \Delta t^{n+1} = C^{-}\Delta t^{n}.
\end{aligned} \tag{75}$$

where $C^{+} > 1$ and $C^{-} < 1$ are constants. The goal is to keep the error below but as close to the maximum as possible. Based on the second order convergence of the numerical scheme, we set $C^{+} = 1.08$ and $C^{-} = 0.92$. At every timestep, the CFL condition for the plasma flow velocities is also evaluated, according to:

$$\Delta t_{\rm CFL}^{\rm flows} = 0.1\min\left[\frac{\Delta x}{v_{x,\max}}, \frac{\Delta y}{v_{y,\max}}\right]. \tag{76}$$

The next timestep is the minimum of expressions (75) and (76). It is important to note that the error control and timestep determination loop just described makes the choice of a_0 less vital to the algorithm than otherwise. The reason is that if the calculation is started with values of a_0 and Δt such that the SI scheme is unstable, this will cause the error to diverge until it reaches the maximum allowed, at which point Δt is decreased. This cycle will be repeated until a sufficiently small Δt is found for which the SI scheme is stable at that (fixed) value of a_0. Of course, if a_0 is set to too small a value, stability of the SI algorithm might only be available at values of Δt much smaller than those which can be achieved at larger a_0. Finally, the reader is again reminded that this error is not the total error of the integration, but simply that introduced by the semi-implicit operator. When the iterative procedure has converged, there is still an underlying error which is that of the Crank–Nicolson scheme.

4. Numerical tests

As mentioned, we have written a fully parallel pseudo-spectral code to evolve Eqs. (57)–(59). An explicit time-stepping version of this code has been in use for some time to study the evolution of the resistive tearing mode in the single fluid MHD limit ($\rho_\mathrm{i}, \rho_\mathrm{s} = 0$). The code uses a Fourier basis set on a two-dimensional domain with periodic boundary conditions. Details and extensive benchmarking can be found in [50]; see also [51] for recent results. The explicit algorithm uses a modified version of the Adams–Bashford third order method that allows for variable timesteps [52], combined with a Crank–Nicolson scheme for the dissipative terms (in this case, the use of a CN scheme for these terms is fully justified because $\eta k_\perp^2 \Delta t \sim \eta \ll 1$, since $\Delta t \sim 1/\omega_\mathrm{KAW} \sim 1/k_\perp^2$). The timestep is determined by the CFL condition, which is evaluated at each timestep according to the formula:

$$\Delta t = 0.1 \min\left[\frac{\Delta x}{v_{x,\max}}, \frac{\Delta y}{v_{y,\max}}, \frac{\Delta x}{B_{x,\max}}, \frac{\Delta y}{B_{y,\max}}, \frac{2}{\omega_{\mathrm{KAW},\max}}\right], \tag{77}$$

where $\omega_{\mathrm{KAW},\max}$ is an estimate of the maximum frequency of the KAW on the grid, defined as

$$\omega_{\mathrm{KAW},\max} = k_{\perp,\max}\left(\rho_\mathrm{s}^2 - \frac{\rho_\mathrm{i}^2}{\Gamma_0(b) - 1}\right)^{1/2} k_{y,\max} B_{\perp,\max}. \tag{78}$$

The code also allows the resolution in the y-direction (i.e. perpendicular to the equilibrium direction) to vary throughout a run, as needed. When more resolution becomes necessary in this direction the run is stopped and the number of grid points doubled. The new data values necessary for this procedure are obtained via linear interpolation between two adjacent points. This is a very useful feature for reconnection runs, as the needs for resolution in the y-direction increase greatly from the linear to the non-linear regime.

The base-case chosen for comparison between the SI method of Eqs. (27)–(30) and the explicit code is characterized as follows. The equilibrium is defined by $\psi^{(0)} = \psi_0/\cosh(x)^2$ and $\phi^{(0)} = 0$. We set $\psi_0 = 3\sqrt{3}/4$ so that the maximum value of the equilibrium magnetic field is 1. Other parameters are the resistivity and the viscosity coefficients, set to $\eta = \nu = 5 \times 10^{-4}$, and the instability parameter $\Delta' = 17.3$ (i.e. the simulation box size is $L_x = 2\pi$, $L_y = 2.18\pi$). The ion and the ion-sound Larmor radius, ρ_i and ρ_s, are set to 0.02. The hyper-resistivity η_H and hyper-viscosity ν_H are grid dependent, and their value is calculated at each timestep, according to the formula:

$$\eta_\mathrm{H} = \nu_\mathrm{H} = 0.1\omega_{\mathrm{KAW},\max}/k_{\perp,\max}^4. \tag{79}$$

Tests have been performed to insure that these coefficients are sufficiently small not to alter the physics of the system. The tearing mode instability is initialized by perturbing this equilibrium with $\psi^{(1)} = -10^{-5}\cos(2\pi/L_y y)$. The initial resolution is 3072×128 grid points; the number of grid points the y-direction is doubled throughout the run, up to 2048 for $t > \approx 210$. The most crucial test is to check how closely the semi-implicit method described in this paper reproduces the results obtained by a conventional explicit integration. This comparison is performed in Figs. 2 and 3. Fig. 2 shows the growth rate, defined as $\gamma = \mathrm{d}\log\psi_X/\mathrm{d}t$, where X is the location of the X-point, obtained by the two methods. The agreement between both approaches is remarkable. After an initial transient ($t < \sim 30$), the tearing instability is seen here to evolve through four different stages: the linear stage (constant growth rate), the early non-linear period, from $t \approx 130$–170, followed by the explosive stage, during which the growth rate dramatically increases ($t \approx 170$–220) and finally the growth rate slow down and saturation period.

