

On the velocity space discretization for the Vlasov–Poisson system: Comparison between implicit Hermite spectral and Particle-in-Cell methods

E. Camporeale^{a,*}, G.L. Delzanno^b, B.K. Bergen^c, J.D. Moulton^b

^a Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands

^b T-5 Applied Mathematics and Plasma Physics, Los Alamos National Laboratory, 87545 Los Alamos, NM, USA

^c CCS-7 Applied Computer Science, Los Alamos National Laboratory, 87545 Los Alamos, NM, USA

ARTICLE INFO

Article history:

Received 15 June 2015

Received in revised form

19 August 2015

Accepted 3 September 2015

Available online 11 September 2015

Keywords:

Vlasov equation

Spectral methods

Multiscale simulations

Plasma physics

ABSTRACT

We describe a spectral method for the numerical solution of the Vlasov–Poisson system where the velocity space is decomposed by means of an Hermite basis, and the configuration space is discretized via a Fourier decomposition. The novelty of our approach is an implicit time discretization that allows exact conservation of charge, momentum and energy. The computational efficiency and the cost-effectiveness of this method are compared to the fully-implicit PIC method recently introduced by Markidis and Lapenta (2011) and Chen et al. (2011). The following examples are discussed: Langmuir wave, Landau damping, ion-acoustic wave, two-stream instability. The Fourier–Hermite spectral method can achieve solutions that are several orders of magnitude more accurate at a fraction of the cost with respect to PIC.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The Vlasov equation represents the cornerstone for the kinetic modeling of collisionless plasmas. It describes the time evolution of the distribution function of a population of charged particles that respond self-consistently to the effect of self- and externally induced electromagnetic fields. The numerical solution of the Vlasov equation for collisionless plasmas represents a very active area of research. The main challenge resides in the fact that the distribution function lives in a six dimensional phase space. Unarguably, the most widely used technique to solve the Vlasov equation is the Particle-In-Cell (PIC) method [1,2]. The main idea, originally developed in the 1950s, is to sample the distribution function in velocity space by means of a discrete number of super-particles. The electromagnetic field is represented on a grid in the computational domain, and the super-particles move through the grid according to the Lorentz force that is calculated by interpolation from the grid to the particles position. The particles interactions are therefore mediated by the grid, and in this way the number of operations per time step is reduced from $\sim N_p^2$ (if the total force on a particle is calculated by summing the

pair interaction with every other particle) to $\sim N_p$ (with N_p the number of super-particles). A comprehensive review of the PIC methodology can be found in [3].

The primary shortcoming of PIC simulations is related to the numerical noise: even starting from an equilibrium configuration, the discrete nature of the particles generates instantaneous (i.e. within the first time step) unphysical perturbations that produce a ‘noise ground’ level in the fields, below which any physical signal is lost. The most obvious way to reduce the noise in PIC is by increasing the number of super-particles, i.e. refining the discretization in velocity space. The main problem is that while the simulation time scales roughly linearly with N_p , the noise ground level decreases only as $N_p^{-1/2}$, as typical of Monte Carlo methods, implying that problems that require a high signal-to-noise ratio require significant computational resources. Some examples that illustrate the impact of the particle noise on the efficiency of PIC codes are the recent studies in space plasmas on the acceleration of particles in the solar wind and the coupling between large scale turbulence and small scale (i.e. kinetic) dissipative effects [4–6]. PIC codes have demonstrated very poor performances for these problems. In two-dimensional (2D) simulations, [7] have shown that at least $\sim 10,000$ particles per cell (equivalent to $\sim 10^6$ in 3D) are needed to achieve a sufficiently large signal-to-noise ratio. For comparison, one of the biggest and most advanced PIC simulations in the world (of a magnetic reconnection problem) was done

* Corresponding author.

E-mail address: e.camporeale@cwi.nl (E. Camporeale).

with 250 particles per cell and required the use of the highest performance, petascale computer available at that time [8,9]. We note that methods based on δf or re-mapping of the distribution function have been developed in the PIC community to reduce particle noise. This has led to some very interesting work (see for instance [10–12]), but has not yet resulted in a commonly accepted denoising technique adopted by the PIC community.

The non-optimal scaling of the noise level with the number of particles in the PIC method has long been recognized in the computational plasma physics community [13–16], and it has perhaps been the main motivation for developing alternative ways of solving the Vlasov equation numerically. Nevertheless, despite its intrinsic noise, the PIC method still represents the preferred approach in the community, probably due to its simplicity and robustness, the relative ease to parallelize it, and the recent impressive advances in available computing power. However, one should always keep in mind the resource-intensive nature of PIC and the related cost. As a figure of merit on the cost of large scale PIC simulations, a single run using the global gyrokinetic PIC code developed within the SciDAC Gyrokinetic Particle Simulation Center (GPSC) [17] for investigating the long-time evolution of turbulent transport in nuclear fusion devices was estimated, in 2008, to take approximately 25 million CPU hours, on 100, 000+ cores [18]. The monetary cost of such a simulation can be estimated at about \$1,000,000 [19].

