Journal of Computational Physics 228 (2009) 4773–4788

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

VALIS: A split-conservative scheme for the relativistic 2D Vlasov–Maxwell system

N.J. Sircombe [a,b,*], T.D. Arber [b]

[a] AWE Plc. Aldermaston, Reading, Berkshire RG7 4PR, UK
[b] Centre for Fusion, Space and Astrophysics, University of Warwick, Coventry CV4 7AL, UK

ARTICLE INFO

Article history:
Received 5 December 2008
Received in revised form 16 March 2009
Accepted 21 March 2009
Available online 31 March 2009

MSC:
65P10
82D10

Keywords:
Vlasov
Eulerian
Time-splitting
Piecewise Parabolic Method
Plasma
Raman

ABSTRACT

An accurate treatment of the relativistic Vlasov–Maxwell system is of fundamental importance to a broad range of plasma physics topics, including laser–plasma interaction, transport in solar and magnetospheric plasmas and magnetically confined plasmas. This paper introduces VALIS: an algorithm for the numerical solution of the Vlasov–Maxwell system in two spatial dimensions and two momentum dimensions.

Particular attention is given to the problems of particle gyromotion on Eulerian momentum grids, satisfying Poisson's equation without introducing additional workload or altering the dispersion properties of the solver and the potential problems of applying time-splitting algorithms to relativistic systems.

This work demonstrates that by adopting a conservative, split-Eulerian scheme based on the Piecewise Parabolic Method for the update of the particle distribution function and utilising the exact particle fluxes to calculate the current in the solution of Maxwell's equations, these concerns can easily be addressed.

1. Introduction

The accurate modelling of kinetic plasma physics phemomena is of fundamental importance to a number of problems including: laser absorption and subsequent transport in short-pulse laser-matter interaction [1–3]; growth and saturation of parametric instabilities in long-pulse laser–plasma interaction [4–6]; the laser-driven particle accelerator [7,8]; thermal transport in solar [9] and magnetospheric [10] plasmas; as well as non-linear effects in magnetically confined fusion devices [11].

Kinetic models derive from a statistical treatment of the plasma retaining information about the distribution of particle momenta. In addition to the conventional spatial dimensions, a kinetic model may have up to three velocity (or equivalently momentum) dimensions. Together, these describe a phase space of up to six dimensions. The complexity of such models allows the consideration of complex non-linear kinetic phenomena, but renders problems analytically and computationally challenging. These models can be solved, together with the appropriate equations of electrodynamics, either directly or via particle in cell (PIC) methods (see, for example, Refs. [12–14], and references therein). The PIC approach is widely adopted

* Corresponding author. Address: AWE plc, Aldermaston, Reading, Berkshire RG74PR, UK. Tel.: +44 118 981 4111.
E-mail addresses: nathan.sircombe@awe.co.uk (N.J. Sircombe), t.d.arber@warwick.ac.uk (T.D. Arber).
URL: http://www.warwick.ac.uk/go/cfsa (T.D. Arber).

0021-9991/$ - see front matter Crown Copyright © 2009 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.029

owing to its flexibility, stability and underlying simplicity. The Vlasov approach proposed here is complimentary to the PIC method, with each suited to different problems. The issue of when direct Vlasov or PIC should be used can be addressed by following the argument presented by Besse et al. [15]. Here the ratio of the computational cost of Vlasov compared to PIC is estimated as $(N_v)^{d_v}/n_{\text{pic}}$ where N_v is the number of points in velocity space used in the Vlasov code, d_v is the number of velocity dimensions and n_{pic} is the number of particles per cell in the PIC code. PIC codes exhibit a noise level, i.e. fluctuations in distribution function, which is inversely proportional to n_{pic}. Thus physics determines the required level of accuracy in the distribution function, i.e. the order of magnitude of n_{pic}. Assuming $N_v = 100$ for a reasonable resolution by the Vlasov code it is clear that for 1D velocity space problems that Vlasov is always the preferred method. In 2D velocity space the preferred method depends on the problem. For instabilities which depend on the detailed structure of the distribution function one would need $n_{\text{pic}} \simeq 10^4$ and PIC and Vlasov would have comparable computational cost. If higher resolution is needed then Vlasov is to be preferred. An example where Vlasov is always to be preferred is in the treatment of the Fokker–Planck collision term. Collisional transport properties are determined by high-order velocity moments of the distribution function and many thousands of PIC particles are required to achieve the accuracy of established Vlasov–Fokker–Planck codes [1,16,17]. This is because a raw estimate of the cost of running PIC based on n_{pic} misses the fact that very few of the n_{pic} particles are present in the high energy tail of the distribution function where the resolution is needed most whereas for Vlasov the resolution in velocity space is uniform and noise free. For gross, large scale phenomena fewer particles are needed and PIC becomes the ideal choice for 2D velocity space. In 3D velocity space PIC codes are currently the only computationally tractable approach.

Most numerical solutions to Vlasov's equation adopt some form of time-splitting scheme [18,19]. Conservative [20,21] and semi-Lagrangian [22] approaches have been most popular, but there are alternative schemes [23,24]. Most schemes adopt a fixed Eulerian phase-space grid [26,27], but, for the case of collision dominated plasmas, the closely related Vlasov–Fokker–Planck (VFP) problem has been tackled using a harmonic decomposition of the particle distribution function in momentum space [1,16], with considerable success. More recently some attempts have been made at adaptive Vlasov codes [15]. A recent review of Vlasov solvers [28] compared a number of conservative schemes, based on calculating fluxes between computational cells, and semi-Lagrangian schemes based on updating the distribution by the method of characteristics. This paper showed that low order conservative schemes [29] are too diffusive and that higher order schemes require limiters of some form to maintain positivity in the distribution function. Semi-Lagrangian schemes are accurate for some problems but there was no single scheme which excelled in all tests. The positive and flux conservative (PFC) method [28] was either the best performing, or close to the best, for all tests and did not fail any tests. These findings are supported by an earlier comparison of conservative schemes [21] in which the Piecewise Parabolic Method (PPM) [30] was show to be both robust and accurate across all tests. The PPM method is formally third order away from strong limiters and conservative and thus has very similar accuracy and stability properties as the PFC method. However PPM also ensures monotonicity so that false oscillations, possible sources of kinetic instabilities, cannot be introduced. Thus, while for some tests the semi-Lagrangian scheme may be more accurate than high-order conservative schemes (PFC or PPM), these differences are small. In addition semi-Lagrangian schemes require extra steps to enforce positivity [28]. These additional steps have to date only been implemented in dimensionally split form. Unfortunately the semi-Lagrangian technique must be applied in full multi-dimensional, i.e. unsplit, form for relativistic problems to ensure global conservation [35]. Thus previous papers have shown that to have a robust, high-order Vlasov scheme in which positivity can be easily enforced that conservative schemes (PFC or PPM) are to be preferred. Furthermore if we require that the scheme is monotonicity preserving then PPM is preferred over PFC even though they are both of the same formal order. All schemes to solve Vlasov's equation must introduce some dissipation to continue running past the point when phase-space filamentation reaches grid scale. When this happens the codes necessarily must increase the entropy S defined by $S = -\int f \log f \, dv$, where f is the distribution function. The properties of PFC and PPM approaches with regard to entropy are detailed elsewhere [21,28] but they are similar. Note that for schemes which are not positivity preserving it is not possible to define the real entropy S due to negative f.

In this paper we outline the design, development and testing of VALIS, a split-Eulerian solver for the two-dimensional Vlasov–Maxwell system. We detail the algorithm for updating the particle distribution function and three different approaches to the advance of Maxwell's equations. Particular emphasis is placed on addressing the following key issues:

- Several papers [24,25,22,31,1] have commented that using split-Eulerian advections will not correctly resolve the circular motion of simple gyration about a fixed magnetic field. To avoid this potential problem, schemes have been developed in which the momentum update is 2D [24] or polar phase-space co-ordinates [31] have been used. Here we show that with the formulation of split-Eulerian advections adopted in VALIS the gyromotion is treated correctly.
- It is well known [32] that not all schemes for updating the electromagnetic fields in PIC codes also ensure that Poisson's equation is satisfied to machine precision. To overcome this, PIC codes use either Poisson correctors, which may be local, as for example in Ref. [14], possibly followed by filtering to avoid numerical Cherenkov radiation at short wavelengths [33]. A method does exist for ensuring that Poisson's equation is exactly satisfied for PIC codes [34] and in this paper we show that the same approach can be applied in a split-Eulerian Vlasov solver at no additional computational cost.
- Moving from the leapfrog scheme (often used for Yee staggered grids) to the predictor corrector method (required for the implementation of the scheme which exactly satisfies Poisson's equation) does not significantly change the numerical wave dispersion relations.

- Some Vlasov codes use the semi-Lagrangian approach for the advections. For such schemes there is a potential problem for relativistic problems due to the time-splitting [35]. These problems are avoided by casting the Vlasov equation in conservative form and using conservative transport of the electron distribution function via PPM [30,21].

The final scheme to be described therefore is shown to correctly handle rotation, even though it is on an Eulerian grid; exactly satisfy Poisson's equation without affecting EM wave dispersion or introducing numerical Cherenkov radiation and finally by employing conservative form it is easily extended to relativistic simulations.