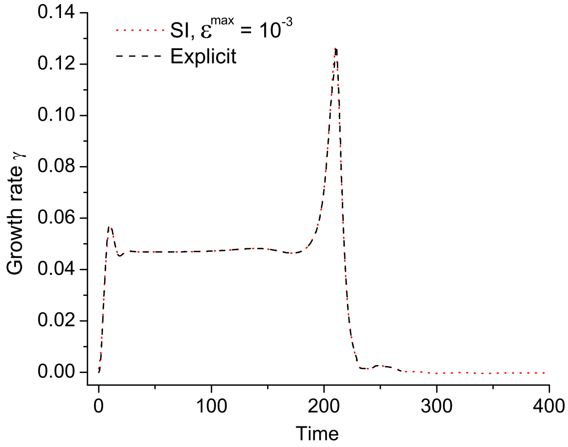

Fig. 2. Comparison between the explicit and SI results. The figure shows the growth rate of the tearing instability, $\gamma = \mathrm{d}\log\psi_X/\mathrm{d}t$, obtained by the explicit method (dashed black line) and the SI method with $\mathcal{E}^{\max} = 10^{-3}$ (dotted red line). The two curves show excellent agreement and visually overlap exactly. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A comparison between the contour plots for all the fields at the early stages of the explosive period ($t = 204$) is shown in Fig. 3. Again, the figure yields excellent agreement between the explicit and the semi-implicit approaches.

Plotted in Fig. 4a is the CPU speed-up factor, determined as follows:

$$\frac{\mathrm{CPU}}{\mathrm{CPU}_{\mathrm{exp}}} = \frac{\Delta t}{(N_{\mathrm{RHS}}/2)\Delta t_{\mathrm{exp}}}, \tag{80}$$

where N_{RHS} is the number of evaluations of the RHS required at each timestep for the SI scheme to converge, compared to an explicit scheme that requires only two such evaluations per timestep. $\mathrm{CPU}_{\mathrm{exp}}$ and Δt_{exp} are the explicit integration CPU time and timestep, respectively. As mentioned, convergence is determined by $\mathcal{E} \leqslant 10^{-3}$ at each timestep. A plot of the time evolution of the maximum error is shown in Fig. 4b. The dashed lines in this figure identify the "comfort zone" for the SI error, defined by expression (75). We see that for both values of $p_{\max}$ plotted, the error is indeed largely confined to that region, indicating the second order convergence of the iterative scheme. The large undershoots mark the points at which the run was restarted with a larger resolution in the y-direction. At each restart, the Δt is set to be equal to that determined by the CFL condition. As shown in Fig. 4a, the SI method is remarkably successful, yielding a CPU speed-up factor ranging from a minimum of ~20 during the explosive growth period, to several hundreds during non-linear saturation of the instability. We also observe that, in this case, the iterative scheme is very efficient, as is shown by the larger speed-up factors allowed for $p_{\max} = 2$ (red dashed line). The actual timesteps taken by the SI method are shown in Fig. 5. We see that the SI method performs so well that the minimum CPU enhancement (~20 at $t \approx 210$) is set by the flow speed, i.e. the actual dynamics that we are interested in following (recall that the CFL condition for the flows is imposed as a maximum for the timestep).

4.1. Numerical test of the second order accuracy of the algorithm

We discussed the second order nature of the algorithm in Section 2, and in Section 2.5, we provided an analytical demonstration that the algorithm remains second order accurate for sufficiently small Δt even if the diffusion and advection operators do not commute. These arguments are rigorously valid only if $\Delta t\,\partial\mathcal{F}/\partial\psi \ll 1$. However, the goal of implicit and semi-implicit algorithms is to allow usage of very large timesteps compared to explicit algorithms. Thus, although the dominant modes of interest have eigenfrequencies ω of $\mathrm{i}\partial\mathcal{F}/\mathrm{d}\psi$ such that $\Delta t\omega$ is presumably small, there will be some high frequency modes for which the effective $\Delta t\,\partial\mathcal{F}/\partial\psi$ is large. To demonstrate that the scheme can indeed be second order accurate (at least for our test