An alternative to the PIC method is represented by solving the Vlasov equation directly in phase-space (namely with a sixth dimensional computational grid in space and velocity), by means of a so-called Eulerian Vlasov code. This has been done, usually for problems with reduced dimensionality, with a variety of techniques such as high-order finite differences [20,21], finite elements [22,23], or finite volumes [24,25]. The reader is referred to [26,27] for a recent review of different methods. Notably, successful algorithms include the semi-Lagrangian schemes [28–30, 22,31–38] and positive flux conservative methods [39].

Yet another class of techniques is represented by spectral methods where the velocity space is projected onto a complete orthogonal basis [40–45].

Clearly, all these methods have advantages and shortcomings. Historically, the PIC method has been favored for large-scale, multidimensional problems which are still not doable with Vlasov solvers (mainly due to memory limitations) [46]. On the other hand, Vlasov codes allow the study of fine scale details of the distribution functions that are typically inaccessible in PIC codes, due to the above mentioned noise problem. Vlasov codes, however, can suffer from serious numerical problems that can lead to a lack of discrete conservation properties, the occurrence of unphysical oscillations, and the generation of non-positive values in the distribution functions [24].

In this paper, we focus on a spectral method where the distribution function is projected onto an Hermite basis in velocity space. This gives rise to a set of nonlinear, time- and space-dependent PDEs for the coefficients of the expansion. The use of Hermite functions to discretize the velocity space in the Vlasov equation dates back to the works of [47,40,48], and [41]. More recently this approach has been investigated by [49,50,45] and, in the context of linearized equations, by [51] and [52].

The expansion of the distribution function using an Hermite basis in velocity space can be appealing for several reasons. First, the Hermite functions are a complete orthogonal basis with respect to a Gaussian weight function. As such, they are the optimal basis to represent Maxwellian-like distributions. In fact, an exact Maxwellian in velocity can be represented by only one term of the Hermite basis. Second, as already noted by [47], the low order expansion coefficients are directly related to the low order moments of the distribution functions (i.e. density, mean velocity, energy,

heat flux, etc.) that describe the plasma macroscopically and are usually the physical quantities of interest. As a consequence, the use of the Hermite basis allows a smooth transition from a fluid to a kinetic description by simply increasing the number of coefficients retained [53]. This is an important and crucial feature of this method in approaching multi-scale problems and in assessing the importance of kinetic effects, which is not available in PIC or Vlasov methods that are forced to treat the full distribution function.

This paper builds largely upon the work of [50]. In Ref. [50] two different Hermite bases (so called ‘asymmetrically’ and ‘symmetrically’ weighted) and their properties were discussed, and a qualitative comparison between the Fourier–Hermite (FH) method and the PIC method was presented for simulations of Landau damping and bump-on-tail instability. Here we expand that work presenting an implicit time discretization, and quantitatively comparing the new scheme with an implicit PIC. Furthermore, we will show that with this discretization and periodic boundary conditions it is possible to conserve the total charge, momentum, and energy exactly in a finite time step (in contrast, as we discuss below, only charge and energy is conserved in the implicit PIC of [54]).

Explicit PIC has often been criticized due to its stringent requirements on the choice of the time step and grid size, for numerical stability reasons. In fact, an explicit PIC code requires the resolution of the smallest time scale and the shortest spatial scale, even when the physics of interest only involves larger time/spatial scales. Of course, this generally translates into the requirement for large computational resources. Historically, the search for more efficient PIC schemes based on implicit time discretization dates back to the 1980s, when the implicit moment [55–57] and the direct implicit [58,59] methods were introduced. Both methods rely on a linearization of the equations for the electromagnetic fields, and, thus, they should be more properly regarded as *semi-implicit* methods. Using these techniques, the numerical stability is greatly improved (with respect to explicit PIC), but energy is still not conserved exactly and, at each time step, there is a small inconsistency between the charge and current densities calculated from the particles and the one that is used for advancing the fields. Some authors have suggested that such limitations are responsible for the accumulation of numerical errors that preclude semi-implicit PIC simulations to run for long time intervals [54,60]. Recently, however, [61] and [54] have formulated and successfully implemented a fully-implicit, one-dimensional PIC code (see also [62] for an electromagnetic extension). The main feature that characterizes fully-implicit PIC is that particles and fields are advanced simultaneously through a Jacobian-Free Newton–Krylov (JFNK) solver, and converged nonlinearly within a certain tolerance. Moreover, by using the so-called particle enslavement, the nonlinear solver converges on a residual that does not contain particle positions and velocities. In the fully-implicit PIC method, energy is conserved within the level imposed by the nonlinear tolerance (i.e. almost exactly). [54] have shown that by implementing a sub-stepping procedure that makes particles stop every time they cross a cell boundary, the charge is also conserved exactly. On the other hand, momentum is not conserved and must be monitored throughout the simulation (in contrast to semi-implicit PIC that exactly conserves the momentum, but not the energy).