1.1. The 2D electromagnetic Vlasov system

The full, relativistic, Vlasov–Maxwell system of equations is given by Vlasov's equation:

$$\frac{\partial f_j}{\partial t} + \frac{\mathbf{u}}{\gamma}\cdot\nabla_{\mathbf{x}} f_j + \frac{q_j}{m_j}\left(\mathbf{E} + \frac{\mathbf{u}}{\gamma}\wedge\mathbf{B}\right)\cdot\nabla_{\mathbf{u}} f_j = 0 \tag{1}$$

where j represents the particle species, together with Maxwell's equations:

$$\nabla\cdot\mathbf{E} = \frac{\rho(\mathbf{x},t)}{\epsilon_0} \tag{2}$$

$$\nabla\cdot\mathbf{B} = 0 \tag{3}$$

$$\nabla\wedge\mathbf{E} = -\frac{\partial\mathbf{B}}{\partial t} \tag{4}$$

$$\nabla\wedge\mathbf{B} = \mu_0\mathbf{J} + \epsilon_0\mu_0\frac{\partial\mathbf{E}}{\partial t} \tag{5}$$

The number density of each species is then defined as

$$n_j(\mathbf{x}) = \int f_j(\mathbf{x},\mathbf{u})\,d\mathbf{u} \tag{6}$$

The charge and current density are defined as

$$\rho(\mathbf{x}) = \sum_j\left(q_j\int f_j(\mathbf{x},\mathbf{u})\,d\mathbf{u}\right) \tag{7}$$

$$\mathbf{J}(\mathbf{x}) = \sum_j\left(q_j\int\frac{\mathbf{u}}{\gamma}f_j(\mathbf{x},\mathbf{u})\,d\mathbf{u}\right) \tag{8}$$

and γ can be defined in terms of $\mathbf{u}$

$$\gamma = \sqrt{1+\frac{|\mathbf{u}|^2}{c^2}} \tag{9}$$

These equations describe the self-consistent evolution of a multi-species system in a six-dimensional phase space, often referred to as a 3D3P model (signifying three spatial and three momentum dimensions). Such an extensive system presents a significant computational challenge for any kinetic model, even with a relatively poorly sampled phase space [36]. Indeed for Vlasov solvers the computational requirements make 3D3P impractical at present. Such high dimensional systems can however be address by PIC techniques and hybrid, i.e. containing both fluid and kinetic species, PIC based simulations are not too demanding for large scale systems where fine details of the tails of distributions functions are not required. From here on we shall consider a 2D slice through the full 3D system, retaining only two spatial and two momentum co-ordinates (2D2P), two electric field components (in the plane), a single magnetic field component (out of the plane) and a single particle species: the electrons. The ions will be assumed to form an immobile background.

Without loss of generality we can choose the (x,y) plane, defining our electric and magnetic field vectors to be $\mathbf{E} = (E_x,E_y,0)$ and $\mathbf{B} = (0,0,B_z)$. We also adopt a system of normalised units where mass is normalised to the electron mass m_e, velocity to the speed of light c, time to the inverse electron plasma frequency ω_{pe}^{-1} (where $\omega_{pe} = (n\,e^2/\epsilon_0 m_e)^{1/2}$), distance to $c\omega_{pe}^{-1}$, electric and magnetic fields to $\omega_{pe}cm_e/e$ and $\omega_{pe}m_e/e$ respectively, and temperatures to $k_B m_e\,c^2$ (where k_B is Boltzmann's constant).

In these units the full 2D2P Vlasov–Maxwell system, with immobile ions, is given by Vlasov's equation for the electron distribution function f_e, that is Eq. (1) with $j = e, m_j = 1, q_j = -1$, $\mathbf{x} = (x,y)$ and $\mathbf{u} = (u_x,u_y)$. Eqs. (4) and (5) become

$$\frac{\partial E_x}{\partial t} = -J_x + \frac{\partial B_z}{\partial y} \tag{10}$$

$$\frac{\partial E_y}{\partial t} = -J_y - \frac{\partial B_z}{\partial x} \tag{11}$$

$$\frac{\partial B_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \tag{12}$$

with the electron density and current defined as

$$n_e = \int f_e \, d\mathbf{u} \tag{13}$$

$$J_{x,y} = -\int \frac{u_{x,y}}{\gamma} f_e \, d\mathbf{u} \tag{14}$$

2. Numerical scheme

The approach taken in the development of VALIS is to build on an existing [21], and proven [37,4,38,39,11] 1D1P algorithm and write the code in a parallelised domain decomposed form in order that it can scale onto current, and future, massively parallel machines.

The update of the particle distributions is based on a split-Eulerian scheme. The fields must be updated by a time-centred and (at least) second order method. We outline three possible approaches: leapfrog, predictor corrector and a third method, based on predictor corrector which is shown to obey Poisson's equation for no extra computational cost. VALIS with a leapfrog field solver will be referred to as `Leapfrog` and as `Predictor Corrector` with the standard predictor corrector. As this work will demonstrate, the third scheme has some advantages over the other two and has been adopted as the default solver in this code. For this reason, it is simply refered to as VALIS.

2.1. The simulation grid

The distribution function is described on a 4D phase-space grid. By convention a cell within this grid is indexed (i,j,m,n). Where i and j represent the cell number in real space (x,y); m and n represent the cell number in momentum space (u_x,u_y). This is a fixed Eulerian grid and runs from $x_{\min}$ to $x_{\max}$ in x,$y_{\min}$ to $y_{y_{\max}}$ in y, $-u_{x_{\max}}$ to $u_{x_{\max}}$ in u_x and $-u_{y_{\max}}$ to $u_{y_{\max}}$ in u_y. The electron distribution function, f_e, is defined on the grid at cell centre in all dimensions (see Fig. 1).

Fields and currents are defined on the cell edges in a similar manner to the Yee mesh common to most finite difference Maxwell solvers [40]. This choice of staggered grid is to simplify the calculation of the curls of **E** and **B** and to avoid the checkerboard instability. We also define two additional co-ordinate grids, see Fig. 1. These are the advection co-ordinates, the velocities required by the advection sweeps of the particle distribution function (as discussed in the following section). They include the relativistic factor γ. For the spatial advections we define

$$h_x(m,n) = \frac{u_x(m)}{\sqrt{1+u_x^2(m)+u_y^2(n)}} \tag{15}$$

$$h_y(m,n) = \frac{u_y(n)}{\sqrt{1+u_x^2(m)+u_y^2(n)}} \tag{16}$$

and for the momentum space advections

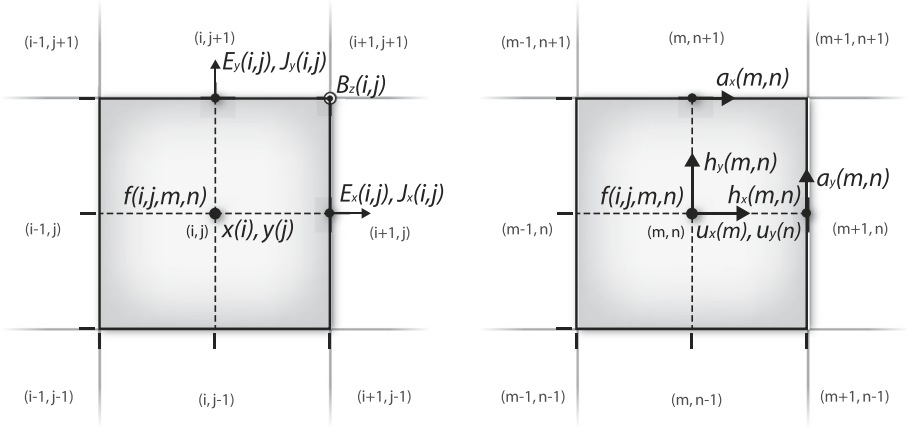

Fig. 1. (Left) The (i,j) spatial grid cell for VALIS. The distribution function is defined at the cell centre. The fields are defined on a staggered grid at the cell faces (for electric fields) and cell corners (for magnetic field). (Right) The (m,n) momentum grid cell. The distribution function is defined at cell centre and the advection co-ordinates $a_{x,y}$ and $h_{x,y}$ are defined as shown.

$$a_x(m,n) = \frac{u_x(m)}{\sqrt{1 + u_x^2(m) + \left(\frac{u_y(n)+u_y(n+1)}{2}\right)^2}} \tag{17}$$

$$a_y(m,n) = \frac{u_y(m)}{\sqrt{1 + \left(\frac{u_x(m)+u_x(m+1)}{2}\right)^2 + u_y^2(n)}} \tag{18}$$

These definitions are easily adapted for non-relativistic, electrostatic problems by setting $h_x = a_x = u_x$ and $h_y = a_y = u_y$. In this case all velocities are normalised to the electron thermal velocity and lengths to the Debye length.

2.2. Split-Eulerian schemes

Split-Eulerian schemes have been applied extensively to the solution of Vlasov's equation. These are reviewed at some length in Refs. [20,21,19,27,26].