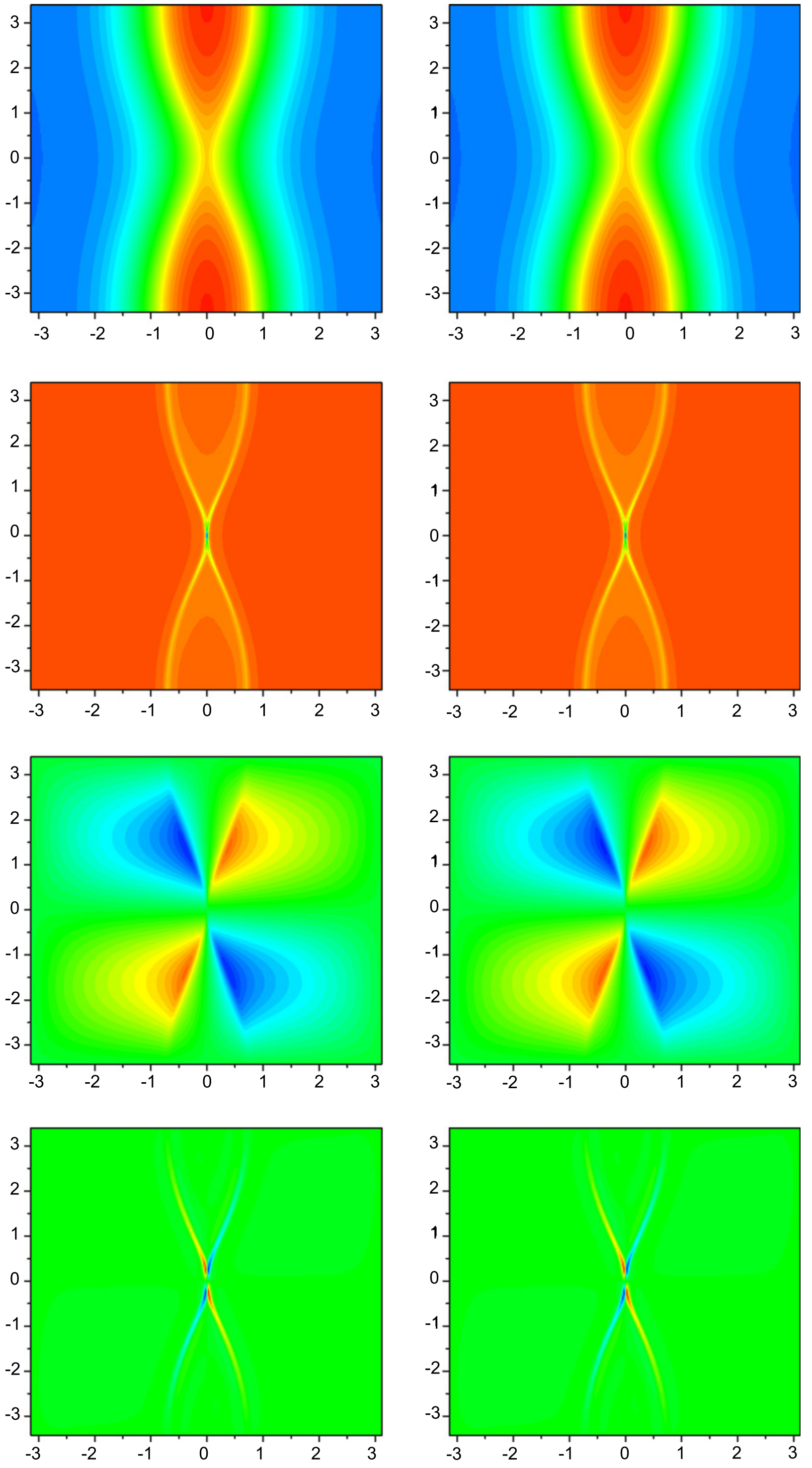

Fig. 3. Contour plots of the fields measured at an island width $W \approx 1.6$ ($t \approx 204$). Left column show results obtained with the explicit integration scheme, right column those obtained with the semi-implicit scheme. From top to bottom, shown are the contours of ψ, $j_{\|}$, ϕ and n_e.

case) at desirable implicit values of Δt (i.e. which greatly exceed those required by an explicit integration), we performed a scaling study of the error of the algorithm with the timestep Δt, using as a test case the application

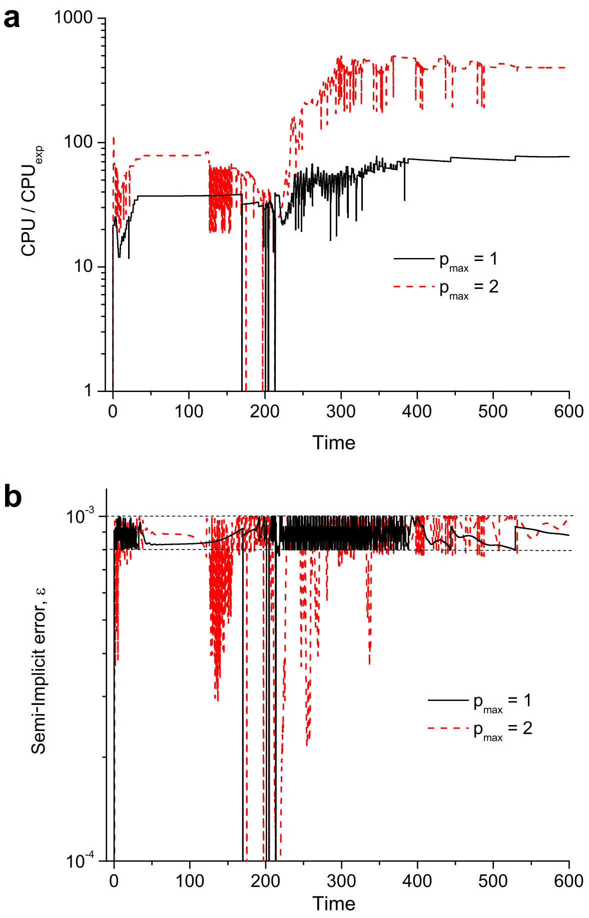

Fig. 4. (a) The CPU speed-up factor over a conventional explicit integration for $p_{\max} = 1, 2$. The timestep taken by the SI scheme is determined by imposing $\mathcal{E} < 10^{-3}$, as seen in (b). The dashed horizontal lines limit the error "comfort zone", defined by expression (75). The largest undershoots in both figures (at times in the vicinity of $t \approx 200$) are due to code restarts at those times, when the resolution in the y-direction is doubled.

