Journal of Computational Physics 316 (2016) 519–533

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An implicit δf particle-in-cell method with sub-cycling and orbit averaging for Lorentz ions

Benjamin J. Sturdevant [a,b,*], Scott E. Parker [a], Yang Chen [a], Benjamin B. Hause [a]

[a] Department of Physics, University of Colorado at Boulder, Boulder, CO 80309, USA
[b] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 15 April 2016
Accepted 17 April 2016
Available online 19 April 2016

Keywords:
Orbit averaging
Sub-cycling
δf method
FLR effects
Implicit particle-in-cell
Lorentz ions
Magnetized plasma simulation
GPU computing

ABSTRACT

A second order implicit δf Lorentz ion hybrid model with sub-cycling and orbit averaging has been developed to study low-frequency, quasi-neutral plasmas. Models using the full Lorentz force equations of motion for ions may be useful for verifying gyrokinetic ion simulation models in applications where higher order terms may be important. In the presence of a strong external magnetic field, previous Lorentz ion models are limited to simulating very short time scales due to the small time step required for resolving the ion gyromotion. Here, we use a simplified model for ion Landau damped ion acoustic waves in a uniform magnetic field as a test bed for developing efficient time stepping methods to be used with the Lorentz ion hybrid model. A detailed linear analysis of the model is derived to validate simulations and to examine the significance of ion Bernstein waves in the Lorentz ion model. Linear analysis of a gyrokinetic ion model is also performed, and excellent agreement with the dispersion results from the Lorentz ion model is demonstrated for the ion acoustic wave. The sub-cycling/orbit averaging algorithm is shown to produce accurate finite-Larmor-radius effects using large macro-time steps sizes, and numerical damping of high frequency fluctuations can be achieved by formulating the field model in terms of the perturbed flux density. Furthermore, a CPU–GPU implementation of the sub-cycling/orbit averaging is presented and is shown to achieve a significant speedup over an equivalent serial code.

1. Introduction

Modern research on low-frequency, ion-Larmor-radius scale fluctuations in magnetized plasmas is based on gyrokinetic ion models. One advantage for using gyrokinetic ion models, as opposed to models using the full Lorentz force equations of motion, is that the analytical elimination of the ion gyration time-scale in gyrokinetic models relaxes time step size constraints in numerical implementations. Additionally, gyrokinetic simulations accurately model $k_\perp \rho_i \sim O(1)$ effects without introducing noise associated with ion Bernstein waves, where $k_\perp$ is the wavenumber perpendicular to $\mathbf{B}$ and ρ_i is the ion gyroradius. Gyrokinetic theory, however, is based on a number of ordering assumptions which must hold to ensure the accuracy of the model. In certain applications where gyrokinetic ordering assumptions may be in questions, for example, in the tokamak edge pedestal region where gradient scale lengths can be comparable to the ion-Larmor-radius, higher order

* Corresponding author at: Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, USA.
E-mail address: benjamin.j.sturdevant@gmail.com (B.J. Sturdevant).

http://dx.doi.org/10.1016/j.jcp.2016.04.036
0021-9991/

terms may be important. Extending gyrokinetic ion models for such applications, however, can be non-trivial and can lead to challenging numerical implementations [1–4].

There has been recent interest in developing models using the full Lorentz force equations of motion for ions [5–7]. Such models offer formal simplicity over gyrokinetic models and can provide an important validation tool or replacement for gyrokinetic ion models in applications where higher order terms may be important. Since time step size restrictions in simulation models which include kinetic electrons often require modern gyrokinetic codes to be run with time step sizes $\Omega_i \Delta t \sim 1$, the use of models which fully resolve the ion gyromotion may be feasible without a large increase in computational effort. Furthermore, recent efforts in optimizing particle-in-cell (PIC) algorithms for modern computing architectures, such as graphics processing units (GPUs), holds promise for handling the more expensive particle integration of the Lorentz ion model [8–11].

In this paper, we explore an implicit orbit averaging/sub-cycling (OASC) time stepping algorithm which may be useful for extending the ability of Lorentz force ion models to simulate longer time scales. This algorithm is shown to accurately produce finite-Larmor-radius (FLR) effects at perpendicular wave numbers $k_\perp \rho_i \sim O(1)$ while advancing the fields on a macro time step ΔT larger than that required to resolve the ion gyromotion. The accuracy of the ion gyromotion is preserved by sub-cycling the computational particles on a micro time step Δt chosen such that $\Omega_i \Delta t \ll 1$. The algorithm is applied to a model problem for ion Landau damped ion acoustic waves in a magnetized plasma. This model problem may be easily extended to model the ion temperature gradient (ITG) instability in slab geometry as in [7]. Linear theory for the model is derived to validate simulation results. Comparisons are also made with a linear dispersion relation obtained from the analysis of a gyrokinetic ion model. The dispersion results show very good agreement between the two models for the low frequency ion acoustic wave.

A notable effect in simulations using Lorentz force ions is the introduction of ion Bernstein waves near harmonics of the ion gyro-frequency [12,13]. These are electrostatic normal modes, which are analytically eliminated in gyrokinetic models, but are present when full ion dynamics are included. Linear theory based on the Laplace transform method is presented to determine the amplitudes of the normal modes relative to the initial perturbation size. The theory predicts ion Bernstein wave amplitudes which are comparable to the ion acoustic wave amplitude. Since the ion Bernstein waves are not damped, their presence in simulations may be undesirable for studies of low-frequency fluctuations. It is demonstrated that formulating the electrostatic field equation in terms of the ion particle flux density results in numerical damping for the ion Bernstein waves.

This paper is organized as follows. In Section 2, our model problem for ion Landau damped ion acoustic waves in a magnetized plasma is presented. Section 3 gives the linear theory for the model problem, including an analysis to derive information on the amplitudes of the normal modes. Section 4 gives the numerical methods used in our simulation model. In Section 5, simulation results are presented to demonstrate the numerical propertied of the implicit OASC algorithm and the accurate production of FLR effects at large macro time step sizes. Here, a comparison with the gyrokinetic ion model is also presented. A hybrid CPU–GPU implementation of our simulation model is discussed in Section 6 and is shown to achieve a speedup factor of ~ 48 compared to an equivalent serial CPU implementation. Section 7 contains further discussion and a summary.

2. Kinetic model for magnetized ion acoustic waves

Here we introduce the equations for the ion Landau damped ion acoustic wave model. We consider a uniform equilibrium distribution $\nabla f_0 = 0$ in a straight, uniform magnetic field $\mathbf{B} = B_0 \hat{z}$ and a self generated electrostatic field $\mathbf{E} = -\nabla\phi$, where ϕ is the electrostatic potential. The model is 2D-3V, meaning it is defined over two spatial dimensions and three velocity dimensions. The spatial dependence of quantities is over the two dimensional domain $(y, z) \in [0, L_\perp) \times [0, L_\parallel)$ and periodicity is assumed in both y and z outside the domain with periods $L_\perp$ and $L_\parallel$ respectively. The velocity dependence of quantities is over $(v_x, v_y, v_z) \in \mathbb{R}^3$. The ion distribution function f_i is taken to follow the Vlasov equation:

$$\frac{\partial f_i}{\partial t} + \mathbf{v} \cdot \nabla f_i + \frac{q_i}{m_i} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_i = 0, \tag{1}$$

where q_i and m_i are the ion charge and mass respectively. The electrons are assumed to be adiabatic, with number density n_e following:

$$n_e = n_0 \left(1 + \frac{e\phi}{T_e}\right), \tag{2}$$

where n_0 is the equilibrium density, e is the electron charge, and T_e is the electron temperature. Finally, quasi-neutrality is assumed:

$$n \equiv n_e = n_i = \int_{\mathbb{R}^3} f_i d\mathbf{v}. \tag{3}$$

Equations (1)–(3), along with the periodicity assumptions form a closed model. In particular, Eq. (1) can be solved for the ion distribution function f_i, Eq. (3) then used to provide an electron number density n_e, and finally Eq. (2) provides a way

to calculate ϕ. A Maxwellian equilibrium distribution is taken, in which case the normal modes of the model are a low frequency, magnetized ion acoustic waves and high frequency ion Bernstein waves.