In a split-Eulerian scheme [18], the Vlasov solver is split into separate spatial and momentum space updates. The 2D2P Vlasov equation is split into the following 1D advections:

$$\frac{\partial f_e}{\partial t} + h_x \frac{\partial f_e}{\partial x} = 0 \tag{19}$$

$$\frac{\partial f_e}{\partial t} + h_y \frac{\partial f_e}{\partial y} = 0 \tag{20}$$

$$\frac{\partial f_e}{\partial t} + (E_x + a_y B_z) \frac{\partial f_e}{\partial u_x} = 0 \tag{21}$$

$$\frac{\partial f_e}{\partial t} + (E_y - a_x B_z) \frac{\partial f_e}{\partial u_y} = 0 \tag{22}$$

The splitting scheme for the advance of the 2D2P Vlasov–Maxwell system from step k to step $k+1$ is then:

1. x advection: Evolve Eq. (19) for $\Delta t/2$ (from k to $k+1/2$);
2. y advection: Evolve Eq. (20) for $\Delta t/2$ (from k to $k+1/2$);
3. u_x advection: Evolve Eq. (21) for $\Delta t/2$ (from k to $k+1/2$);
4. u_y advection: Evolve Eq. (22) for Δt (from k to $k+1$);
5. u_x advection: Evolve Eq. (21) for $\Delta t/2$ (from $k+1/2$ to $k+1$);
6. y advection: Evolve Eq. (20) for $\Delta t/2$ (from $k+1/2$ to $k+1$);
7. x advection: Evolve Eq. (19) for $\Delta t/2$ (from $k+1/2$ to $k+1$);

where the values of $E_{x,\,y}$ and B_z used in the momentum space advections have to be time and space centred. This requires them to be averaged onto the centre of the cell. The $x-y$ ordering is reversed on alternate steps. The timestep size, Δt is determined by the CFL condition.

There are alternative approaches, most noticably the semi-Lagrangian [22,41] and back-substitution methods [24]. The semi-Lagrangian approach was shown to be unstable in the relativistic case [35]. The adoption of 2D velocity advections corrects this and reduces the total number of advections required. This has the added benefit that the circular orbits of particles in strong magnetic fields can be traced accurately on Cartesian velocity grids. This is also the case for the back substitution method. However, these methods typically require high-order spline interpolation schemes and cannot guarantee positivity. By solving Vlasov's equation in explicit conservative form, we can avoid the instabilities inherent in the standard semi-Lagrangian scheme. Describing circular particle orbits via a series of 1D advections does present a potential source of cumulative error, as has been highlighted in the literature [24,1]. However, we find that the scheme described here is able to resolve circular particle orbits and accurately maintain the phase and amplitude of the momentum rotor with some resolution-dependent numerical dissipation, which is comparable to that experienced over the same number of linear, parallel, advection steps.

The time-splitting algorithm is second order in time and each of the distribution function updates is a 1D advection equation of the form

$$\frac{\partial U}{\partial t} + c \frac{\partial U}{\partial x} = 0 \tag{23}$$

There are many ways one could conduct the 1D advections at the heart of the time-splitting algorithm; examples include the use of MacCormak's method [42], spline interpolation [43] and a variety of conservative schemes [20].

Any scheme must exclude the introduction of false extrema and should not accentuate existing extrema-it must be monotonicity preserving and bounded. More importantly, the fine scale filamentation of the distribution function must be treated correctly; in a manner that is physically appropriate. The simplest example of the problem of filamentation is linear Landau damping, whereby the perturbation of the distribution function evolves as $\sim\exp(ikvt)$. Hence, for any grid, there

will, eventually, be structure at the grid scale. This must be dissipated numerically. However, Ref. [21] shows that it is not enough that the scheme dissipates this fine scale structure; it must do so in a way that is physically consistent. The most successful schemes manage to dissipate fine scale filamentation in a manner that is entropy increasing.

A robust method for performing these updates, in a split-Eulerian scheme, in conservative form, is PPM [30]. This scheme is formally third order (away from extrema), positivity and monotonicity preserving and dissipates grid-scale structure in an entropy increasing manner without the need for additional filtering. We find that casting the problem in conservative form and adopting a PPM treatment for the advections proves to be robust, scaleable and free from some of the complexities of alternative approaches.

2.3. Updating the fields

VALIS requires an accurate, stable and scalable algorithm for the solution of Maxwell's equations (i.e. Eqs. (10)–(12)) in order to provide the time-centred self-consistent fields used in the momentum space advections (Eqs. (21) and (22)). Such schemes are common in computational plasma physics [12] and engineering [40]. They can potentially suffer from a cumulative error in the solution to Poisson's equation which must be corrected at each step or removed intermittently-this should be achieved without either incurring significant computational cost, or altering the dispersion relation of waves in the system (possibly resulting in numerical Cherenkov instabilities [33]).

The most common scheme is Leapfrog, where E and B fields are advanced for a full timestep at a time but remain half a timestep out of phase. An alternative is to use a Predictor Corrector scheme, which has the advantage that it can, if desired, re-cycle the particle fluxes calculated by the PPM advections during the corrector step, which ensures the solution satisfies to Poisson's equation-this is the approach adopted for VALIS. These schemes are outlined below.

2.3.1. Leapfrog

In a leapfrog scheme (see Fig. 2 and, for example, Ref. [12]) the electric and magnetic fields are one half timestep out of phase so that one is able to provide the time-centred values to advance the other. At step k, in cell (i,j), we have $E_x^{k-1/2}(i,j), E_y^{k-1/2}(i,j)$ and $B_z^k(i,j)$, defined as shown in Fig. 1. These must be advanced to $E_x^{k+1/2}(i,j), E_y^{k+1/2}(i,j), B_z^{k+1}(i,j)$ and provide $B_z^{k+1/2}(i,j)$ for the distribution function update. First the current densities need to be integrated onto the spatial grid, these are denoted by $\tilde{J}_x^k(i,j)$ and $\tilde{J}_y^k(i,j)$. The number of cells in (x,y,u_x,u_y) is given by $(n_x,n_y,n_{u_x},n_{u_y})$ and

$$\tilde{J}_x^k(i,j) = -\sum_{m=1}^{n_{u_x}}\sum_{n=1}^{n_{u_y}}\left(h_x(m,n)f_e^k(i,j,m,n)\Delta u_x(m)\Delta u_y(n)\right) \tag{24}$$

$$\tilde{J}_y^k(i,j) = -\sum_{m=1}^{n_{u_x}}\sum_{n=1}^{n_{u_y}}\left(h_y(m,n)f_e^k(i,j,m,n)\Delta u_x(m)\Delta u_y(n)\right) \tag{25}$$

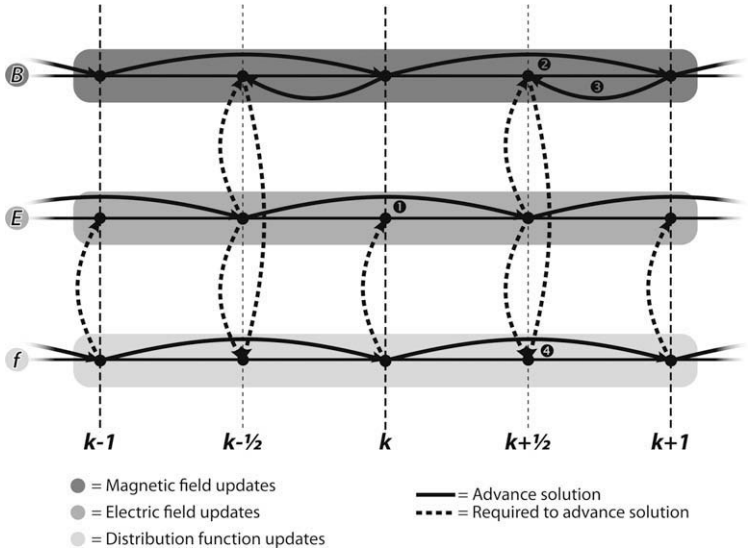

Fig. 2. Time-stepping algorithm with Leapfrog field updates. (1) Integrate currents onto the simulation grid and then average to cell faces. Use these currents, and B_z^k, to update $E_{x,y}^{k-1/2}$ to $E_{x,y}^{k+1/2}$. (2) Advance magnetic field from B_z^k to B_z^{k+1} using time-centred electric fields, $E_{x,y}^{k+1/2}$. (3) Calculate time-centred magnetic field $B_z^{k+1/2} = 0.5(B_z^k + B_z^{k+1})$. (4) Advance distribution function by a complete timestep using time-centred fields.

To advance Maxwell's equations, currents are required at the cell faces so a linear interpolation is applied:

$$J_x^k(i,j) = 0.5\left(\tilde{J}_x^k(i,j) + \tilde{J}_x^k(i+1,j)\right) \tag{26}$$

$$J_y^k(i,j) = 0.5\left(\tilde{J}_y^k(i,j) + \tilde{J}_y^k(i,j+1)\right) \tag{27}$$

and the electric field components can be advanced by a complete timestep:

$$E_x^{k+1/2}(i,j) = E_x^{k-1/2}(i,j) + \frac{\Delta t}{\Delta y}\left(B_z^k(i,j) - B_z^k(i,j-1)\right) - \Delta t J_x^k(i,j) \tag{28}$$

$$E_y^{k+1/2}(i,j) = E_y^{k-1/2}(i,j) - \frac{\Delta t}{\Delta x}\left(B_z^k(i,j) - B_z^k(i-1,j)\right) - \Delta t J_y^k(i,j) \tag{29}$$

followed by the magnetic field:

$$B_z^{k+1}(i,j) = B_z^k(i,j) - \frac{\Delta t}{\Delta x}\left(E_y^{k+1/2}(i+1,j) - E_y^{k+1/2}(i,j)\right) + \frac{\Delta t}{\Delta y}\left(E_y^{k+1/2}(i,j+1) - E_x^{k+1/2}(i,j)\right) \tag{30}$$

$$B_z^{k+1/2}(i,j) = \frac{1}{2}\left(B_z^k(i,j) + B_z^{k+1}(i,j)\right) \tag{31}$$

All that remains is to advance the distribution function using the time-centred fields.