discussed in the preceding section. This study was done as follows: starting at time $t \approx 210$, when the simulation is fully non-linear (see Fig. 2) and the SI algorithm is using a timestep of $\Delta t = 9 \times 10^{-4}$, we run the code for an additional time of 0.135 using a very small time step of $\Delta t = 10^{-6}$. This value is smaller than even the explicit timestep $\Delta t \sim 1.5 \times 10^{-5}$ that would be required at this stage, and we take the result of this integration as our exact result. Then, we perform several runs over this same time period of 0.135 with various fixed values of the timestep Δt. To isolate the scaling of the error with Δt, this scaling study was performed with a fixed number of iterations per timestep, $p_{\max} = 2$. The error is the difference between the results of these integrations and the "exact" result (we have chosen to plot the relative error in ψ at the X-point). The result is shown in Fig. 6, where a clear scaling of the error with $(\Delta t)^2$ is seen. Note that, at each value of the iteration index p, we have $\psi^{n,p} = \psi_{\text{true}} + \mathcal{E}_{\text{SI}}^{n,p} + \mathcal{E}_{\text{discr.}}^{n}$ (and similarly for ϕ), where ψ_{true} is the true solution (here taken to be that given by the integration with $\Delta t = 10^{-6}$), $\mathcal{E}_{\text{SI}}$ is the error introduced by the SI operator, and $\mathcal{E}_{\text{discr.}}$ is the error due to the particular discretization scheme used (e.g. when our scheme is converged in the p-iterations, $\mathcal{E}_{\text{discr.}}$ is simply the error of the CN method). Thus, what is plotted in the figure is the total error, due to both the SI error and the time discretization error. In the figure, the vertical dotted line identifies the timestep that would have been required for stability by an explicit algorithm at this point of the simulation. The dashed line shows a Δt^2 slope. We thus see that the second order accuracy of the algorithm extends at least up to the typical SI values of $\Delta t \sim 9 \times 10^{-4}$, which is 60 times larger than the explicit timestep limit.

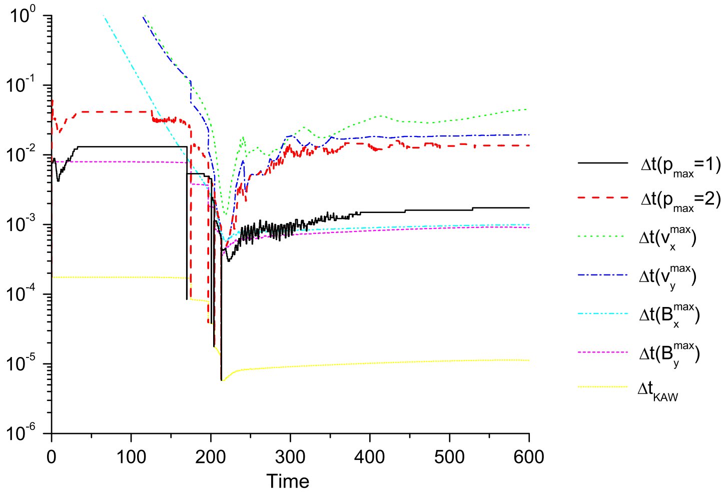

Fig. 5. Plot of Δt vs. time achieved by the numerical scheme of Eqs. (27)–(30)for the test case and for $p_{\max} = 1$ (black full line) and $p_{\max} = 2$ (red dashed line). Also plotted for comparison are the timesteps due to each term contributing to the CFL condition (77). Note that the timestep taken by the semi-implicit method coincides with that based on the flow velocity during the explosive part of the instability.

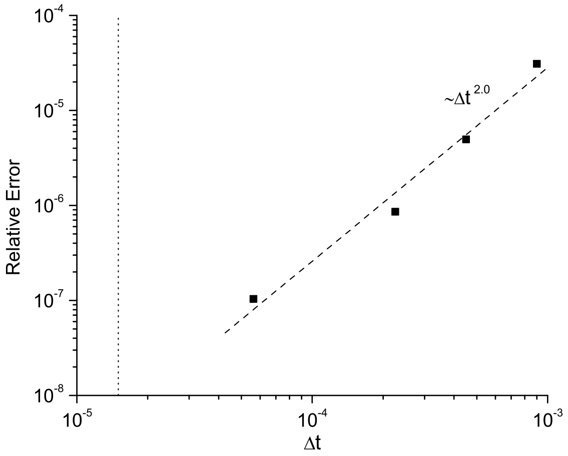

Fig. 6. Scaling of the relative error with Δt. Dashed line indicates $(\Delta t)^2$ slope. Vertical dotted line marks the explicit timestep limit that would be required at this point in the simulation.

5. Conclusions

We have presented an efficient and accurate numerical method for the time integration of stiff wave coupled partial differential equations. The method allows for timesteps which greatly exceed those imposed by the CFL constraint, while accurately reproducing explicit calculations. CPU time savings over a conventional explicit scheme range from factors of ~20 to several hundreds for the test case chosen (gyrofluid magnetic reconnection). The method is second order accurate even if the diffusion and non-linear (advection) operators do not commute, and exhibits the property of robust damping. That is, it correctly captures the analytic solution in the limit $\mathcal{D}_\eta \Delta t \gg 1$, where $\mathcal{D}_\eta$ is the diffusion operator and Δt the timestep. In the limit of no physical dissipation, the method is symplectic (at least for a linear wave test problem), i.e. it does not introduce any numerical dissipation.

The derivation of the SI numerical method presented here relies on the physics-based assumption that the fundamental wave dynamics arises as a result of coupling between equations. In the linear regime (small perturbation amplitude) this statement is exact for many physical systems. Our results demonstrate that this same SI operator (scaled to approximate an upper bound on the frequency in the non-linear regime) continues to be effective in the non-linear regime as well, at least for the test problem explored here. We note that the application chosen to study the efficiency of our algorithm supports only one wave (the KAW). It is therefore easier to formulate an efficient SI operator (or an efficient preconditioner for an implicit algorithm) for this system that would be the case for a system of equations that supported multiple waves (i.e. multiple polarizations at each wave number). While the isotropic and homogeneous SI operator used here was fairly effective for the test case we considered, which had some degree of anisotropy and inhomogeneity, there may be cases with stronger anisotropies or inhomogeneities where a more complicated SI operator or preconditioner is needed. Some approaches for dealing with these issues (of multiple waves with anisotropies and inhomogeneities) are discussed in Ref. [11]. The efficiency of the iterative SI algorithm for a multiple wave system remains to be tested.