3. Linear analysis of the model problem

The linearized Vlasov equation is

$$\frac{\partial \delta f}{\partial t} + \mathbf{v}\cdot\nabla\delta f + (\mathbf{v}\times\boldsymbol{\Omega}_i)\cdot\nabla_{\mathbf{v}}\delta f = -\frac{q_i}{m_i}\mathbf{E}\cdot\nabla_{\mathbf{v}} f_0, \tag{4}$$

where the full ion distribution function is given as $f_i = f_0 + \delta f$ and we have defined $\boldsymbol{\Omega}_i = q_i B_0/m_i\hat{z}$. We consider a Maxwellian equilibrium distribution function

$$f_0(\mathbf{v}) = \frac{n_0}{(2\pi)^{3/2}v_{th}^3} e^{-\frac{|\mathbf{v}|^2}{2v_{th}^2}}, \tag{5}$$

where v_{th} is the ion thermal velocity. The field model couples to Eq. (4) through the perturbed number density δn. We have

$$\frac{e\phi}{T_e} = \frac{\delta n}{n_0} = \frac{1}{n_0}\int_{\mathbb{R}^3} \delta f\, d\mathbf{v}. \tag{6}$$

3.1. Model parameters

An analysis of the model equations shows that there are three dimensionless parameters which determine the behavior of the system. These are the parallel and perpendicular system lengths normalized by the thermal ion gyroradius

$$\frac{L_\parallel}{\rho_i}, \qquad \frac{L_\perp}{\rho_i}, \tag{7}$$

and a ratio involving charges and temperatures

$$\theta \equiv \frac{q_i T_e}{e T_i}. \tag{8}$$

We define the thermal ion gyroradius by $\rho_i = v_{th}/\Omega_i$ and the ion temperature by $T_i = m v_{th}$. For the linear analysis, we are interested in the propagation of plane waves through the plasma, assuming spatial and time dependent quantities vary as:

$$\psi(\mathbf{x}, t) = \tilde{\psi}(\mathbf{k}, \omega) e^{i(\mathbf{k}\cdot\mathbf{x} - \omega t)} \tag{9}$$

where $\mathbf{k} = k_\parallel\hat{z} + k_\perp\hat{y}$ is the wave number and ω is the frequency, which may be complex valued. In this case, it is convenient to use the parameters $k_\parallel\rho_i$ and $k_\perp\rho_i$ in place of the first two. We consider cases where $k_\parallel\rho_i \ll 1$ and $k_\perp\rho_i \sim O(1)$ in order to work in a regime where the validity of gyrokinetic ion models should overlap with the Lorentz ion model. A rough estimate for the time scale separation is also determined by these parameters. Defining the ion sound speed by $c_s^2 \equiv q_i T_e/e T_i$, the time scale for an ion acoustic wave propagating nearly parallel to $\mathbf{B}$ is then roughly $c_s k_\parallel$. The ion gyromotion and Bernstein waves evolve on the time scale of Ω_i. In terms of our parameters, we have

$$\sqrt{\theta} k_\parallel \rho_i = \frac{c_s k_\parallel}{\Omega_i} \ll 1. \tag{10}$$

3.2. Linear theory

The normal modes for the system are studied for the linearized model. The resulting dispersion relation is given by

$$\epsilon(\mathbf{k}, \omega) = 1 - \frac{\theta}{2}\sum_{n=-\infty}^{\infty} Z'\left(\frac{\omega/\Omega_i + n}{\sqrt{2}k_\parallel\rho_i}\right) \mathrm{I}_n(k_\perp^2\rho_i^2) e^{-k_\perp^2\rho_i^2} = 0 \tag{11}$$

where Z is the plasma dispersion function of Fried and Conte [14] and I_n is the nth modified Bessel function of the first kind. The solutions of Eq. (11) give the normal modes of the model, which include a low-frequency, ion Landau damped ion acoustic wave and undamped ion Bernstein waves near harmonics of the ion gyro-frequency. The ion Bernstein waves are a unique feature of the full Lorentz ion model and are not present in gyrokinetic ion models. In our numerical simulations with finite $k_\perp\rho_i$, it was found that the Bernstein waves had amplitudes comparable to the ion acoustic wave. Since the Bernstein waves are undamped, they were found to quickly become dominant in the time histories of ϕ. To validate that

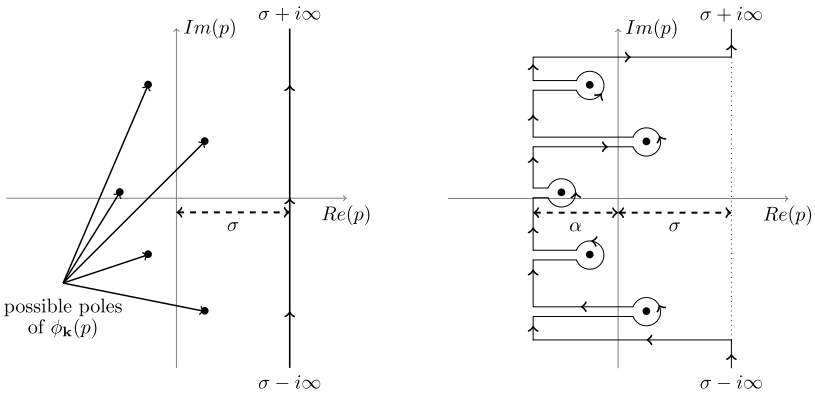

Fig. 1. Contour deformation for obtaining Eq. (15). The figure on the left (a) illustrates the original contour used in the inverse Laplace transform. The figure on the right (b) illustrates the deformed contour.

this feature of our simulations was consistent with the continuous model, we have further developed the linear theory using the Laplace transform in time.

The linear system is studied as an initial value problem using the Laplace transform to determine the amplitudes of each normal mode. The Laplace transform method applied to the one dimensional Landau problem as an initial value problem is presented in a number of plasma physics texts. See for example, Chapter 8 of [15]. The initial condition for the perturbed distribution function is taken to be

$$\delta f(\mathbf{x}, \mathbf{v}, t = 0) = A_0 f_0(\mathbf{v}) e^{i\mathbf{k}\cdot\mathbf{x}}. \tag{12}$$

The Laplace transform pair for a time dependent quantity $\varphi(t)$ is

$$\varphi(p) = \int_0^\infty \varphi(t) e^{-pt} dt, \qquad \varphi(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \varphi(p) e^{pt} dp \tag{13}$$

where p is complex valued and σ can be chosen as any real number which is to the right of all singularities of $\varphi(p)$ in the complex p-plane. The complex variable p is related to the complex frequency of a plane wave ω simply by $p = -i\omega$. Equations (4)–(6) with an initial condition given by Eq. (12) can be solved for the transformed electrostatic potential. The solution of the Laplace (in time) and Fourier (in space) transformed electrostatic potential in terms of p is

$$\frac{e\phi_{\mathbf{k}}}{T_e}(p) = \frac{A_0}{i\sqrt{2}k_\parallel \rho_i} \frac{\sum_{n=-\infty}^{\infty} Z\left(\frac{ip/\Omega_i + n}{\sqrt{2}k_\parallel \rho_i}\right) \mathrm{I}_n(k_\perp^2 \rho_i^2) e^{-k_\perp^2 \rho_i^2}}{\epsilon(\mathbf{k}, ip)}. \tag{14}$$

The time dependent solution of the electrostatic potential can be obtained from the inverse Laplace transform of Eq. (14), which is given by the contour integral in Eq. (13). The evaluation of this contour integral is simplified by deforming the contour of integration as shown in Fig. 1 with a possible set of poles of $\phi_{\mathbf{k}}(p)$, corresponding to the zeros of $\epsilon(\mathbf{k}, ip)$. Justification for the contour deformation is given in Appendix B.