2.3.2. Predictor corrector

The `Predictor Corrector` field update scheme (Fig. 3) starts by updating the magnetic field from $k - 1/2$ to $k + 1/2$, and interpolating back to k. Then, using B_z^k and $J_{x,y}^k$ given by Eqs. (24)–(27) apply a first-order predictor to give $E_{x,y}^{k+1/2}$:

$$E_x^{k+1/2}(i,j) = E_x^k(i,j) + \frac{\Delta t}{2\Delta y}\left(B_z^k(i,j) - B_z^k(i,j-1)\right) - \frac{\Delta t}{2} J_x^k(i,j) \tag{32}$$

$$E_y^{k+1/2}(i,j) = E_y^k(i,j) - \frac{\Delta t}{2\Delta x}\left(B_z^k(i,j) - B_z^k(i-1,j)\right) - \frac{\Delta t}{2} J_y^k(i,j) \tag{33}$$

The distribution function can then be advanced for a complete timestep using $B_z^{k+1/2}$ and $E_{x,y}^{k+1/2}$.

Finally we apply a second order corrector to the electric field using time-centred currents given by:

$$J_x^{k+1/2}(i,j) = 0.5\left(J_x^k(i,j) + J_x^{k+1}(i,j)\right) \tag{34}$$

$$J_y^{k+1/2}(i,j) = 0.5\left(J_y^k(i,j) + J_y^{k+1}(i,j)\right) \tag{35}$$

Where $J_{x,y}^{k+1}(i,j)$ are calculated using the updated distribution function at step $k + 1$. Now $E_{x,\,y}$ can be advanced from k to $k + 1$:

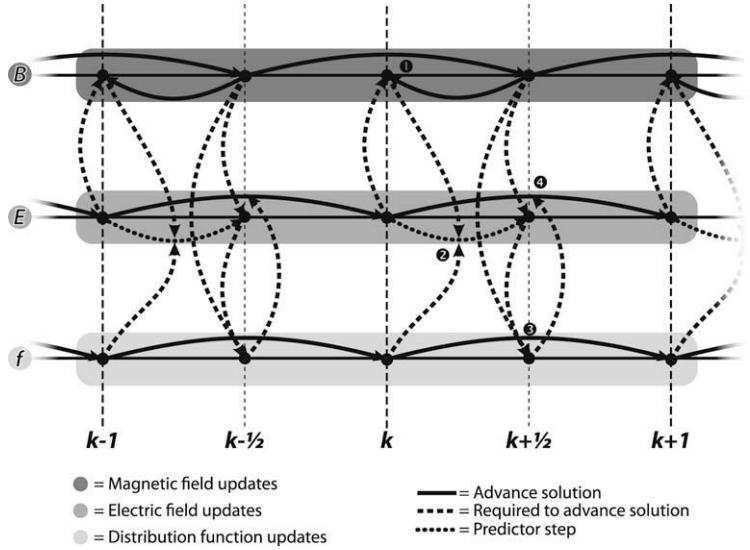

Fig. 3. Time-stepping algorithm with `Predictor Corrector` field updates. (1) Advance magnetic field from $B_z^{k-1/2}$ to $B_z^{k+1/2}$ using $E_{x,y}^k$. (2) First-order predictor for $E_{x,y}^{k+1/2}$ using face-centred grid currents at k and fields B_z^k and $E_{x,y}^k$. (3) Advance distribution function by a complete timestep using time-centred fields. (4) Electric field corrector step; advance electric field from $E_{x,y}^k$ to $E_{x,y}^{k+1}$ using $B_z^{k+1/2}$ and time-centred currents. The `VALIS` algorithm follows the same steps, except that the currents used by the corrector are derived from the exact fluxes calculated during the spatial distribution function advections – this ensures that Poisson's equation is satisfied to round-off.

$$E_x^{k+1}(i,j) = E_x^k(i,j) + \frac{\Delta t}{\Delta y}\left(B_z^{k+1/2}(i,j) - B_z^{k+1/2}(i,j-1)\right) - \Delta t J_x^{k+1/2}(i,j) \tag{36}$$

$$E_y^{k+1}(i,j) = E_y^k(i,j) - \frac{\Delta t}{\Delta x}\left(B_z^{k+1/2}(i,j) - B_z^{k+1/2}(i-1,j)\right) - \Delta t J_y^{k+1/2}(i,j) \tag{37}$$

2.3.3. VALIS

In PIC based kinetic models it is common to calculate the total charge density and current at grid points by interpolating the charge and current density from individual particles. It is however well known [12,34] that the time evolution of these charge (ρ) and current (J) densities in the code do not satisfy the finite difference version of charge conservation $\partial_t \rho = -\nabla \cdot J$. Note that while they do not satisfy this equation charge is still conserved, it is the interpolated currents which are inconsistent with the charge conservation equation. As a result such schemes do not satisfy Poisson's equation, even if this is imposed at the start of the simulation. The solution for PIC based schemes is to not interpolate the currents to the grid points but instead calculate the charge that crosses the cell boundaries [34]. This all also applies to direct Vlasov solvers. If the charge and current densities are calculated at grid points by integrating up the distribution function there is no guarantee that these satisfy charge conservation or Poisson's equation. Instead the current which is used in Maxwell's equation's must be the calculated from the flux of charge through computational cell edges. Since these fluxes are exactly the fluxes which were used to update the distribution function in each cell they must satisfy the charge conservation equation at the finite difference level. Hence Poisson's equation must be satisfied to machine precision.

During the spatial advections in the `Predictor Corrector` approach, the fluxes through each cell face are calculated as part of the PPM algorithm. These can be integrated up and used to calculate the average current over the timestep. For cell (i,j), we define $J_{x,y}^{k+1/2}$ as

$$J_x^{k+1/2}(i,j) = -\frac{1}{\Delta t}\sum_{m=1}^{n_{ux}}\sum_{n=1}^{n_{uy}}\left(h_x(m,n)\left[\Delta t \Phi_x^{k+1}(i,j,m,n)\right]\Delta u_x(m)\Delta u_y(n)\right) \tag{38}$$

$$J_y^{k+1/2}(i,j) = -\frac{1}{\Delta t}\sum_{m=1}^{n_{ux}}\sum_{n=1}^{n_{uy}}\left(h_y(m,n)\left[\Delta t \Phi_y^{k+1}(i,j,m,n)\right]\Delta u_x(m)\Delta u_y(n)\right) \tag{39}$$

where $\Phi_{x,y}^{k+1}(i,j,m,n)$ is the flux through the far boundary for cell (i,j,m,n) (right-hand in the case of Φ_x^{k+1}, top in the case of Φ_y^{k+1}), as calculated by the PPM routine during the advance of the distribution function from k to $k+1$. These currents can be used for the time-centred currents required by the corrector step in the `Predictor Corrector` algorithm, rather than integrating the currents onto the spatial grid again at $k+1$. This is the method used in `VALIS`. By performing a corrector step with the currents from the advection, we ensure that the Poisson's equation is obeyed to machine precision, without the additional effort of solving it directly. The approach of using the total current actually moved during advection to update the electric field is the same as that previously devised for PIC codes [34]. This is the first time that this scheme has been applied in a direct Vlasov solver and for an Eulerian split Vlasov scheme this comes at no additional computational cost.

3. Particle gyromotion with Eulerian momentum grids

A simple test of the split-Eulerian scheme is to model the gyration of charged particles about a fixed magnetic field. The particles are represented by a Gaussian distribution of equal width in u_x and u_y but some initial velocity offset. At this stage we do not consider any spatial extent and perform no field updates, fixing $E_x = E_y = 0$ and $B_z = 1$ for the duration. We also take the non-relativistic limit but fixing $\gamma = 1$ and normalising all velocities to the electron thermal velocity v_{Te} rather than c, the velocity grid extends to $\pm 8 v_{Te}$ in both directions. Under these conditions, the distribution should orbit about $(u_x, u_y) = (0,0)$ with a period of 2π, the timestep is fixed at $\Delta t = 2\pi/8n_u$ where $n_u = n_{ux} = n_{uy}$ is the grid resolution.

As shown in Fig. 4, the split advection scheme is able to preserve both the phase and total momentum of the particle gyromotion. The initial Gaussian is poorly resolved and dissipates considerably over the first few gyrations. This is accompanied by a significant fall in the maxima of the distribution. This clipping of smooth, but poorly resolved, extrema is a direct result of the first-order representation used in PPM at extrema which results from the limiters chosen to ensure the solution is total variation diminishing. This can be addressed [44], but at the expense of positivity.

As shown in Fig. 5, increasing the resolution to 256 by 256 cells in velocity space shows a considerable reduction in the numerical dissipation. A series of simple parallel advections over the same number of cells shows a comparable level of dissipation. We can quantify how the error scales with resolution by comparing the L_1 norm on the variation of f_e from its initial value:

$$L_1(k) = \sum_{m,n}\frac{|f(t=0) - f(t=k\times 2\pi)|}{n_{ux}n_{uy}} \tag{40}$$

where $k \in \mathbb{N}$ is the orbit number. Results are shown in Fig. 6, and clearly demonstrate that the error falls as n_u^{-2} where $n_u = n_{ux} = n_{uy}$.