One of the advantages of a physics-based semi-implicit operator is that one only has to approximate $\omega^2(\boldsymbol{k})$, not the dispersion relation $\omega(\boldsymbol{k})$ itself (and in fact, one only needs an upper bound on ω^2 to insure convergence). Chacón and Knoll [11], Knoll et al. [13] and Knoll and Keyes [14] have discussed how this kind of physics-based semi-implicit operator can be related to preconditioning, and how it turns an originally hyperbolic problem into a diagonally dominant parabolic problem that can be efficiently solved by multigrid and/or preconditioned Krylov (if the preconditioner is sufficiently effective) iterative techniques. In some cases, the resulting parabolic problem can also be efficiently solved with FFTs, which we use here.

The iterative SI approach used here provides a way to monitor and control the SI operator splitting errors, and essentially turns a semi-implicit algorithm into a fully implicit algorithm. Accuracy, rather than stability, sets the timestep Δt. There are several variations of SI algorithms in the literature (such as [17,18,6,7,53,19]), some using staggered time grids. If only one $p_{\max} = 1$ iteration is done, the underlying SI algorithm used here is equivalent to Ref. [18]. The iterative approach presented here could also be applied to other variations of SI algorithms. This could be a straightforward way to extend other existing codes that use SI algorithms.

It is pertinent to compare the numerical method developed here to more elaborate implicit schemes and preconditioning strategies. Within these, Jacobian–Free Newton–Krylov (JFNK) methods seem to be amongst the most sophisticated and promising [14]. (As discussed in Section 2.4, our algorithm has similarities to a Jacobian–Free Newton method, but with simple functional iteration instead of Krylov iteration within each Newton step.) In this respect, the main comparison factor is the CPU time speed-up over explicit approaches – the main purpose of implicit schemes is, after all, to maximize this parameter. Of course, this is highly problem specific, and a general comparison is not possible. For a similar problem to the one dealt with in this paper, i.e. implicit integration of high frequency dispersive plasma waves, a JFNK implicit scheme was developed by Chacón and Knoll [11]. The aim of these authors is the accurate implicit integration of the whistler wave, which has a dispersion relation of the same type as the KAW, i.e. $\omega \sim k_\perp^2$. In that work, CPU time enhancements by factors of up to a maximum of $\sim$30 is quoted. This is roughly comparable to the enhancement factors of 30 and higher found for our method, as shown in Fig. 4a. However, a definitive comparison is not warranted at this time, since both the physics model and the problem are different (Chacón and Knoll use a more complex four-field model, the flux-bundle coalescence problem is different than the reconnection problem done here, grid sizes differ, there are subtle differences in the definition of the CFL condition that influence this comparison, and the amount of speedup depends sensitively on the stage of the non-linear dynamics in our simulation).

It is interesting to note that the case in Table I of [11] that shows the largest speedup factors also corresponds to needing no Krylov iterations for each Newton iteration (as the authors point out, the preconditioner itself is providing a very good initial guess). In that case, the algorithm of Ref. [11] becomes similar to the one presented here, (except for differences in the predictor step and in the treatment of dissipation terms), so one would in fact expect comparable performance. But short of more detailed comparisons, one should simply conclude that the iterative semi-implicit method developed here seems competitive, at least for some problems, with these somewhat more complex approaches. The iterative semi-implicit algorithm has the relative advantage of being simpler to implement (particularly for codes that are already semi-implicit) and works well for the type of problem we have studied here, while a full JFNK algorithm would be more robustly convergent and may be more efficient for other types of problems where the present algorithm needs more than a few iterations.

With regards to the implementation of a similar treatment of the diffusive terms in real-space codes, we begin by noting that, in the final formulation of the numerical scheme Eqs. (27)–(30), what has to be evaluated are operators like $e^{-\mathcal{D}\Delta t}$. One way to evaluate this with a real-space code could be with a variation of exponential propagation iterative methods [20]. Another way would be to use a second order rational-function approximation:

$$e^{-\mathcal{D}\Delta t} \approx (1 + a\mathcal{D}\Delta t)(1 + b\mathcal{D}\Delta t)^{-1}(1 + b\mathcal{D}\Delta t)^{-1}$$

with $a = 1 + \sqrt{2}$, $b = \left(2 + \sqrt{2}\right)/2$. This retains the second order accuracy of a Crank–Nicolson treatment of damping while having the advantage of robust damping in the large $\mathcal{D}\Delta t$ limit (unlike Crank–Nicolson where $e^{-\mathcal{D}\Delta t} \approx (1 + \mathcal{D}\Delta t/2)^{-1}(1 - \mathcal{D}\Delta t/2) \rightarrow -1$), and still remaining relatively easy to evaluate, since inverting the $(1 + b\mathcal{D}\Delta t)$ operator is equivalent to a standard implicit step. Note, however, that this has not been tested by us in this work.