When comparing different schemes, the key information is represented by the CPU time needed to obtain a solution with a certain accuracy (this metric was not considered in [50]), in order to be able to assess which method must be preferred (and for which conditions). Hence, our comparisons between FH and PIC are presented in terms of computational efficiency and efficacy. The former is essentially represented by the CPU time required to achieve a certain accuracy in the solution, while the latter is a measure of the cost-effectiveness of the algorithm, i.e. how much

an increased resolution and consequently a longer CPU time is paid off in terms of better accuracy [63].

For the examples presented in this paper involving mostly near-equilibrium Maxwellian plasmas in one dimension, the FH method is several orders of magnitude more accurate than PIC for equal CPU time or, conversely, results with the same level of accuracy can be obtained in a fraction of CPU time.

The paper is organized as follows. In Section 2, we introduce the Fourier–Hermite expansion of the Vlasov–Poisson system in one dimension (1D). We use a fully-implicit discretization of the equation based on the Crank–Nicolson scheme, which leads to a non-linear system of equations that is solved numerically by means of a JFNK solver. We also show that the fully-implicit discretization ensures exact charge, momentum and energy conservation. Section 3 presents the comparison between FH and PIC for four standard cases: Langmuir wave, linear Landau damping, two-stream instability, and ion acoustic wave. We present conclusions in Section 4.

2. Numerical method: Fourier–Hermite basis

We study the Vlasov–Poisson system in the 2-dimensional phase space (x, v) , where x denotes position and v velocity. The phase space is assumed to be periodic in x . In order to describe the method we specialize to the case of a plasma consisting of an electron and a singly charged ion species. The quantities are normalized as follows. Time is normalized on the electron plasma frequency ω_{pe} , velocities on the electron thermal velocity $v_{te} = \sqrt{kT_e/m_e}$, lengths on the electron Debye length λ_D , the electric field on $\frac{m_e}{e} v_{te} \omega_{pe}$ (k is the Boltzmann's constant, T_e is the electron temperature, m_e is the electron mass and e is the elementary charge), and densities on a reference density n_0 . The Vlasov equation for the species s reads:

$$\frac{\partial f_s}{\partial t} + v \frac{\partial f_s}{\partial x} + \frac{q_s m_e}{e m_s} E \frac{\partial f_s}{\partial v} = 0. \quad (1)$$

Here f_s is the particle distribution function, q_s and m_s are the charge and mass of the particles of species s ($s = e, i$ for electrons and ions, respectively) and E is the electric field. The electric field is self-consistently calculated through the Poisson equation:

$$\frac{\partial E}{\partial x} = \int_{-\infty}^{\infty} f_i dv - \int_{-\infty}^{\infty} f_e dv. \quad (2)$$

The discretization in velocity employs the asymmetrically-weighted Hermite basis:

$$\Psi_n(\xi) = (\pi 2^n n!)^{-1/2} H_n(\xi) e^{-\xi^2} \quad (3)$$

$$\Psi^n(\xi) = (2^n n!)^{-1/2} H_n(\xi) \quad (4)$$

where H_n is the n th Hermite polynomial, defined as

$$H_n(\xi) = (-1)^n \exp(\xi^2) \frac{d^n}{d\xi^n} \exp(-\xi^2). \quad (5)$$

The distribution function $f(x, v, t)$ is defined as:

$$f_s(x, v, t) = \sum_{n=0}^{N_H-1} C_n^s(x, t) \Psi_n(\xi_s), \quad (6)$$

with $\xi_s = (v - u_s)/\alpha_s$, where u_s and α_s are two constant parameters for each species, and N_H is the number of Hermite modes (equal for both species). The following closure is used for both species: $C_n^s = 0$ for $n \geq N_H$. The spatial dependence of the expansion coefficients C_n^s will be treated later by means of a Fourier decomposition. We note that the completeness of the Hermite basis does not depend on the choice of the free parameters u_s and α_s . It is

well known, however, that a proper choice of the rescaling coefficient α_s can substantially accelerate the convergence of the series [50,64,51]. In this work, we allow even more flexibility by introducing the free parameter u_s , which is a shift in the Hermite function argument. The usefulness of such a parameter will be discussed in Section 4.4. Formulas for calculating the optimal values for α_s and u_s were presented in [51].