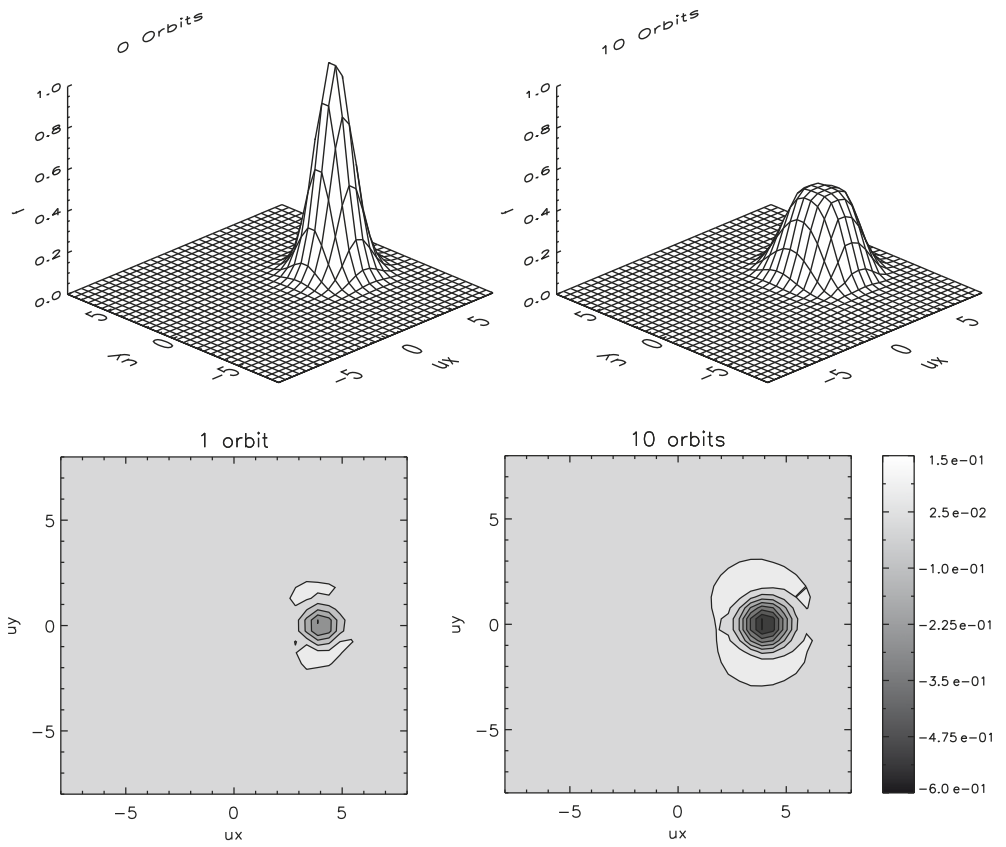

Fig. 4. Particle gyromotion test on 32 by 32 cell velocity grid. Initial particle distribution is a Gaussian with $T = 1$ and an initial momentum offset of $(u_x, u_y) = (4, 0)$. Top left: initial particle distribution. Top right: particle distribution after ten orbits. Bottom left: fractional change in the distribution function after one orbit. Bottom right: fractional change in the distribution after ten orbits. Numerical dissipation and 'clipping' of the extrema (a result of the limiters employed in the PPM steps) is evident, but the phase and total momentum of the orbiting distribution is conserved.

By performing these circular orbits in the form of interleaved 1D advections, there is a danger that the cumulative error at each step will quickly result in a phase or momentum error [24]. Performing the momentum update as a 2D advection (working backwards along the circular characteristic at each point) [22] or adopting polar phase-space co-ordinates [31] can avoid this potential pitfall. However, 2D advection schemes bring added complexity and cannot guarantee positivity, while polar co-ordinates are not suitable for resolving particle motion in strong electromagnetic fields. These tests demonstrate that the explicit, conservative, split-Eulerian scheme is capable of preserving momentum and phase, with some grid-dependent numerical dissipation.

4. Wave dispersion and the solution of Poisson's equation

The code's ability to reproduce the linear dispersion relations for electrostatic and electromagnetic waves can be tested by perturbing a uniform, periodic system sinusoidally at various ω and k. Reconstructing the dispersion curve in this way is potentially time consuming and does not stress the model in a realistic manner. A much simpler method is to perturb all wave numbers simultaneously and compare the Fourier transform of the resulting electric field time-series data to the known dispersion relations. This method is applied to test both the Langmuir and electromagnetic dispersion relations. The various algorithms are able to reproduce the dispersion relations correctly, for wavelengths which are not close to the Nyquist wavelength, as one would expect. We find that all schemes are stable, but that `VALIS` is able to correctly solve Poisson's equation, to round-off, without altering the dispersion of electrostatic or electromagentic modes.

4.1. Langmuir dispersion relation

The linear dispersion relation for electron plasma waves (in one-dimension) is

$$\omega^2 = \omega_{pe}^2 + 3v_{Te}^2 k^2. \tag{41}$$

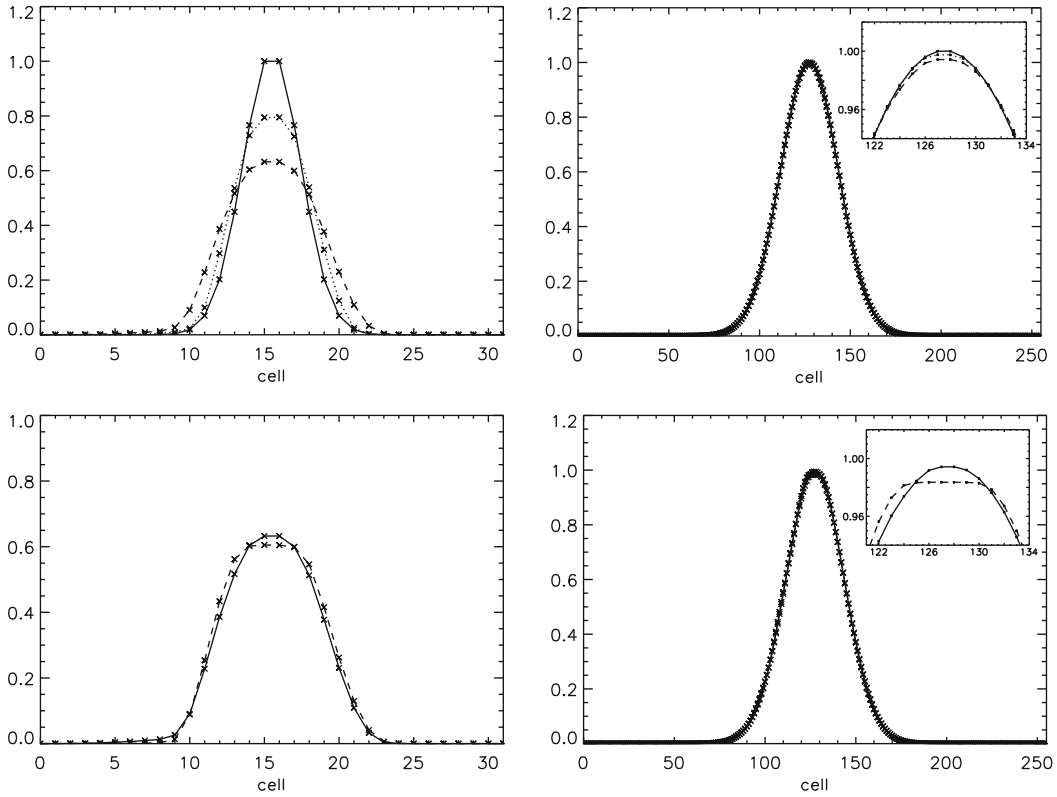

Fig. 5. Top: particle gyromotion test on 32 by 32 (left) and 256 by 256 cell velocity grids (right), initially (solid line) and after two (dotted line) and ten (dashed line) orbits (i.e. $t = 2 \times 2\pi$ and $t = 10 \times 2\pi$). Bottom: comparison of 2D gyromotion (solid line) and linear advection (dashed line) at $t = 10 \times 2\ \pi$ on 32 by 32 (left) and 256 by 256 cell velocity grids (right). Initial particle distribution is a Gaussian with $T = 1$ and an initial momentum offset of (u_x, u_y)=(4,0).

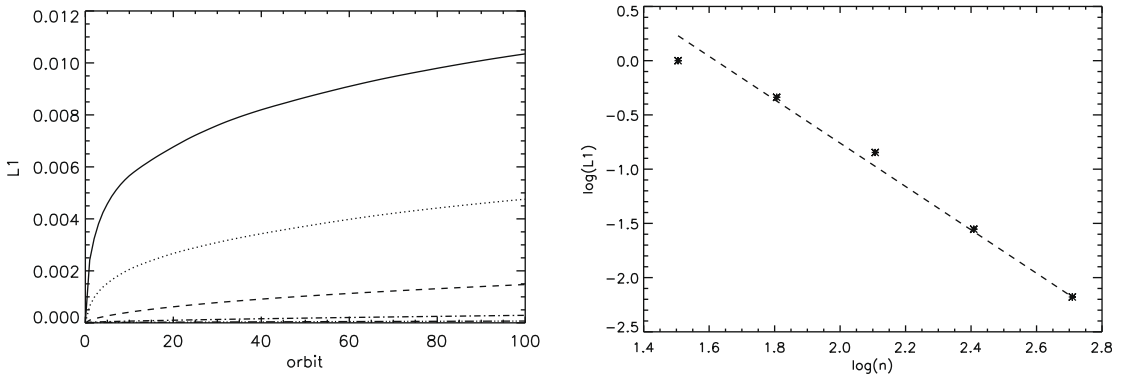

Fig. 6. L_1 norm (Eq. (40)) on variation of f from its initial value during the gyromotion test. Left: The value of $\ln(L_1)$ against number of orbits for the case of: 32 by 32 cells (solid line); 64 by 64 cells (dotted line); 128 by 128 cells (dashed line); 256 by 256 cells (dot-dashed line); and 512 by 512 cells (double dot-dashed line). Right: The value of ln (L_1) (normalised to the 32 by 32 cell case) against ln (n), where $n = n_{ux} = n_{uy}$, after 100 orbits. The numerical dissipation scales with n^{-2} (dashed line).