By examining the deformed contour in Fig. 1(b), the time dependent solution of the electrostatic potential can be written as

$$\frac{e\phi_{\mathbf{k}}}{T_e}(t) = \sum_j A_j e^{p_j t} + \frac{1}{2\pi i}\left(\int_{\sigma - i\infty}^{-\alpha - i\infty} \frac{e\phi_{\mathbf{k}}}{T_e}(p) e^{pt} dp + \int_{-\alpha - i\infty}^{-\alpha + i\infty} \frac{e\phi_{\mathbf{k}}}{T_e}(p) e^{pt} dp + \int_{-\alpha + i\infty}^{\sigma + i\infty} \frac{e\phi_{\mathbf{k}}}{T_e}(p) e^{pt} dp \right) \tag{15}$$

where $\{p_j\}$ in the first term are the simple roots of $\epsilon(\mathbf{k}, ip)$, and

$$A_j = \mathrm{Res}\left[\frac{e\phi_{\mathbf{k}}}{T_e}(p), p_j \right]. \tag{16}$$

The main contribution in Eq. (15) comes from the first term. Provided that $\phi_{\mathbf{k}}(p)$ decays rapidly as $\mathrm{Im}(p) \to \pm\infty$, the second and fourth terms in Eq. (15) will vanish. Furthermore, the third term becomes exponentially small compared to the contributions from the poles as $t \to \infty$, leaving the normal modes given by the first term as the time asymptotic solution. In Section 5, we numerically solve for the amplitudes of the normal modes relative to the initial perturbation size A_0

and compare with the amplitudes found in our simulations. This is accomplished by first solving the dispersion relation Eq. (11) numerically for the complex frequencies and then evaluating the corresponding residues in Eq. (16) to obtain the amplitudes A_j.

4. Numerical methods

A number of numerical methods are used to obtain stable, accurate, and low noise simulations of the low frequency ion acoustic wave at large time step sizes. Key features of our 2D-3V simulation model include the δf method which reduces discrete particle noise levels by solving for small perturbations from a Maxwellian equilibrium, a perturbed flux density formulation of the field model which introduces numerical damping of high frequency modes, orbit averaging and sub-cycling (OASC) using separate time step sizes for the particles and fields, a second order implicit integrating scheme to advance the particle orbits and weight equations, and a Picard iterative process to solve the implicit equations.

4.1. δf method

The δf method is utilized, which is effective in reducing discrete particle noise by solving for departures from an known equilibrium distribution [16–20]. The assumption is made that f can be separated into a known, time independent equilibrium part and an unknown perturbed part as $f = f_0 + \delta f$. Particle weights are defined for each computational particle as $w_p = \delta f_p / f_p \approx \delta f_p / f_{0p}$, where the subscript p indicates an evaluation at the phase space location of particle p. The particle weights evolve according to the weight equation, which for linear simulations is

$$\frac{dw_p}{dt} = -\frac{q_i}{m_i}\mathbf{E}_p \cdot \nabla_{\mathbf{v}_p} \ln f_{0p}. \tag{17}$$

For the linear δf scheme, the particles' phase space locations evolve according to their equilibrium trajectories:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \tag{18}$$

$$\frac{d\mathbf{v}_p}{dt} = \mathbf{v}_p \times \mathbf{\Omega}_i \tag{19}$$

Once the particle weights and phase space locations are known, the perturbed number and flux densities at grid point $\mathbf{X}_j$ can be calculated as follows:

$$\delta n_j = \frac{1}{|\Delta \mathbf{X}|}\sum_p w_p S(\mathbf{X}_j - \mathbf{x}_p) \tag{20}$$

$$\delta(n\mathbf{u})_j = \frac{1}{|\Delta \mathbf{X}|}\sum_p w_p \mathbf{v}_p S(\mathbf{X}_j - \mathbf{x}_p) \tag{21}$$

where $|\Delta \mathbf{X}|$ is the cell size and S is the "shape" function [21], which we take to be linear splines.

4.2. Field equation formulations

Recent numerical analysis of implicit δf models has shown that numerical damping can depend on the velocity moments used in the field model equations [22]. Here, we consider two formulations of Eq. (6) to be used in the simulation model. These formulations are equivalent in the continuous limit $\Delta \mathbf{X} \to 0$, $\Delta t \to 0$ but exhibit different properties in the discrete models. The first formulation uses the perturbed number density directly, and the second formulation uses the continuity equation to give the field model in terms of the perturbed flux density. We will refer to these formulations as the perturbed number density form (PND) and the perturbed flux density form (PFD). Simulation results presented in Section 5 show that the PFD form introduces numerical damping of the ion Bernstein waves, where as, the PND form leaves the Bernstein waves undamped. For the PND form, we have simply

$$\frac{e\phi_j^\nu}{T_e} = \frac{\delta n_j^\nu}{n_0}, \tag{22}$$

where ν is the time step index. To derive the PFD form, the partial derivative with respect to time is taken in Eq. (6) and the continuity equation is used to give

$$\frac{\partial}{\partial t}\left(\frac{e\phi_j}{T_e}\right) = -\nabla \cdot \frac{\delta(n\mathbf{u})_j}{n_0} \tag{23}$$

Our baseline time stepping method for the PFD form, without OASC, uses the trapezoidal rule to discretize Eq. (23) in time as

$$\frac{e\phi_j^\nu}{T_e} = \frac{e\phi_j^{\nu-1}}{T_e} - \frac{\Delta t}{2}\left(\nabla\cdot\frac{\delta(n\mathbf{u})_j^\nu + \delta(n\mathbf{u})_j^{\nu-1}}{n_0}\right) \tag{24}$$

where the divergence is taken spectrally in Fourier space. The electric field for both field models is also computed by taking the gradient of ϕ^ν spectrally from the discrete Fourier transform. Simulation results using these two field equation formulations are presented in Section 5.

4.3. Baseline time stepping algorithm

Our baseline time stepping algorithm to which we apply OASC uses a second order implicit scheme to advance the particle positions, velocities, and weight equations. Our motivation for using implicit schemes is for the greater stability that is generally offered. In the process of designing our numerical schemes, we keep in mind future applications using more complex models that may involve higher frequency modes posing severe constraints on the time step size for explicit schemes. For example, in [6], it is known that the compressional Alfven wave can be the source of a numerical instability when it is not well resolved. This can be restrictive since $\omega/\Omega_i \gg 1$ in low-β plasmas. The following implicit time discretization scheme is applied to the ion equations of motion and weight equation to address these difficulties

$$\mathbf{x}^\nu = \mathbf{x}^{\nu-1} + \frac{\Delta t}{2}\left(\mathbf{v}^\nu + \mathbf{v}^{\nu-1}\right) \tag{25}$$

$$\mathbf{v}^\nu = \mathbf{R}\cdot\mathbf{v}^{\nu-1} \tag{26}$$

$$w^\nu = w^{\nu-1} - \frac{\Delta t}{2}\left[G^\nu(\mathbf{x}^\nu,\mathbf{v}^\nu) + G^{\nu-1}(\mathbf{x}^{\nu-1},\mathbf{v}^{\nu-1})\right]. \tag{27}$$

In Eq. (26), the rotation matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} \cos(\Omega_i\Delta t) & \sin(\Omega_i\Delta t) & 0 \\ -\sin(\Omega_i\Delta t) & \cos(\Omega_i\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{28}$$

which produces the correct gyrophase at each time step, and in Eq. (27) we have

$$G^\nu = \frac{q}{m}\mathbf{E}^\nu(\mathbf{x}^\nu)\cdot\nabla_{\mathbf{v}^\nu}\ln f_0(\mathbf{v}^\nu), \tag{29}$$

where the evaluation of the electric field at a particle's position is performed through interpolation as

$$\mathbf{E}^\nu(\mathbf{x}^\nu) = \sum_j \mathbf{E}_j^\nu S(\mathbf{X}_j - \mathbf{x}^\nu). \tag{30}$$

The velocity advance may be extended for nonlinear simulations by including half accelerations due to $\mathbf{E}^{\nu-1}(\mathbf{x}^{\nu-1})$ and $\mathbf{E}^\nu(\mathbf{x}^\nu)$ before and after the rotation, respectively, similar to the Boris push [23,21]. Modification to the weight equation is also needed for nonlinear simulations as in [16].