Our code works in Fourier-space, where it is very easy to invert the SI operator presented in Section 3.2, but it should also be possible to implement a very similar SI operator in real-space. Using a Padé approximation for the Γ_0 function and Fourier-transforming to real-space, Eq. (72) becomes $\hat{\omega}^2 = -a_0^2 B_{\perp,\max}^2 \nabla_\perp^2 \left(1 - \rho_\tau^2 \nabla_\perp^2\right)$. The resulting SI operator $(1 + \Delta t^2 \hat{\omega}^2/4)$ is a symmetric positive-definite parabolic operator that could be solved with a preconditioned iterative Krylov solver or multigrid algorithm. (In some cases, a direct sparse matrix solver can also be used.) It should be noted that if a Krylov solver is used, it is important to use a very good preconditioner. For these types of parabolic problems without preconditioning, the number of iterations required for Krylov solvers like Conjugate-Gradients or GMRES scales with the square root of the condition number of the matrix [47,48], $N_{\text{iters}} \propto \sqrt{1 + \Delta t^2 \hat{\omega}_{\max}^2/4} \propto \Delta t \hat{\omega}_{\max}$, which would give it at most an order unity speed-up over an explicit code. To get a significant speed-up, an iterative solver must be aided by further algebraic preconditioning, such as modified-ILU [49], multigrid, or multilevel additive Schwarz [14], or some combination thereof. Again, we note that this has not been tested by us: in our application, direct inversion of the operator (72) is trivial since we use a pseudo-spectral code.

An important observation from the physics point of view is that these results, i.e. the fact that the KAW can be integrated implicitly without affecting the physics of the system, confirm the parasitic role of this wave in the reconnection process, although the terms from which they arise are essential to the observed speed-up of the instability growth rate. A similar conclusion has been reported by Chacón and Knoll [11] for the whistler wave.

Finally, we note that this numerical method could be useful for other problems where an implicit treatment of high frequency waves is desired, and in particular is trivially applicable to fluid studies of collisionless reconnection, where electron inertia is added to Eqs. (57)–(59) and replaces the role played by the resistivity in breaking the frozen flux condition.

Acknowledgments

It is a pleasure to acknowledge a very useful discussion with L. Chacón. Conversations with R. Samtaney and V.S. Lukin are also gratefully acknowledged. This work was funded by The Center for Multiscale Plasma Dynamics, the US Department of Energy Grant No. DE-FC02-04ER54784, and at the Princeton Plasma Physics Laboratory by US Department of Energy Grant No. DE-AC02-76CH03073.

Appendix A. Exact integration of the diffusive exponential in Eqs. (7) and (8)

It is instructive to consider one logical alternative to the discretization of Eqs. (7) and (8) performed in Eqs. (9) and (10). For simplicity, let us instead consider the more compact form of the problem given by Eq. (45). A formal solution to this equation can be written as

$$\tilde{\psi}^{n+1} = \tilde{\psi}^{n} + \int_{t_n}^{t_{n+1}} \mathrm{d}t\, e^{\mathcal{D}_\eta t} \mathcal{F}(\psi). \tag{A.1}$$

In the Crank–Nicolson algorithm of Eqs. (9) and (10), what is done is to approximate the integral with a second order midpoint evaluation, i.e.

$$\int_{t_n}^{t_{n+1}} \mathrm{d}t\, \mathrm{e}^{\mathcal{D}_\eta t}\mathcal{F}(\psi) \approx \frac{\Delta t}{2}[\mathrm{e}^{\mathcal{D}_\eta t_n}\mathcal{F}(\psi^n) + \mathrm{e}^{\mathcal{D}_\eta t_{n+1}}\mathcal{F}(\psi^{n+1})], \tag{A.2}$$

from which formulas of the kind of (9) and (10) directly follow. Instead, one can exactly integrate the exponential term, while using the CN discretization for the $\mathcal{F}$ operator alone. Undoing the variable substitutions of (6), we obtain:

$$\psi^{n+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + \frac{\Delta t}{2}\left(\frac{1-\mathrm{e}^{-\mathcal{D}_\eta \Delta t}}{\mathcal{D}_\eta \Delta t}\right)[\mathcal{F}(\psi^n) + \mathcal{F}(\psi^{n+1})]. \tag{A.3}$$

Now following a procedure similar to that of Section 2.5, it can be shown that this method reduces to Eq. (52) in the limit of $\mathcal{F}\Delta t \ll 1$, $\mathcal{D}_\eta \Delta t \ll 1$, i.e. it is also second order accurate even if the operators $\mathcal{F}$ and $\mathcal{D}_\eta$ do not commute. It does not, however, exhibit the property of robust damping discussed in Section 2.6 as can easily be seen by evaluating the above expression in the limit $\mathcal{D}_\eta \Delta t \gg 1$ (but it does correctly treat the case of $\mathcal{F}(\psi) = \text{const.}$, which our proposed algorithm does not, see Appendix B. A method to accurately deal with both cases in the limit of $\mathcal{D}_\eta \Delta t \gg 1$ is left to a future publication).

Yet another second order accurate alternative is Strang-splitting [54]. For the model problem Eq. (55), one form of Strang-splitting is $\psi^{n+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t/2}(1+\mathrm{i}\omega\Delta t/2)^{-1}(1-\mathrm{i}\omega\Delta t/2)\mathrm{e}^{-\mathcal{D}_\eta \Delta t/2}\psi^n$. Similarly to our numerical scheme of Eqs. (27)–(30), this also exhibits robust-damping. However, we found this method to be much less efficient in terms of the timestep enhancements that can be obtained over the CFL condition. Further tests would be required to understand why our splitting scheme is a better alternative than Strang-splitting for this problem; the interested reader is referred to a recent paper by Kozlov et al. [55] on the nature of the local error for splitting methods applied to stiff problems.