We consider the non-relativistic limit and start with a periodic system of length $512\lambda_D$ and extending to $\pm 4.5 v_{Te}$ in u on a grid of $(n_x, n_y, n_{ux}, n_{uy})$=(256,8,32,32). The extent of the system in y is incidental, the y and u_y updates can be disabled without affecting the problem (similarly the problem can be cast in the y-direction and x updates disabled).

Each harmonic of the longitudinal electric field is perturbed sinusoidally with equal amplitude (0.01) and a randomised phase. In order to satisfy Poisson's equation initially, the local density at each point is calculated from the electric field using $n_e(i,j) = 1 - (E_x(i,j) - E_x(i-1,j))/\Delta x$, ion density is, by implication, uniform. When the simulation reaches $t = 256/\omega_{pe}$ the dispersion relation is calculated by taking the Fourier transform of $E_x(x,t)$ to give $\widehat{E}_x(k,\omega)$. Plotting the location of the maxima of $\widehat{E}_x(k,\omega)$ in (ω,k) space generates the dispersion relation, as shown in Fig. 7. In Fig. 7 the procedure outlined above gives zero

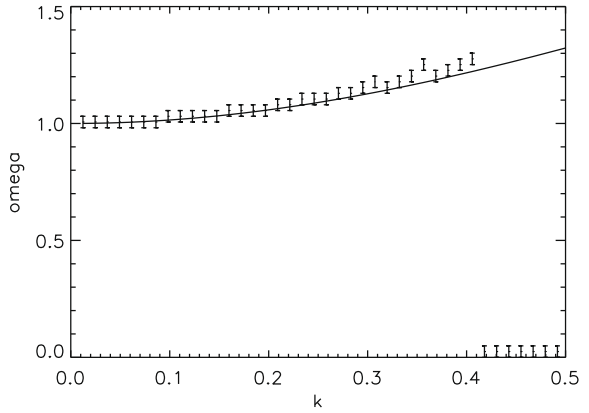

Fig. 7. Comparison of analytic Langmuir dispersion relation (solid curve) with that calculated by VALIS. Error bars represent the frequency space resolution afforded by the time-series data from VALIS. The frequency ω is in units of the electron plasma frequency ω_{pe} and k is in units of $2\pi/\lambda_D$.

frequency for $\omega > 0.42$. This is because these modes are so strongly Landau damped that the 2D FFT spectrum does not have a clear maximum when taken over the full simulation runtime. If instead ω is estimated from the zero crossings of E_x at a fixed point in space then $\omega = 1.31$ for $k = 0.5$ and indeed the correct frequency is obtained all the way up to the Nyquist frequency at $k = 0.785$. These dispersion properties are reproduced by all three approaches to the field update (i.e. `Leapfrog`, `Predictor Corrector` and VALIS).

4.2. Electromagnetic dispersion relation

The linear dispersion relation for electromagnetic waves in a plasma is

$$\omega^2 = \omega_{pe}^2 + c^2k^2. \tag{42}$$

The transverse electric and magnetic fields are subjected to a low amplitude perturbation of all wave numbers in the same manner as the electrostatic dispersion test. In this case the system is relativistic with $v_{Te} = \sqrt{0.001}c$. The system length in the direction of perturbation is $256c/\omega_{pe}$ and the momentum grids extend to $\pm 1.2 m_e c$. Updates in the transverse spatial dimension can be disabled, but care must be taken to ensure the transverse currents are still accurately computed. The same grid resolution as the Langmuir dispersion test is used here, and the simulation end time is $t = 256/\omega_{pe}$.

Fig. 8 shows a comparison of the dispersion relation derived from VALIS and the analytic form (Eq. (42)). Agreement at low k is good but there is a growing deviation as k approaches the Nyquist wavelength at $k = \pi$. At $k \approx 0.79$, where there are eight cells per wavelength (a poor, but acceptable, sinusoidal waveform) VALIS is able to correctly determine the electromagnetic dispersion. Again, these dispersion properties are reproduced by all three approaches to the field update (i.e. `Leapfrog`, `Predictor Corrector` and VALIS).

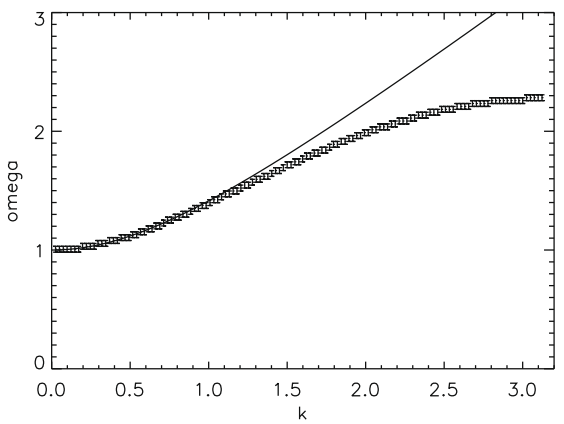

Fig. 8. Comparison of analytic EM dispersion relation (solid line) with that calculated by VALIS. Error bars represent the frequency space resolution afforded by the time-series data from VALIS. The frequency ω is in units of the electron plasma frequency ω_{pe} and k is in units of $2\pi\omega_{pe}/c$.

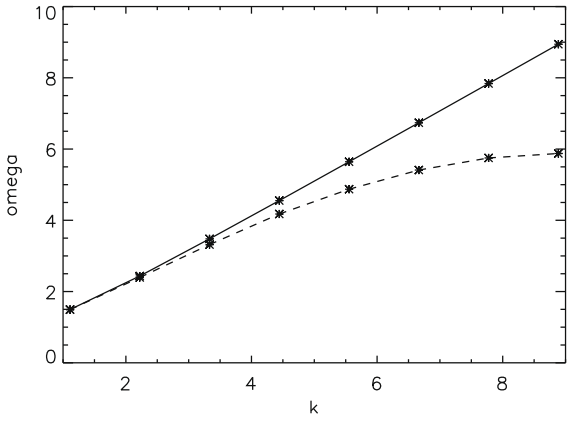

Fig. 9. Comparison of the analytic EM dispersion relation (solid line) with that calculated by VALIS for EM waves propagating diagonally across a periodic domain. The frequency ω is in units of the electron plasma frequency ω_{pe} and k is in units of ω_{pe}/c. For points from the VALIS code the error in the values of ω is ±0.005.

To compare further the properties of the various time-stepping schemes each scheme was tested with 2D electromagnetic waves. In these tests $v_{Te} = \sqrt{0.001}c$, the momentum grid extends from $\pm 0.4 m_e c$ and the spatial grid is periodic in x and y both of which were of length $8c/\omega_{pe}$. The resolution was fixed at $(n_x, n_y, n_{u_x}, n_{u_y}) = (16, 16, 32, 32)$, the timestep was fixed at $\Delta t = 0.16 c/\omega_{pe}$ and the initial equilibrium was perturbed with an electromagnetic wave, of amplitude 0.01 in normalised units, propagating diagonally across the domain. Thus the minimum normalised wave-number possible was $k = 1.11$ and the maximum, i.e. Nyquist, wave-number was $k = 8.886$. This was therefore both a test of the schemes for high wave-number EM wave propagation and the ability of the schemes to move EM waves diagonally across the domain using the split-Eulerian scheme in phase space. Fig. 9 compares the analytic dispersion relation and the result for the VALIS algorithm from $k = 1.11$ up to the Nyquist wave-number. These results were reproduced with both the Leapfrog and Predictor Corrector schemes. The greatest discrepancy in the calculated value of ω between the schemes was 5% over the whole range tested. Thus moving from the Leapfrog scheme to Predictor Corrector does not seriously affect the properties of EM wave propagation and furthermore using VALIS, i.e. correcting the errors in Poisson's equation from the Predictor Corrector scheme, does not lead to additional slowing of EM waves for short wavelengths. This is different to similar tests using PIC codes, where correcting the error in Poisson's equation by using a 'Poisson correction' step at times through the simulation must be performed [32]. A similar technique as employed here, i.e. use of the actual transported current through cell boundaries to update the electric field, has been used for PIC codes [34], although the implementation for the Vlasov solver in split-Eulerian form is easier as the current fluxes are calculated anyway for the update of f_e. Thus, while techniques exist to exactly enforce Poisson's equation in PIC codes [34] in the Vlasov formulation described here for VALIS this can be done with no additional work, while also avoiding problems associated with Cherenkov radiation at short wavelengths.

5. Raman scattering example

The properties of the particle and field updates have been studied in some detail. By way of an example of the scheme in action, we consider the Raman scattering problem outlined in Ref. [43]. However, we initialise the problem diagonally across the box to demonstrate the use of the full 4D phase space.

The system is initialised with an EM parent wave and a density perturbation to seed the growth of the Langmuir daughter wave. The parent and seed Langmuir daughter waves are at 45° to the x-axis. The system is periodic and the parent EM wave is not externally driven.

The parent EM wave, EM1, has amplitude $E_{E1} = 0.28$, the density perturbation at $k = k_L$ has amplitude $\delta n = 0.01$. The three-wave system is completed by a backscattered EM wave, EM2.