4.4. Orbit averaging and sub-cycling

For the OASC scheme, the electric field and the computational particles are advanced on separate time steps [24]. The long term goal of this research is to model low frequency ($\omega \ll \Omega_i$) well magnetized plasma physics where gyrokinetics is applicable using a direct Lorentz force method. The main issue at hand is to accurately model the ion FLR effects without including the ion Bernstein waves which are a source of high frequency noise. Because we are interested in low frequency phenomena, we will sub-cycle to resolve the ion cyclotron motion, then orbit-average numerically to accurately resolve ion FLR effects. Orbit averaging and sub-cycling have been explored previously in the context of multi-scale implicit PIC. Besides the seminal orbit-averaging work of B. Cohen and co-workers [25,24,26], a multi-scale method was developed to advance particles depending on their local accuracy in phase space [27,28]. More recently an exact charge and energy-conserving scheme incorporates a "sub-stepping" in time algorithm to avoid particles tunneling through an electrostatic potential barrier and improves momentum conservation [29,10,30]. In our algorithm, the micro time step Δt is used to resolve the fast gyromotion of the ions and the macro time step ΔT is used to resolve the low-frequency fields. These are chosen such that $\Omega_i\Delta t \ll 1$, $\Omega_i\Delta T \gg 1$, and $\Delta T/\Delta t = M$ for $M \in \mathbb{N}$. The particle trajectories and weights are sub-cycled on the micro time step according to Eqs. (25)–(29) where $\mathbf{E}^\nu$ is replaced with $\mathbf{E}^{(N,\nu)}$ for $1 \le \nu \le M-1$. We define $\mathbf{E}^{(N,\nu)}$ as the electric field interpolated in time to the micro time step $t^{(N,\nu)}$ from the fields defined at the macro time steps t^{N-1} and t^N. A simple linear interpolation is used

$$\mathbf{E}^{(N,\nu)} = (1 - \frac{\nu}{M})\mathbf{E}^{N-1} + \frac{\nu}{M}\mathbf{E}^N. \tag{31}$$

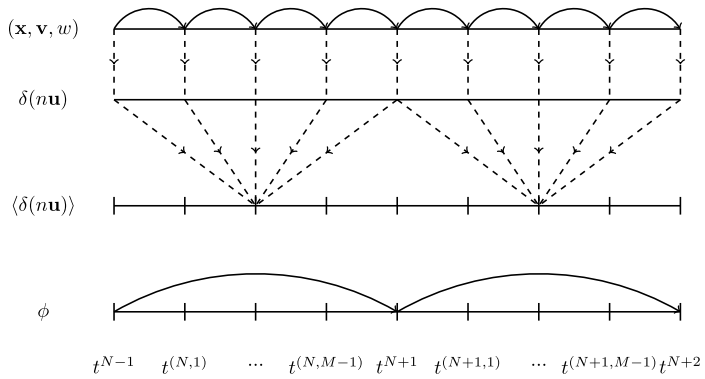

Fig. 2. Illustration of the OASC algorithm. Particle quantities $(\mathbf{x}, \mathbf{v}, w)$ are advanced on the micro time steps using a time interpolated electric field. The flux density $\delta(n\mathbf{u})$ is deposited from the particles at each micro time step to obtain the orbit averaged flux density $\langle\delta(n\mathbf{u})\rangle$, which is used to advance ϕ over the macro time step.

The orbit averaging scheme is derived for the flux form of the field equation Eq. (23). Integrating Eq. (23) between macro time steps t^{N-1} and t^N, we have

$$\frac{e\phi^N}{T_e} = \frac{e\phi^{N-1}}{T_e} - \nabla \cdot \int_{t^{N-1}}^{t^N} \frac{\delta(n\mathbf{u})}{n_0} dt. \tag{32}$$

The integral on the right hand side is then approximated using the composite trapezoidal rule

$$\int_{t^{N-1}}^{t^N} \delta(n\mathbf{u}) dt \approx \frac{\Delta T}{2M} \left(\delta(n\mathbf{u})^{N-1} + 2\delta(n\mathbf{u})^{(N,1)} + 2\delta(n\mathbf{u})^{(N,2)} + \ldots + 2\delta(n\mathbf{u})^{(N,M-1)} + \delta(n\mathbf{u})^N \right), \tag{33}$$

where the perturbed flux densities $\delta(n\mathbf{u})^{(N,\nu)}$ are deposited using particle trajectories and weights on the micro time step $t^{(N,\nu)}$. We will refer to the right hand side of Eq. (33) as the orbit averaged flux density and denote it as $\langle\delta(n\mathbf{u})\rangle^{N-1/2}$. In this notation, our discretized field model is

$$\frac{e\phi^N}{T_e} = \frac{e\phi^{N-1}}{T_e} - \nabla \cdot \langle\delta(n\mathbf{u})\rangle^{N-1/2}. \tag{34}$$

The OASC algorithm is illustrated in Fig. 2. Note that the OASC algorithm reduces to the baseline time stepping algorithm with the PFD form of the field equation when $M = 1$.

4.5. Solution method for the implicit equations

The OASC scheme is implicit and therefore requires the self consistent solution of the particles and electric field at the macro time step t^N. There has been recent progress made in efficient solution methods for fully implicit PIC. These efforts have focused on the use of Jacobian-free Newton–Krylov (JFNK) solvers [31,32,29] and preconditioning to accelerate the convergence of the GMRES iterations [33]. The use of a JFNK solver has not been explored in this work but may hold future promise for increased computational efficiency of the algorithm. For our simulation model, we adopt a Picard iteration scheme to solve the implicit equations. An initial guess is made for ϕ^N and successive corrections to ϕ^N are made by repeatedly advancing the sub-cycled particle system. This process is carried out until the L^2 norm of the residual in Eq. (34) is reduced to a specified tolerance. For our simulations, we have taken both the absolute and relative tolerances to be 5.0×10^{-7}. For the initial guess, we take the value of ϕ at the previous time step. In the tests performed in Section 5, the Picard scheme typically converged in 4–8 iterations, with the number of iterations increasing as expected with larger values of ΔT and $k_\perp \rho_i$.

5. Simulation results

To demonstrate the numerical properties of the algorithms discussed in Section 4 applied to the Lorentz ion model, linear simulations are performed and compared to the theory derived in Section 3. Of interest are the use of the PFD form to damp the high frequency ion Bernstein waves, the accurate production of FLR effects in the OASC algorithm, and the effects of the sub-cycling parameter M in producing accurate simulations over long time scales.

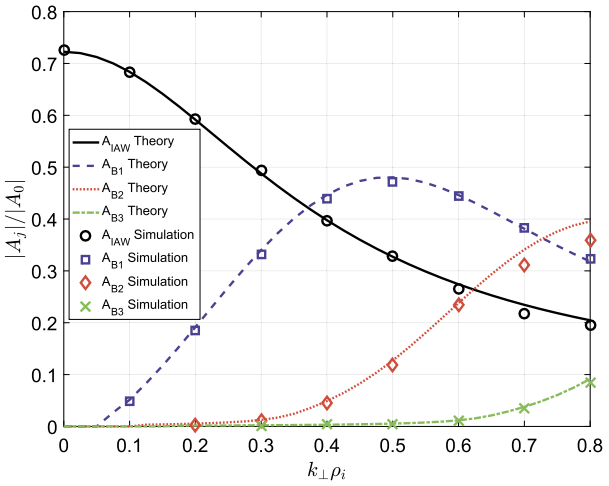

Fig. 3. Amplitudes from simulations using the PND form of the field model are compared to theory for the ion acoustic wave, A_{IAW}, and the first three Bernstein waves, (A_{B1}, A_{B2}, A_{B3}). The amplitudes are given relative to the initial perturbation size, A_0.

5.1. Effects of the field equation formulation on ion Bernstein waves

A notable feature of the Lorentz ion simulation model is the presence of ion Bernstein waves near harmonics of the ion gyrofrequency, which are superimposed on the ion acoustic wave. Although consistent with the physical model, these are undamped high frequency modes and their presence may be undesirable for studies of low frequency phenomena. In addition, Berstein waves are eliminated in gyrokinetic models, which may cause difficulty for comparisons with Lorentz ion models. In simulations using the PND form of the field model, ion Bernstein waves with large amplitudes were found to be present for finite $k_\perp \rho_i$ and to quickly obscure the low frequency ion acoustic wave as this parameter was increased.

In Fig. 3, simulations are performed using the PND form of the field model. The amplitudes for the ion acoustic wave and the first three Bernstein waves are measured relative to the initial perturbation size A_0 for increasing values of $k_\perp \rho_i$ and compared to the theoretical amplitudes from Eq. (16). We use the model parameters $\theta = 5$ and $k_\parallel \rho_i = 6.28 \times 10^{-3}$. The simulations are performed using the baseline time stepping algorithm from Section 4.3 with 131072 computational particles, a mesh size of $n_y \times n_z = 16 \times 32$, and a time step size $\Omega_i \Delta t = 0.125$. Both theory and simulations, using the PND form of the field model, show Bernstein waves with amplitudes comparable to or exceeding that of the ion acoustic wave for finite $k_\perp \rho_i$.