Appendix B. Inclusion of the equilibrium source term in Eqs. (27)–(30)

The equilibrium source term on the RHS of Eq. (58), $E_{\text{ext}} = \mathcal{D}_\eta \psi_{\text{eq}}$, is not explicitly accounted for in the derivation of the numerical method of Eqs. (27)–(30). Since $\mathcal{F}$ is a general operator, it can, of course, include this term. A more accurate alternative is obtained using the fact that $\partial\psi_{\text{eq}}/\partial t = 0$. Thus, Eq. (58) can be recast in terms of $\psi_1 = \psi - \psi_{\text{eq}}$ in the form:

$$\frac{\partial \psi_1}{\partial t} = \mathcal{F}(\phi, n_{\mathrm{e}}, \psi) - \mathcal{D}_\eta \psi_1. \tag{B.1}$$

Now defining $\psi_1 = \mathrm{e}^{-\mathcal{D}_\eta t}\tilde{\psi}_1$, we obtain:

$$\frac{\partial \tilde{\psi}_1}{\partial t} = \mathrm{e}^{\mathcal{D}_\eta t}\mathcal{F}(\phi, n_{\mathrm{e}}, \psi), \tag{B.2}$$

which is of the same form as Eq. (7) and so the derivation of the method can be carried out as in Section 2.1. Eqs. (28) and (29) are thus replaced by

$$\psi^{n+1,*} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + (1-\mathrm{e}^{-\mathcal{D}_\eta \Delta t})\psi_{\text{eq}} + \frac{\Delta t}{2}(1+\mathrm{e}^{-\mathcal{D}_\eta \Delta t})\mathcal{F}(\phi^n, \psi^n), \tag{B.3}$$

$$\psi^{n+1,p+1} = \mathrm{e}^{-\mathcal{D}_\eta \Delta t}\psi^n + (1-\mathrm{e}^{-\mathcal{D}_\eta \Delta t})\psi_{\text{eq}} + \frac{\Delta t}{2}\mathrm{e}^{-\mathcal{D}_\eta \Delta t}\mathcal{F}(\phi^n, \psi^n) + \frac{\Delta t}{2}\mathcal{F}(\phi^{n+1,p}, \psi^{n+1,p}) - \frac{\hat{\omega}^2 \Delta t^2}{4}(\psi^{n+1,p+1} - \bar{\psi}^{n+1,p}). \tag{B.4}$$

References

[1] D. Keyes, A science-based case for large-scale simulation, vol. 2, Technical Report, Office of Science, US Department of Energy, 2004. <http://www.pnl.gov/scales/>.

[2] W.M. Tang, V.S. Chan, Advances and challenges in computational plasma science, Plasma Phys. Control. Fusion 47 (2005) R1–R34.

[3] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover Publications Inc., 2001.

[4] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer-Verlag Inc., New York, 1999.

[5] A. Robert, T.L. Yee, H. Ritchie, A semi-Lagrangian and semi-implicit numerical integration scheme for multilevel atmospheric models, Mon. Weather Rev. 113 (1985) 388–394.

[6] D.D. Schnack, D.C. Barnes, Z. Mikic, D.S. Harned, E.J. Caramana, Semi-implicit magnetohydrodynamic calculations, J. Comp. Phys. 70 (1987) 330–354.

[7] D.S. Harned, Z. Mikic, Accurate semi-implicit treatment of the Hall effect in magnetohydrodynamic computations, J. Comp. Phys. 83 (1989) 1–15.

[8] A.H. Glasser, C.R. Sovinec, R.A. Nebel, T.A. Gianakon, S.J. Plimpton, M.S. Chu, D.D. Schnack, the NIMROD team, The NIMROD code: a new approach to numerical plasma physics, Plasma Phys. Control. Fusion 41 (1999) A747–A755.

[9] C.R. Sovinec, A.H. Glasser, T.A. Gianakon, D.C. Barnes, R.A. Neber, S.E. Kruger, D.D. Schnack, S.J. Plimpton, A. Tarditi, M.S. Chu, the NIMROD team, Nonlinear magnetohydrodynamics simulation using high-order finite elements, J. Comp. Phys. 195 (2004) 355–386.

[10] L. Chacón, D.A. Knoll, J.M. Finn, An implicit, nonlinear reduced resistive MHD solver, J. Comp. Phys. 178 (2002) 15–36.

[11] L. Chacón, D.A. Knoll, A 2D high-β Hall MHD implicit nonlinear solver, J. Comp. Phys. 188 (2003) 573–592.

[12] D.R. Reynolds, R. Samtaney, C.S. Woodward, A fully implicit numerical method for single-fluid resistive magnetohydrodynamics, J. Comp. Phys. 219 (2006) 144–162.

[13] D.A. Knoll, V.A. Mosseau, L. Chacón, J. Reisner, Jacobian–Free Newton–Krylov methods for the accurate time integration of stiff wave systems, J. Sci. Comp. 25 (1/2) (2005) 213–230.

[14] D.A. Knoll, D.E. Keyes, Jacobian–Free Newton–Krylov methods: a survey of approaches and applications, J. Comp. Phys. 193 (2004) 357–397.

[15] Satish Balay, William D. Gropp, Lois C. McInnes, Barry F. Smith, Petsc users manual, Technical Report ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2003.

[16] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc web page, 2001. <http://www.mcs.anl.gov/petsc>.

[17] D.S. Harned, W. Kerner, Semi-implicit method for three-dimensional compressible magnetohydrodynamic simulation, J. Comp. Phys. 60 (1985) 62–75.

[18] D.S. Harned, D.D. Schnack, Semi-implicit method for the long time scale magnetohydrodynamic computations in three dimensions, J. Comp. Phys. 65 (1986) 57–70.

[19] E.J. Caramana, Derivation of implicit difference schemes by the method of differential approximation, J. Comp. Phys. 96 (1991) 484–493.

[20] M. Tokman, Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods, J. Comp. Phys. 213 (2006) 748.