The amplitudes of the incident, reflected and Langmuir modes show the reflected EM wave growing at the expense of the pump wave as a result of scattering from the Langmuir wave (see Fig. 10). The process creates a feed-back loop and is unstable with the two daughter waves initially growing exponentially. This growth is bounded and the system settles into an oscillatory phase after $t \approx 250$. During the saturation of the instability, energy is transfered from the electrostatic field to the electrons as they are accelerated and trapped by the wave. This is manifested as a series of 'buckets' in the electron distribution function (see Fig. 11).

The results outlined above are in close agreement with those in Ref. [43]. This problem demonstrates some of the key physics issues in laser–plasma interaction (such as particle trapping and hot electron production) and highlights the strengths of the Eulerian Vlasov solver, in particular the ability to resolve the particle phase space to high resolution even in regions of low particle density. There are minor differences during the saturation of the instabilities; these can be

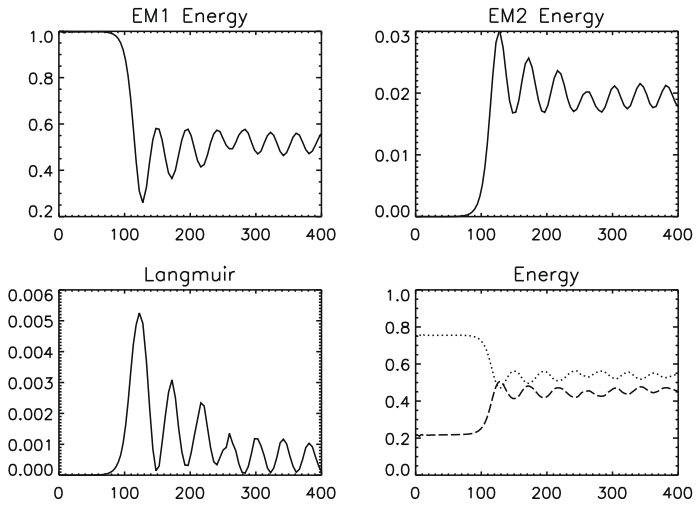

Fig. 10. Energy partition between the parent EM wave (EM1, top left) and the daughter EM (EM2, top right) and Langmuir (bottom left) waves in a 2D SRS test problem. In this case the system is initialised with all waves travelling diagonally across the simulation domain. As in the 1D problem, the Daughter EM and Langmuir waves grow at the expense of the parent wave until the system saturates and begins to oscillate about an equilibrium energy partition. The bottom right figure shows the partition of energy between electric (dotted) and kinetic (dashed). The Landau damping of the daughter wave and the trapping of electrons results in a transfer of energy from the electric fields to the electrons.

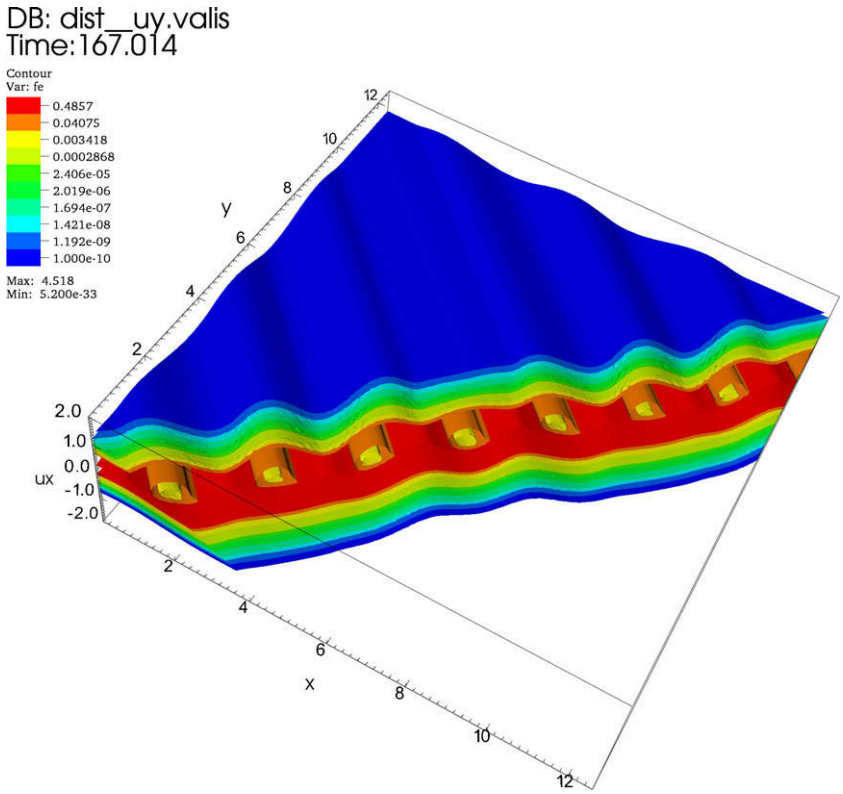

Fig. 11. Isocontour rendering of the electron distribution function $f_e(x,y,u_x,u_y)$ integrated over u_y showing electron trapping during the saturation of the Raman scattering instability.

attributed to the effect of the transverse electron distribution (not present in previous work). Heating in this direction (numerical or otherwise) will detune the instability and alter its evolution slightly. Furthermore, previous work did not include relativistic effects in the transverse motion of the plasma. At the wave amplitudes considered here, these can be expected to be minor but will nevertheless have some impact on the evolution of the system.

6. Scaling on massively parallel machines

The explicit Eulerian algorithm adopted for VALIS can be parallelised via domain decomposition over some or all of the four dimensions in the system (although there is some additional complexity and communication overhead in parallelising over the momentum dimensions, since some non nearest-neighbour operations are required to calculate currents and densities for a given point in physical space). VALIS is domain decomposed over the two spatial dimensions, this involves the exchange of ghost values for f_e,$E_{x,y}$ and B_z along each processing element (PE) boundary. A simulation domain of size $(n_x, n_y, n_{ux}, n_{uy})$ is divided up onto n_{pe} processing elements, where $n_{pe} = n_{pex} \times n_{pey}$, in local domains of size $(n_x/n_{pex}, n_y/n_{pey}, n_{ux}, n_{uy})$. Therefore, each processing element must advance the fields and distribution functions for a fixed sub-set of the physical space and the entire momentum grid in this region. Communication between processing elements is carried out using the standard Message Passing Interface (MPI) libraries for simplicity and portability.

A simple linear Landau damping problem (see, for example, Ref. [21]) was used to test the scaling properties of the decomposition scheme. We consider the non-relativistic, electrostatic limit. The initial velocity distribution is Maxwellian and we consider the damping of a wave parallel to the x-axis. The system extends from -2π to 2π in x and $-4.5v_{Te}$ to $4.5v_{Te}$ in u_x. For this the extent of the y and u_y are set to match x and u_x respectively. We perturb the $k = 0.5$ mode with a low amplitude density perturbation of the form $n_1 = \alpha\cos(kx)$, where $\alpha = 0.01$. Under these conditions, the linear Landau damping rate is $\Gamma_L = 0.153359$.

The scaling tests started with a phase-space grid of $(n_x, n_y, n_{u_x}, n_{u_y}) = (64, 64, 32, 32)$, on one processing element, with all IO disabled and the run limited to 100 timesteps. Tests were completed on a Cray XT3 system based on AMD's 2.6 Ghz dual-core Opteron processor. As the number of processing elements is doubled (alternately in x and y), so is the extent of the phase-

Table 1
Scaling performance of domain-decomposition scheme tested using the linear Landau damping problem. As the number of processors increases, so too do the dimensions of the system and the phase-space, so that the domain allocated to each processor is identical irrespective of the total number of PEs involved. This allows the fairest possible comparison, for perfect scaling the runtimes would be identical. A linear fit through the values for $n_{pe} = 2$–1024 (see Fig. 12) indicates near perfect scaling. The 'cost' of doubling the number of processors is approximately a 0.3% slow-down.

Number of PEs	Runtime (s)	% Change
1	567.313	0.0
2	581.243	2.455
4	578.124	1.906
8	582.237	2.630
16	589.830	3.969
32	586.583	3.397
64	589.821	3.967
128	589.529	3.916
256	590.132	4.022
512	594.065	4.716
1024	592.677	4.471

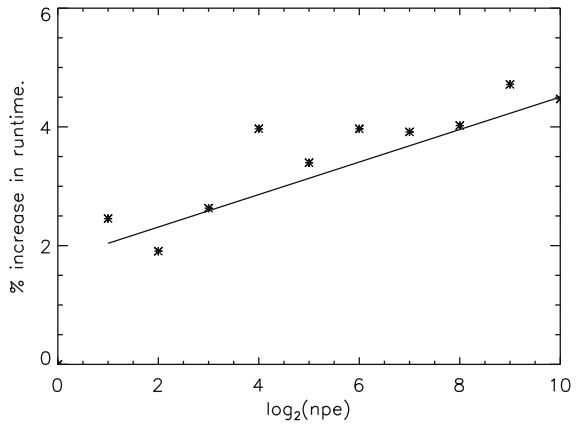

Fig. 12. The linear Landau damping problem was used to test the scaling properties of the decomposition scheme. The percentage variation in runtime is shown against $\log_2(n_{pe})$, where n_{pe} is the number of processing elements (PEs). Results are relative to a single PE with a phase-space grid of $(n_x, n_y, n_{u_x}, n_{u_y}) = (64, 64, 32, 32)$ and all IO disabled. As the number of processors increases, so too do the dimensions of the system and the phase-space, so that the domain allocated to each processor is identical, irrespective of the total number of PEs involved. This allows the fairest possible comparison, for perfect scaling the runtimes would be identical. A linear fit through the points for $n_{pe} = 2$–1024, shown as a solid line, indicates near perfect scaling. The 'cost' of doubling the number of processors is approximately a 0.3% slow-down.

space grid, the system dimensions, and the perturbed wave-number, so that the domain allocated to each processor is identical, irrespective of the total number of processing elements involved. This allows the fairest possible comparison, if the scaling were perfect then the runtimes would be identical. Final runtimes (see Table 1) varied from 9′27.313″ (for a single processing element) to 9′54.065″ (for 512 processing elements, with the 1024 processing element case being slightly quicker at 9′52.677″). The percentage variation in runtime (relative to the single processing element case) is shown in Fig. 12. These results indicate that the 'cost' of doubling the number of PEs (beyond two) is a slow-down of less than 0.3% in comparison to perfect linear scaling.