In Fig. 4, we compare simulations at $k_\perp \rho_i = 0.3$ between the PND and PFD forms of the field model. The physical and numerical parameters are taken the same as in Fig. 3. The time history of the first Fourier mode is plotted, demonstrating the numerical damping of the ion Bernstein waves which is achieved only for the PFD form. The difference in the numerical behaviors of the two field model formulations is consistent with the numerical analysis in [22]. In this paper, it is shown that numerical dissipation which is normally present when using implicit schemes can be absent in the δf method when the field model used contains only the perturbed number density as a source term.

5.2. FLR effects for the orbit averaging/sub-cycling algorithm

An important measure of success for the OASC algorithm is the ability to accurately model FLR effects at large time step sizes. This is demonstrated for the ion acoustic wave using the model parameters $\theta = 5$, $k_\parallel \rho_i = 1.61 \times 10^{-3}$ and scanning over values of $k_\perp \rho_i \sim O(1)$. For these simulations, we use 262144 computational particles, a mesh size of $n_y \times n_z = 64 \times 64$, a macro time step size of $\Omega_i \Delta T = 0.75$, and sub-cycling parameter $M = 18$, which corresponds to a micro time step size of $\Omega_i \Delta t = 4.17 \times 10^{-2}$. In Fig. 5, the dispersion results of the simulations are compared to the exact linear dispersion theory for the Lorentz ion model given by Eq. (11). Comparisons are also made with the linear dispersion theory for a gyrokinetic ion model, which is presented in Appendix A. The simulations show excellent agreement with the Lorentz ion dispersion theory using a macro time step size larger than that required to resolve the gyromotion of the ions. Furthermore, the dispersion relation for the gyrokinetic ion model yields nearly identical results to that of the Lorentz ion model for the ion acoustic wave.

5.3. Effects of the sub-cycling parameter

In order to produce accurate simulations over long time scales, sufficient resolution of the ion gyromotion on the micro time step is necessary. Convergence tests are performed, varying the sub-cycling parameter, M. In the first test, the macro time step is kept fixed at $\Omega_i \Delta T = 1.0$ and the sub-cycling parameter, M, is increased from $M = 1$, which corresponds to

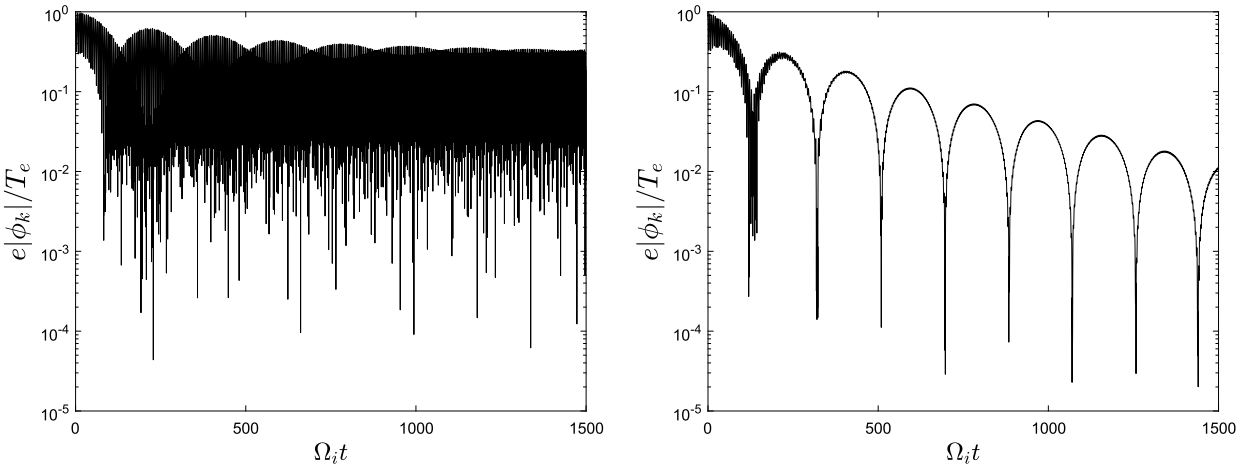

Fig. 4. Time histories of the first Fourier mode amplitude of $\phi_{\mathbf{k}}$. The PND form is used on the left and the PFD form is used on the right. Numerical damping of the ion Bernstein waves occurs only for the PFD form of the field model, leaving a clean simulation of the low frequency ion acoustic wave.

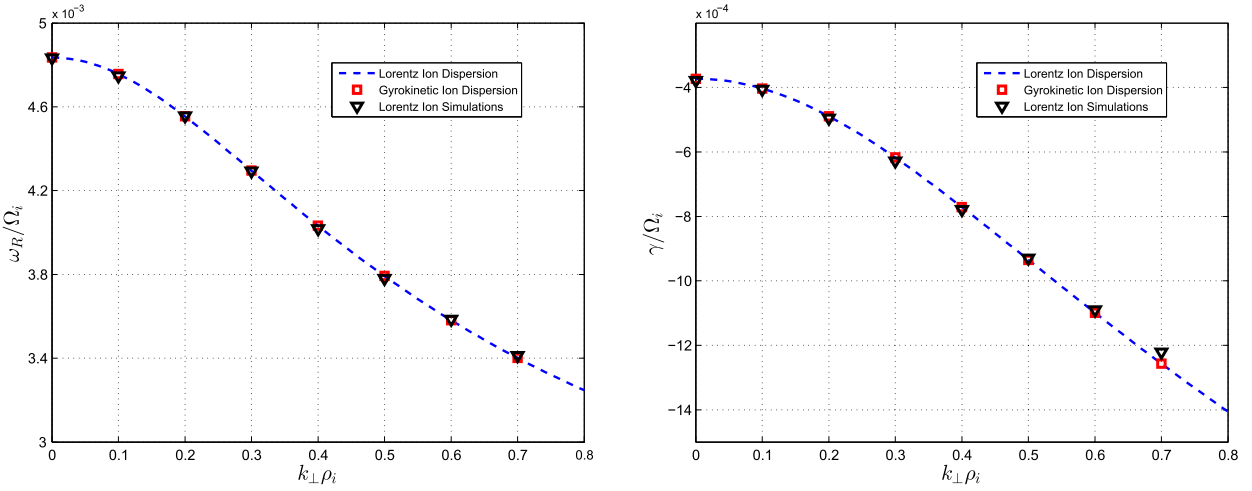

Fig. 5. Dispersion results showing FLR effects on the ion acoustic wave using the parameters $\theta = 5.0$, $k_\parallel \rho_i = 1.61 \times 10^{-3}$, $\Omega_i \Delta T = 0.75$, and $M = 18$ in the OASC algorithm. Data points obtained by solving the gyrokinetic dispersion relation given in Appendix A are also shown.

the baseline time stepping algorithm (without OASC) over the macro time step, up to $M = 16$ which well resolves the ion gyromotion. The time histories of the first Fourier mode amplitude of ϕ for increasing values of M in the first test are given in Fig. 6(a). As expected, the quality of the simulations improves as the sub-cycling parameter is increased. When M is taken too small, large inaccuracies in the simulations develop quickly in time. In the second test, the micro time step size is kept fixed at $\Omega_i \Delta t = 6.25 \times 10^{-2}$ and M is increased to give larger values for the macro time step. The time histories of the first Fourier mode amplitude of ϕ for increasing values of M in the second test are given in Fig. 6(b). It is observed that the time histories are nearly identical in each case, demonstrating the robustness of the algorithm for large macro time steps, provided there is sufficient resolution at the micro time step. For both tests, the same model parameters, particle number, and mesh size are used as in Fig. 5 and $k_\perp \rho_i = 0.4$.