[21] Maple is a software package for symbolic mathematics. <http://www.maplesoft.com>.

[22] A.H. Baker, E.R. Jessup, T. Manteuffel, A technique for accelerating the convergence of restarted GMRES, SIAM J. Matrix Anal. Appl. 26 (4) (2005) 962–984.

[23] Uri M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (1995) 797.

[24] S.J. Ruuth, Implicit–explicit methods for reaction–diffusion problems in pattern formation, J. Math. Biol. 34 (1995) 148.

[25] Dieter Biskamp, Magnetic Reconnection, Cambridge University Press, 2000.

[26] Eric Priest, Terry Forbes, Magnetic Reconnection: MHD Theory and Applications, Cambridge University Press, 2000.

[27] P.A. Sweet, Mechanisms of solar flares, Annu. Rev. Astron. Astrophys. 7 (1969) 149.

[28] T. Yokoyama, K. Akita, T. Morimoto, K. Inoue, J. Newmark, Clear evidence of reconnection inflow of a solar flare, Astrophys. J. 546 (2001) L69.

[29] J.W. Dungey, Interplanetary magnetic field and the auroral zones, Phys. Rev. Lett. 6 (2) (1961) 47.

[30] H.U. Frey, T.D. Phan, S.A. Fuselier, S.B. Mende, Continuous magnetic reconnection at Earth's magnetopause, Nature 426 (2003) 533–537.

[31] C.J. Xiao et al., In situ evidence for the structure of the magnetic null in a 3D reconnection event in the Earth's magnetotail, Nat. Phys. 2 (2006) 478.

[32] R.J. Hastie, Sawtooth instability in tokamak plasmas, Astrophys. Space Sci. 256 (1997) 177.

[33] H.P. Furth, J. Killeen, M.N. Rosenbluth, Finite-resistivity instabilities of a sheet pinch, Phys. Fluid 6 (1963) 459–484.

[34] A.Y. Aydemir, Nonlinear studies of $m = 1$ modes in high-temperature plasmas, Phys. Fluid B 4 (11) (1992) 3469–3472.

[35] J. Birn, J.F. Drake, M.A. Shay, B.N. Rogers, R.E. Denton, M. Hesse, M. Kuznetsova, Z.W. Ma, A. Bhattacharjee, A. Otto, P.L. Pritchett, GEM magnetic reconnection project, J. Geophys. Res. 106 (A3) (2001) 3715–3719.

[36] L.M. Malyshkin, T. Linde, R.M. Kulsrud, Magnetic reconnection with anomalous resistivity in two-and-a-half dimensions. I: Quasistationary case, Phys. Plasmas 12 (2005) 102902.

[37] R.G. Kleva, J.F. Drake, F.L. Waelbroeck, Fast reconnection in high temperature plasmas, Phys. Plasmas 2 (1) (1995) 23–34.

[38] B.N. Rogers, R.E. Denton, J.F. Drake, M.A. Shay, Role of dispersive waves in collisionless magnetic reconnection, Phys. Rev. Lett. 87 (19) (2001).

[39] B. Rogers, L. Zakharov, Collisionless $m = 1$ reconnection in tokamaks, Phys. Plasmas 3 (6) (1996) 2411–2422.

[40] J.A. Breslau, S.C. Jardin, Global extended magnetohydrodynamic studies of fast magnetic reconnection, Phys. Plasmas 10 (5) (2003) 1291–1298.

[41] P.B. Snyder, G.W. Hammett, A Landau fluid model for electromagnetic plasma microturbulence, Phys. Plasmas 8 (7) (2001) 3199–3216.

[42] T.J. Schep, F. Pegoraro, B.N. Kuvshinov, Generalized two-fluid theory of nonlinear magnetic structures, Phys. Plasmas 1 (9) (1994) 2843–2852.

[43] W.W. Lee, Gyrokinetic approach in particle simulation, Phys. Fluid 26 (2) (1983) 556–562.

[44] G.W. Hammett, W. Dorland, F.W. Perkins, Fluid models of phase mixing, Landau damping, and nonlinear gyrokinetic dynamics, Phys. Fluid B 4 (7) (1992) 2052–2061.

[45] H.R. Strauss, Nonlinear three-dimensional magnetohydrodynamics of noncircular tokamaks, Phys. Fluid 19 (1) (1976) 134–140.

[46] D. Grasso, F. Califano, F. Pegoraro, F. Porcelli, Ion Larmor radius effects in collisionless reconnection, Plasma Phys. Rep. 26 (6) (2000) 512–518.

[47] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[48] O. Axelson, Iterative Solution Methods, Cambridge University Press, 1996.

[49] Ivar Gustafsson, A class of first order factorization methods, BIT 18 (1978) 142–156.

[50] N.F. Loureiro, Studies of nonlinear tearing mode reconnection, Ph.D. Thesis, University of London, 2005.

[51] N.F. Loureiro, S.C. Cowley, W.D. Dorland, M.G. Haines, A.A. Schekochihin, *X*-point collapse and saturation in the nonlinear tearing mode reconnection, Phys. Rev. Lett. 23 (2005) 235003.

[52] T. Tatsuno, W. Dorland, Magneto-flow instability in symmetric field profiles, Phys. Plasmas 13 (2006) 092107.

[53] R. Lionello, Z. Mikić, J.A. Linker, Stability of algorithms for waves with large flows, J. Comp. Phys. 152 (1999) 346.

[54] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (3) (1968) 506–517.

[55] R. Kozlov, A. Kværnø, B. Owren, The behaviour of the local error in splitting methods applied to stiff problems, J. Comp. Phys. 195 (2004) 576–593.