7. Conclusion

The VALIS code for the numerical solution of the 2D2P Vlasov–Maxwell system has been described in detail. VALIS adopts a conservative, split Eulerian scheme in which the particle distribution function is split into 1D advections carried out using Peasewise Parabolic Method. This results in a scheme which is not vulnerable to the time-splitting instability encountered by semi-Lagrangian schemes and is able to resolve particle gyromotion on Eulerian momentum grids. Furthermore, the scheme is readily parallelised, by domain decomposition, and near perfect scaling has been demonstrated up to 1024 processing elements. Utilising the exact particle fluxes calculated during the PPM steps to calculate the current in the solution of Maxwell's equations ensures that the advance of Maxwell's equations satisfies Poisson's equation without introducing additional workload or altering the dispersion properties of the solver.

Acknowledgements

This work was partly supported by the AWE Technical Innovation Fund (ATIF) and computing resources made available through the University of Warwick's Centre for Scientific Computing. British Crown Copyright 2009 / MOD.

References

[1] A.R. Bell, A.P.L. Robinson, M. Sherlock, R.J. Kingham, W. Rozmus, Fast electron transport in laser-produced plasmas and the KALOS code for solution of the Vlasov–Fokker–Plank equation, Plasma Phys. Control. Fusion 48 (2006) R37.
[2] A.J. Kemp, Y. Sentoku, V. Sotnikov, S.C. Wilks, Collisional relaxation of superthermal electrons generated by relativistic laser pulses in dense plasma, Phys. Rev. Lett. 97 (2006) 235001.
[3] H. Ruhl, P. Mulser, Relativistic Vlasov simulation of intense fs laser pulse–matter interaction, Phys. Lett. A 205 (1995) 388.
[4] N.J. Sircombe, R.O. Dendy, T.D. Arber, Aspects of electron acoustic wave physics in laser backscatter from plasmas, Plasma Phys. Control Fusion 48 (2006) 1141.
[5] L. Yin, B.J. Albright, K.J. Bowers, W. Daughton, H.A. Rose, Saturation of backward stimulated scattering of a laser beam in the kinetic regime, Phys. Rev. Lett. 99 (2007) 265004.
[6] D. Bénisti, D.J. Strozzi, L. Gremillet, Breakdown of electrostatic prediction for the nonlinear dispersion relation of a stimulated Raman scattering driven plasma wave, Phys. Plasma 15 (2008) 030701.
[7] T. Tajima, J.M. Dawson, Laser electron accelerator, Phys. Rev. Lett. 43 (1979) 267.
[8] S.P.D. Mangles, C.D. Murphy, Z. Najmudin, A.G.R. Thomas, et al, Monoenergetic beams of relativistic electrons from intense laser–plasma interactions, Nature 431 (2004) 535.
[9] T.D. Arber, V.F. Melnikov, Thermal fronts in flaring magnetic loops, Astrophys. J. 690 (2009) 238.
[10] P. Petkaki, M.P. Freeman, T. Kirk, C.E.J. Watt, R.B. Horne, Anomalous resistivity and the nonlinear evolution of the ion-acoustic instability, J. Geophys. Res. 111 (2006) A01205.
[11] R.G.L. Vann, R.O. Dendy, G. Rowlands, T.D. Arber, N. d'Ambrumenil, Fully nonlinear phenomenology of the Berk–Breizman augmentation of the Vlasov–Maxwell system, Phys. Plasmas 10 (2003) 623.
[12] C.K. Birdsall, A.B. Langdon, Plasma physics via computer simulation, IoP (1991).
[13] A. Pukhov, Strong field interaction of laser radiation, Rep. Prog. Phys. 66 (2003) 47.
[14] K.J. Bowers, B.J. Albright, L. Yin, B. Bergen, T.J.T. Kwan, Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation, Phys. Plasma 15 (2008) 055703.
[15] N. Besse, G. Latu, A. Ghizzo, E. Sonnendruker, P. Bertrand, A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov–Maxwell system, J. Comput. Phys. 227 (2008) 7889.
[16] R.J. Kingham, A.R. Bell, An implicit Vlasov–Fokker–Plank code to model non-local electron transport in 2D with magnetic fields, J. Comput. Phys. 194 (2004) 1.
[17] R. Duclous, B. Dubroca, F. Filbet, V. Tikhonchuk, High-order resolution of the Maxwell Fokker–Planck–Landau model intended for ICF applications, J. Comput. Phys., submitted for publication. Available from: <arXiv 0810.4824>.
[18] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys 22 (1976) 330.
[19] A. Mangeney, F. Califano, C. Cavazzoni, P. Travnicek, A numerical scheme for the integration of the Vlasov–Maxwell system of equations, J. Comput. Phys. 179 (2002) 495.
[20] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, J. Comput. Phys. 172 (2001) 166.
[21] T.D. Arber, R.G.L. Vann, A critical comparison of Eulerian-grid-based Vlasov solvers, J. Comput. Phys. 180 (2002) 339.
[22] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, J. Comput. Phys. 149 (1999) 201.
[23] N.V. Elkina, J. Büchner, A new conservative unsplit method for the solution of the Vlasov equation, J. Comput. Phys. 213 (2006) 862.
[24] H. Schmitz, R. Grauer, Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields, Comput. Phys. Commun. 175 (2006) 86.
[25] H. Schmitz, R. Grauer, Darwin–Vlasov simulations of magnetised plasmas, J. Comput. Phys. 214 (2006) 738.
[26] M. Shoucri, Eulerian codes for the numerical solution of the Vlasov equation, Commun. Nonlinear Sci. Numer. Simul. 13 (2008) 174.
[27] E. Pohn, M. Shoucri, G. Kamelander, Eulerian Vlasov codes, Comput. Phys. Commun. 166 (2005) 81.
[28] F. Filbet, E. Sonnendrucker, Comparison of Eulerian Vlasov solvers, J. Comput. Phys. 150 (2003) 247.
[29] E. Fijalkow, A numerical solution of the Vlasov equation, Comput. Phys. Commun. 150 (1999) 319.
[30] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys 54 (1984) 174.

[31] F. Valentini, P. Veltri, A. Mangeney, A numerical scheme for the integration of the Vlasov–Poisson system of equations, in the magnetized case, J. Comput. Phys. 210 (2005) 730.
[32] A.B. Langdon, On enforcing Gauss' law in electromagnetic particle-in-cell codes, Comput. Phys. Commun. 70 (1992) 447.
[33] B.B. Godfrey, Numerical Cherenkov instabilities in electromagnetic particle codes, J. Comput. Phys. 15 (1974) 504.
[34] J. Villasenor, O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69 (1992) 306.
[35] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrücker, O. Coulaud, Instability of the time splitting scheme for the one-dimensional and relativistic Vlasov–Maxwell system, J. Comput. Phys. 185 (2003) 512.
[36] P. Bertrand, M. Albrecht-Marc, T. Réveillé, A. Ghizzo, Vlasov models for laser–plasma interaction, Transp. Theory Stat. Phys. 34 (2005) 103.
[37] N.J. Sircombe, R. Bingham, M. Sherlock, T. Mendonca, P. Norreys, Plasma heating by intense electron beams in fast ignition, Plasma Phys. Control Fusion 50 (2008) 065005.
[38] N.J. Sircombe, M.E. Dieckmann, P.K. Shukla, T.D. Arber, Stabilisation of BGK modes by relativistic effects, Astron. Astrophys. 452 (2006) 371.
[39] N.J. Sircombe, T.D. Arber, R.O. Dendy, Accelerated electron populations formed by Langmuir wave–caviton interactions, Phys. Plasmas 12 (2005) 12303.
[40] K.S. Kunz, J. Luebbers, The Finite Difference Time Domain Method for Electromagnetics, Boca Raton, London, 1993.
[41] A. Ghizzo, F. Huot, P. Bertrand, A non-periodic 2D semi-Lagrangian Vlasov code for laser–plasma interaction on parallel computer, J. Comput. Phys. 186 (2003) 47.
[42] R.B. Horne, M.O. Freeman, A new code for electrostatic simulation by numerical integration of the Vlasov and Ampère's equations using MacCormack's method, J. Comput. Phys. 171 (2001) 182.
[43] A. Ghizzo, P. Bertrand, M.M. Shoucri, T.W. Johnston, E. Fijalkow, M.R. Feix, A Vlasov code for the numerical simulation of stimulated Raman scattering, J. Comput. Phys. 90 (1990) 431.
[44] P. Colella, M.D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema, J. Comput. Phys. 227 (2008) 7069.