6. CPU–GPU implementation

One promising aspect of the OASC algorithm presented here is that it is amenable to implementation on hybrid architecture utilizing graphics processing units (GPUs) or many integrated core co-processors. The reason for this is that the particle pushing over the micro time steps and the orbit averaging can be done locally on the GPU. The resulting velocity moments can be copied to the CPU memory where a global field solve is done. This eliminates the need for communication of particle data between the CPU and GPU. Many applications of gyrokinetic simulation are for situation where $\Omega_i \Delta t \sim 1$. Additionally, the direct Lorentz ion method presented here can take advantage of hybrid architectures, as we show below. Similarly [10] implemented their energy and charge conserving scheme with sub-stepping and saw speedups over a factor of 100 compared to an equivalent serial CPU implementation. To demonstrate the feasibility of utilizing hybrid architectures

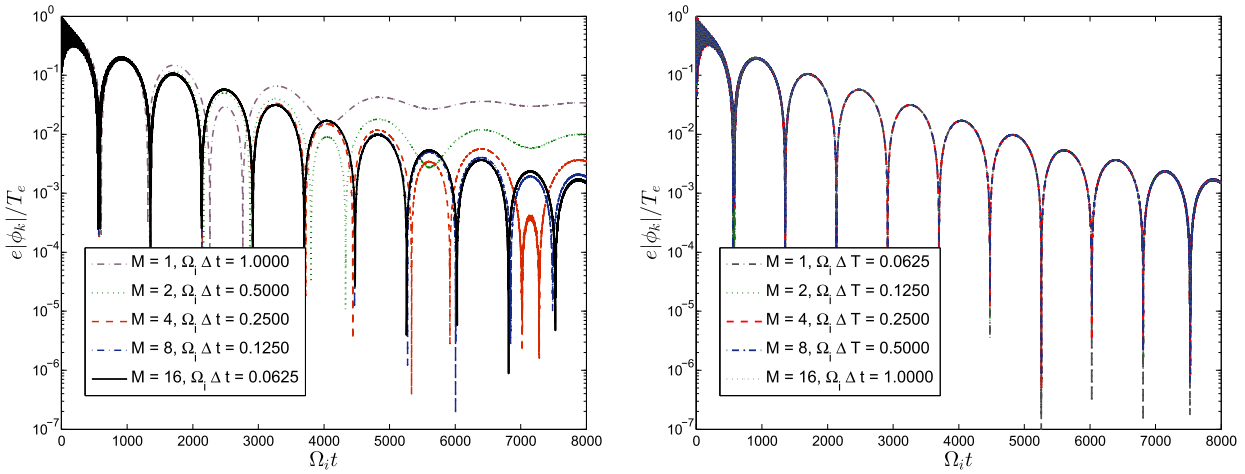

Fig. 6. Time histories of the first Fourier mode amplitude of $\phi_{\mathbf{k}}$. On the left (a), the macro time step size is fixed at $\Omega_i \Delta T = 1.0$ and the sub-cycling parameter M is increased to improve accuracy for long time periods. On the right (b), the micro time step size is fixed at $\Omega_i \Delta t = 6.25 \times 10^{-2}$ and M is increased to give a larger macro time step size.

with sub-cycling and orbit averaging, we have implemented our test bed code on one node of the Titan supercomputer at Oak Ridge National Laboratory. We note this is simply a first step to show the promise of the algorithm. Many node parallelization using MPI is not implemented. Future implementations of the algorithm for solving more realistic turbulence problems will require many nodes (> 100). We also note that MPI optimization is well understood and widely used in PIC codes.

Our CPU–GPU version of the OASC algorithm is implemented in single-precision using CUDA Fortran. Interpolation of field values, particle pushes, and deposits are all performed locally on the GPU, and the field solve is performed on the CPU. The particle data is deposited to arrays stored in global memory using the `atomicadd` function to avoid race conditions, which can occur when more than one thread simultaneously tries to access the same memory location [34]. Although the use of atomic functions can delay the parallel executions in the code, the reduced communication cost between the device and host which is gained by keeping the particle data on the GPU outweighs the serialization that results from the atomic additions. Optimizations to the deposits, including the use of particle sorting, storing multiple copies of the domain in shared memory, and partitioning the grid space into "tiles" have been explored in [11,10]. As a first step in utilizing GPUs, we focus on simplicity of implementation; however, these optimizations are promising for future work to reduce the run-time of the deposit phase in our algorithm.

To benchmark the CPU–GPU implementation, we compare run-times between the single-precision CPU–GPU code and an equivalent single-precision serial CPU code running on the Titan supercomputer. Both codes are compiled with the PGI 15.3.0 compiler using the `-fast` optimization flag. The GPU used is an NVIDIA Tesla K20X, which utilizes the NVIDIA Kepler^{TM} architecture and has a peak theoretical compute performance of 3.95 TFLOPs in single-precision. The host machine is a 16-core 2.2 GHz AMD Opteron^{TM} 6274 processor, for which one core is utilized for both the CPU–GPU and CPU serial implementations.

In Fig. 7, the time per particle per sub-cycle is reported in nano-seconds for the CPU–GPU and CPU serial codes as the number of particles is increased. The test problem uses parameters $\theta = 5.0$, $k_\parallel \rho_i = 1.61 \times 10^{-3}$ and $k_\perp \rho_i = 0.4$. The mesh size is $n_y \times n_z = 64 \times 64$, and the time step size is $\Omega_i \Delta t = 6.25 \times 10^{-2}$ for a sub-cycling parameter $M = 1$, which corresponds to the baseline time stepping algorithm. The largest speedup observed is a factor of 46.9 when 2^{22} particles are used.

In Fig. 8, we examine the effects of the sub-cycling parameter, M, on the run-time of the two implementations. We run with a macro time step size of $\Omega_i \Delta T = 0.75$ for 2^{17}–2^{19} particles, keeping all other parameters the same as for Fig. 7. An additional speedup is observed in the CPU–GPU code as M is increased, due to the increased amount of computation that can be performed on the GPU per communication to the CPU. This speedup is more significant for lower numbers of particles, for example, when using 2^{17} particles, a speedup factor of 3.9 is observed in the CPU–GPU code for $M = 32$ compared to $M = 1$. The serial CPU code; however, is near peak performance for all tests shown in Fig. 8. The largest speedup factor observed between the CPU–GPU and serial CPU codes when increasing M is 47.9 for 2^{19} particles at $M = 32$.

7. Summary and conclusions

In this study, we have explored an implicit δf particle-in-cell method with orbit averaging and sub-cycling algorithm, which is applied to a magnetized plasma simulation model for ion acoustic waves using the full Lorentz force equations of motion for the ions. This algorithm shows promise to extend efforts in the development of direct Lorentz force methods to model low frequency phenomena in well magnetized plasmas. In particular, we were able to produce accurate FLR effects

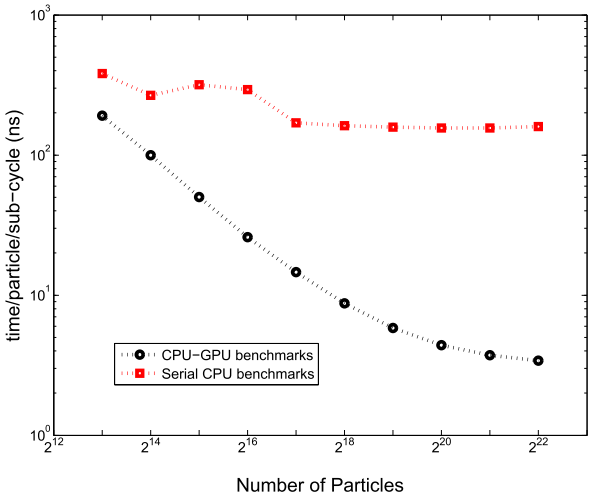

Fig. 7. Benchmarks of the CPU–GPU and serial CPU implementations. The test problem uses sub-cycling parameter $M = 1$, which corresponds to the baseline time stepping algorithm. A speedup factor of 46.9 is observed for 2^{22} particles.

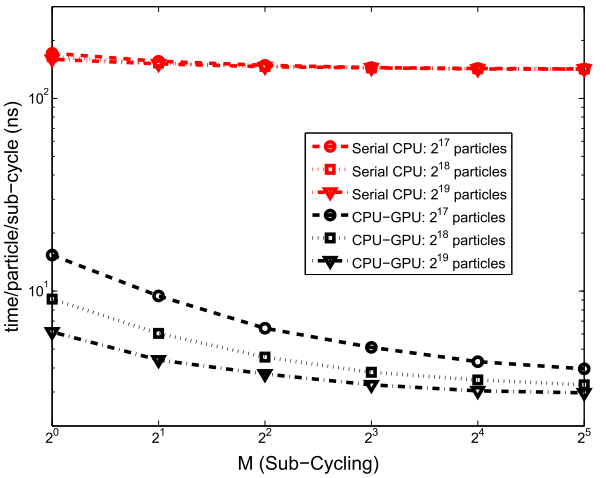

Fig. 8. The sub-cycling parameter, M, is increased and an additional speedup is observed for the CPU–GPU code. The serial CPU code is near peak performance.

over long time scales in our simulations using a full Lorentz force ion model. Additionally, theory has been derived to study the significance of ion Bernstein waves in our model. Ion Bernstein waves are of interest, since they are unique to models using the full Lorentz force equations of motion and are analytically eliminated from gyrokinetic ion models. It is found that the ion Bernstein waves can have a significant effect on simulations, however, numerical damping can be introduced to the ion Bernstein waves when the field equation is formulated in terms of the perturbed flux density in the δf method. This can be beneficial for simulations of low-frequency fluctuations, since the ion Bernstein waves have significant amplitudes for finite $k_\perp \rho_i$, and can obscure physics on longer time scales. Finally, a CPU–GPU implementation of the OASC algorithm has been developed and has achieved a speedup by a factor of ~ 48 compared to an equivalent serial CPU only code. Low communication between the CPU and GPU can be achieved by transferring only grid quantities, making the OASC algorithm well suited to implement on hybrid architectures. Our testbed model of the ion acoustic wave is of interest, since a similar model has been used in the study of the ITG instability in slab geometry [7]. Extending the OASC algorithm to simulate the ITG instability in a toroidal flux tube geometry will be the subject of future work.

Acknowledgements

We would like to acknowledge useful conversations with Drs. T. Manteuffel and C. Leibs regarding this work. This research was supported by the Scientific Discovery through Advanced Computing (SciDAC) program, the Center for Extended Magnetohydrodynamics (CEMM), grant no. DE-FG02-08ER54971, the Edge Physics Simulation Center (EPSI), grant no. DE-

SC000801 and used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Appendix A. Gyrokinetic dispersion relation

We consider a gyrokinetic model similar to Eq. (1) in order to compare the linear theory with that of the Lorentz ion model. For the gyrokinetic model, we consider the ion guiding center distribution function $f_i(\mathbf{R}, \mu, u_\parallel, t)$, where $\mathbf{R}$ is the guiding center coordinate, μ is the magnetic moment, and $v_\parallel$ is the velocity component parallel to the magnetic field. The gyrokinetic Vlasov equation [35] for f_i written in these coordinates and keeping only the terms relevant for a straight, uniform $\mathbf{B}$ field is

$$\frac{\partial f_i}{\partial t} + \left(v_\parallel \hat{\mathbf{b}} + \frac{\bar{\mathbf{E}} \times \hat{\mathbf{b}}}{B} \right) \cdot \frac{\partial f_i}{\partial \mathbf{R}} + \frac{q_i}{m_i} \bar{\mathbf{E}} \cdot \hat{\mathbf{b}} \frac{\partial f_i}{\partial v_\parallel} = 0, \tag{A.1}$$

where $\bar{\mathbf{E}}$ is defined in terms of a gyroaveraging operator as

$$\bar{\mathbf{E}}(\mathbf{R}) = \frac{1}{2\pi} \oint \mathbf{E}(\mathbf{x} = \mathbf{R} - \boldsymbol{\rho}) d\rho, \tag{A.2}$$

and $\boldsymbol{\rho}$ is parameterized by the gyrophase as

$$\boldsymbol{\rho} = \frac{\sqrt{2B\mu}}{\Omega_i} \left(\sin\varphi \hat{x} + \cos\varphi \hat{y} \right) \tag{A.3}$$

for $0 \leq \varphi < 2\pi$. The gyrokinetic quasi-neutrality condition is

$$n_e(\mathbf{x}) = \bar{n}_i(\mathbf{x}) - \frac{q_i}{T_i} \left(\phi(\mathbf{x}) - \tilde{\bar{\phi}}(\mathbf{x}) \right), \tag{A.4}$$

where

$$\bar{n}_i(\mathbf{x}) = \frac{1}{2\pi} \oint \left(\int_0^\infty \int_{-\infty}^\infty f_i(\mathbf{R} = \mathbf{x} - \boldsymbol{\rho}, v_\parallel, \mu) dv_\parallel B d\mu \right) d\rho \tag{A.5}$$

and

$$\tilde{\bar{\phi}}(\mathbf{x}) = \frac{1}{2\pi} \oint \bar{\phi}(\mathbf{R} = \mathbf{x} - \boldsymbol{\rho}) d\rho \tag{A.6}$$

$$\bar{\phi}(\mathbf{R}) = \frac{1}{2\pi} \oint \phi(\mathbf{x} = \mathbf{R} + \boldsymbol{\rho}) d\rho. \tag{A.7}$$

We assume an adiabatic response for the electrons as Eq. (2) to solve for the electrostatic potential. Taking $f_i = f_0 + \delta f$, with

$$f_0(v_\parallel, \mu) = \frac{n_0}{(2\pi)^{3/2} v_{th}^3} e^{-\frac{v_\parallel^2}{2v_{th}^2}} e^{-\frac{B\mu}{v_{th}^2}} \tag{A.8}$$

and linearizing, we have

$$\frac{\partial \delta f}{\partial t} + v_\parallel \hat{\mathbf{b}} \cdot \frac{\partial \delta f}{\partial \mathbf{R}} = -\frac{q_i}{m_i} \bar{\mathbf{E}} \cdot \hat{\mathbf{b}} \frac{\partial f_0}{\partial v_\parallel}. \tag{A.9}$$

To derive the gyrokinetic dispersion relation, we take a plane wave ansatz for all spatial and time dependent quantities and use the following identities when evaluating the gyroaveraging operators:

$$\frac{1}{2\pi} \int_0^{2\pi} e^{ia\cos\varphi} d\varphi = \frac{1}{2\pi} \int_0^{2\pi} e^{ia\sin\varphi} d\varphi = J_0(a) \tag{A.10}$$

where J_0 is the zeroth Bessel function of the first kind. The resulting dispersion relation for the gyrokinetic model is given by

$$\epsilon(\mathbf{k}, \omega) = 1 - \frac{\theta}{2} Z' \left(\frac{\omega/\Omega_i}{\sqrt{2} k_\parallel \rho_i} \right) \mathrm{I}_0(k_\perp^2 \rho_i^2) e^{-k_\perp^2 \rho_i^2} + \theta \left(1 - \mathrm{I}_0(k_\perp^2 \rho_i^2) e^{-k_\perp^2 \rho_i^2} \right) = 0. \tag{A.11}$$

The first two terms are identical to the first two terms of Eq. (11), and the third term is $O(k_\perp^2 \rho_i^2)$, coming from the polarization response in the quasi-neutrality condition. Ion Bernstein waves are no longer present in the gyrokinetic model. The gyrokinetic dispersion relation is solved numerical to make comparisons to the Lorentz ion model in Fig. 5.

Appendix B. Justification for contour deformation in the inverse Laplace transform

The justification for the contour deformation used to obtain Eq. (15) follows from the analyticity of the function G defined by the sum

$$G(u;a,b) = \sum_{n=-\infty}^{\infty} \mathrm{Z}(u+an)\mathrm{I}_n(b)e^{-b}, \tag{B.1}$$

where u is a complex variable and a and b are real parameters. Here Z is the plasma dispersion function of Fried and Conte, which is analytic over the whole complex plane, and I_n is the nth modified Bessel function of the first kind. To prove the analyticity of G, we begin with the following lemma.

Lemma B.1. *The plasma dispersion function is bounded by a function depending only on* $\mathrm{Im}(w)$ *as*

$$|\mathrm{Z}(w)| \leq 2(1+\sqrt{\pi}e^{\mathrm{Im}(w)^2}) \tag{B.2}$$

for all $w \in \mathbb{C}$.

Proof. We begin with an integral definition for Z, which is valid for all complex arguments

$$\mathrm{Z}(w) = 2ie^{-w^2}\int_{-\infty}^{iw} e^{-t^2}dt. \tag{B.3}$$

Setting $w = x+iy$ and taking the contour to be the straight path from $w=-\infty$ to $w=-y$ along the real axis, joined with the straight path from $w=-y$ to $w=-y+ix$ parallel to the imaginary axis, we have

$$\mathrm{Z}(x+iy) = 2ie^{-(x+iy)^2}\left(\int_{-\infty}^{-y} e^{-s^2}ds + i\int_0^x e^{-(y-is)^2}ds\right). \tag{B.4}$$

From this expression, the following bound is readily obtained:

$$|\mathrm{Z}(x+iy)| \leq 2\int_{-\infty}^{-y} e^{y^2-s^2}ds + 2\int_0^{|x|} e^{s^2-x^2}ds. \tag{B.5}$$

The first integral can be bounded by extending the upper limit of integration to $+\infty$. For the second integral, we have

$$\int_0^{|x|} e^{s^2-x^2}ds \leq \int_0^{|x|} e^{sx-x^2}ds = \frac{1-e^{-x^2}}{|x|}. \tag{B.6}$$

This function remains bounded at $x=0$, since $1-e^{-x^2} \sim O(x^2)$. In addition, it remains bounded for all $x \in \mathbb{R}$, since

$$\frac{1-e^{-x^2}}{|x|} = \frac{\int_0^{|x|} 2te^{-t^2}dt}{|x|} \leq \sqrt{2}e^{-1/2} < 1. \tag{B.7}$$

The inequality then follows. □

Theorem B.1. *The series of analytic functions*

$$\sum_{n=-\infty}^{\infty} \mathrm{Z}(u+an)\mathrm{I}_n(b)e^{-b} \tag{B.8}$$

with $a, b \in \mathbb{R}$ *is uniformly convergent over the domain in the complex plane defined by* $D = \{u \in \mathbb{C} : -y_0 < \mathrm{Im}(u) < y_0\}$ *for any* $y_0 > 0$. *Since uniform convergence of a series of analytic functions guarantees analyticity of the sum, the function G is an analytic function of u over D. Furthermore, the series of analytic functions*

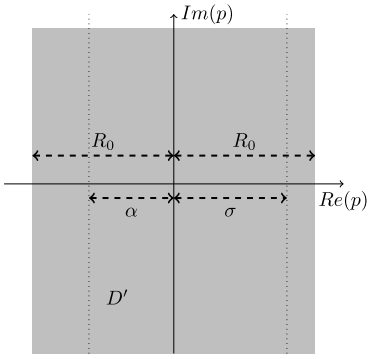

Fig. B.1. The domain $D' = \{p \in \mathbb{C} : -R_0 < \mathrm{Re}(p) < R_0\}$, where $R_0 > \max(\alpha, \sigma)$.

$$\sum_{n=-\infty}^{\infty} Z'(u + an)\mathrm{I}_n(b)e^{-b}, \tag{B.9}$$

also converges to an analytic function in D, which is equal to the derivative of G.

Proof. The proof follows from Lemma B.1 and the Weierstrass M-test. Each term in the series is bounded as

$$|Z(u + an)\mathrm{I}_n(b)e^{-b}| \leq 2(1 + \sqrt{\pi}e^{\mathrm{Im}(w)^2})\mathrm{I}_n(b)e^{-b} \leq 2(1 + \sqrt{\pi}e^{y_0^2})\mathrm{I}_n(b)e^{-b} \tag{B.10}$$

in D. Furthermore, the modified Bessel function series gives

$$\sum_{n=-\infty}^{\infty} 2(1 + \sqrt{\pi}e^{y_0^2})\mathrm{I}_n(b)e^{-b} = 2(1 + \sqrt{\pi}e^{y_0^2}) < \infty. \tag{B.11}$$

The Weierstrass M-test therefore guarantees uniform convergence of the series. The analyticity of G in D follows directly from the uniform convergence of its series definition. Uniform convergence of the series also allows term by term differentiation to obtain a series which converges to the derivative of the sum in D. This follows from a standard theorem in complex analysis. See, for example, Chapter 5 of [36]. □

The contour deformation is justified since $\phi_{\mathbf{k}}(p)$ can be expressed as:

$$\frac{e\phi_{\mathbf{k}}}{T_e}(p) = \frac{A_0}{i\sqrt{2}k_\parallel \rho_i} \frac{G(\frac{ip/\Omega_i}{\sqrt{2}k_\parallel \rho_i}; \frac{1}{\sqrt{2}k_\parallel \rho_i}, k_\perp^2 \rho_i^2)}{1 + \frac{\theta}{2} G'(\frac{ip/\Omega_i}{\sqrt{2}k_\parallel \rho_i}; \frac{1}{\sqrt{2}k_\parallel \rho_i}, k_\perp^2 \rho_i^2)}. \tag{B.12}$$

Since the proof in Theorem B.1 applies to an arbitrarily large portion of the complex plane, we consider in particular the domain $D' = \{p \in \mathbb{C} : -R_0 \leq \mathrm{Re}(p) \leq R_0\}$, where R_0 is chosen such that $R_0 > \max(\alpha, \sigma)$, as illustrated in Fig. B.1. Then $D'/\{p_j\}$ defines a domain in which the contour of Fig. 1(a) can be continuously deformed into that of Fig. 1(b), without crossing any singularities of $\phi_{\mathbf{k}}$.

References

[1] T.S. Hahm, L. Wang, J. Madsen, Phys. Plasmas 16 (2009) 022305.
[2] A.M. Dimits, Phys. Plasmas 19 (2012) 022504.
[3] M. Artun, W.M. Tang, Phys. Plasmas 1 (1994) 2682.
[4] A.J. Brizard, Phys. Plasmas 2 (1995) 459.
[5] J. Cheng, S.E. Parker, Y. Chen, D. Uzdensky, J. Comput. Phys. 245 (2013) 364.
[6] Y. Chen, S.E. Parker, Phys. Plasmas 16 (2009) 052305.
[7] D.D. Schnack, J. Cheng, D.C. Barnes, S.E. Parker, Phys. Plasmas 20 (2013) 062106.
[8] X. Kong, M. Huang, C. Ren, V. Decyk, J. Comput. Phys. 230 (2011) 1676.
[9] V.K. Decyk, T.V. Singh, Comput. Phys. Commun. 182 (2011) 641.
[10] G. Chen, L. Chacon, D.C. Barnes, J. Comput. Phys. 231 (2012) 5374.
[11] V.K. Decyk, T.V. Singh, Comput. Phys. Commun. 185 (2014) 708.
[12] I. Bernstein, Phys. Rev. 109 (1958) 10.
[13] J.P.M. Schmitt, Phys. Rev. Lett. 31 (16) (1973) 982.
[14] B.D. Fried, S.D. Conte, The Plasma Dispersion Function, Academic Press, New York, 1961.
[15] N.A. Krall, A.W. Trivelpiece, Principles of Plasma Physics, McGraw-Hill, 1973.
[16] S.E. Parker, W.W. Lee, Phys. Fluids B 5 (1993) 77.
[17] A.Y. Aydemir, Phys. Plasmas 1 (1994) 822.

[18] G. Hu, J.A. Krommes, Phys. Plasmas 1 (1994) 863.
[19] A.M. Dimits, W.W. Lee, J. Comput. Phys. 107 (1993) 309.
[20] M. Kotschenreuther, Bull. Am. Phys. Soc. 34 (1993) 2107.
[21] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, 1985.
[22] B. Sturdevant, S. Parker, Finite time step and spatial grid effects in δf simulation of warm plasmas, J. Comput. Phys. 305 (2016) 647–663.
[23] J.P. Boris, in: Proc. Fourth Conf. Num. Sim. Plasmas, Naval Res. Lab., Wash., D.C., 1970.
[24] B.I. Cohen, R.P. Freis, V. Thomas, J. Comput. Phys. 45 (1982) 345.
[25] B.I. Cohen, T.A. Brengle, D.B. Conley, R.P. Freis, J. Comput. Phys. 38 (1980) 45.
[26] B.I. Cohen, R.P. Freis, J. Comput. Phys. 45 (1982) 367.
[27] A. Friedman, S.E. Parker, S.L. Ray, C.K. Birdsall, J. Comput. Phys. 96 (1991) 54.
[28] S.E. Parker, A. Friedman, S.L. Ray, C.K. Birdsall, J. Comput. Phys. 107 (1993) 388.
[29] G. Chen, L. Chacon, D.C. Barnes, J. Comput. Phys. 230 (2011) 7018.
[30] L. Chacon, G. Chen, D.C. Barnes, J. Comput. Phys. 233 (2013) 1.
[31] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[32] S. Markidis, G. Lapenta, J. Comput. Phys. 230 (2011) 7037.
[33] G. Chen, L. Chacon, C.A. Leibs, D.A. Knoll, W. Taitano, J. Comput. Phys. 258 (2014) 555.
[34] PGI, CUDA Fortran Programming Guide and Reference, Version 2015.
[35] E.A. Frieman, L. Chen, Phys. Fluids 25 (1982) 502.
[36] D.A. Wunsch, Complex Variables with Applications, 3rd edition, Pearson, 2005.