The Hermite basis has the following properties ($\delta_{n,m}$ is the Kronecker delta):

$$\int_{-\infty}^{\infty} \Psi_n(\xi) \Psi^m(\xi) d\xi = \delta_{n,m}, \quad (7)$$

$$v \Psi_n(\xi) = \alpha \sqrt{\frac{n+1}{2}} \Psi_{n+1}(\xi) + \alpha \sqrt{\frac{n}{2}} \Psi_{n-1}(\xi) + u \Psi_n(\xi), \quad (8)$$

$$\frac{d\Psi_n(\xi)}{d\xi} = -\frac{1}{\alpha} \sqrt{2(n+1)} \Psi_{n+1}(\xi). \quad (9)$$

One can multiply Eq. (1) by $\Psi^n(\xi)$, integrate in $d\xi$ and, by using such properties, obtain:

$$\begin{aligned} \frac{\partial C_n^s}{\partial t} + \alpha_s \left(\sqrt{\frac{n+1}{2}} \frac{\partial C_{n+1}^s}{\partial x} + \sqrt{\frac{n}{2}} \frac{\partial C_{n-1}^s}{\partial x} + \frac{u_s}{\alpha_s} \frac{\partial C_n^s}{\partial x} \right) \\ - \frac{q_s m_e}{e m_s} \frac{\sqrt{2n}}{\alpha_s} E C_{n-1}^s = 0. \end{aligned} \quad (10)$$

Similarly, the Poisson Eq. (2) becomes:

$$\frac{\partial E(x)}{\partial x} = \alpha_i C_0^i(x) - \alpha_e C_0^e(x). \quad (11)$$

One can notice the well-known property that the electric field is related uniquely to the 0th order expansion coefficients C_0^s , i.e., the density. By now treating $E(x)$ and $C_n(x, t)$ by means of a standard discrete Fourier decomposition (with L the domain length, and $2N+1$ modes):

$$C_n^s(x) = \sum_{k=-N}^N C_{n,k}^s e^{\frac{2\pi i k x}{L}} \quad (12)$$

$$E(x) = \sum_{k=-N}^N E_k e^{\frac{2\pi i k x}{L}}, \quad (13)$$

and using the orthogonality of the Fourier basis, one can finally derive an infinite set of ordinary differential equations for the coefficients $C_{n,k}^s(t)$:

$$\begin{aligned} \frac{dC_{n,k}^s}{dt} + \alpha_s \frac{2\pi i k}{L} \left(\sqrt{\frac{n+1}{2}} C_{n+1,k}^s + \sqrt{\frac{n}{2}} C_{n-1,k}^s + \frac{u_s}{\alpha_s} C_{n,k}^s \right) \\ - \frac{q_s m_e}{e m_s} \frac{\sqrt{2n}}{\alpha_s} \sum_{m=-N}^N E_{k-m} C_{n-1,m}^s = 0, \end{aligned} \quad (14)$$

where $n(k)$ is the index associated with the Hermite (Fourier) basis. The expression for the electric field reads:

$$E_k = \frac{iL}{2\pi k} (\alpha_e C_{0,k}^e - \alpha_i C_{0,k}^i) \quad \text{for } k \neq 0, \quad (15)$$

$$E_0 = 0. \quad (16)$$

We note that the Fourier representation of Poisson Eq. (11) leaves the constant E_0 undefined, while imposing the constraint $\alpha_i C_{0,0}^i = \alpha_e C_{0,0}^e$ (which physically means that the plasma is neutral). However, the absence of an externally imposed electric field and the periodicity of the domain dictates that $E_0 = 0$.

The solution of Eq. (14) is the main objective of this paper. An important point to realize is that the only non-linearity of this

equation is in the last term, which couples the electric field with the distribution function via a convolution. Also, in the Hermite (velocity) space, the n th coefficient C_n is coupled only to C_{n-1} , C_{n+1} , to itself (if $u_s \neq 0$), and to C_0 . Eq. (14) is discretized in time with a second order accurate fully-implicit Crank–Nicolson scheme [65], which, for each species, reads (omitting subscript s):

$$\begin{aligned} & \frac{C_{n,k}^{t+1} - C_{n,k}^t}{\Delta t} + \alpha \frac{\pi i k}{L} \left[\sqrt{\frac{n+1}{2}} (C_{n+1,k}^{t+1} + C_{n+1,k}^t) \right. \\ & \left. + \sqrt{\frac{n}{2}} (C_{n-1,k}^{t+1} + C_{n-1,k}^t) + \frac{u_s}{\alpha} (C_{n,k}^{t+1} + C_{n,k}^t) \right] \\ & - \frac{q m_e}{e m} \frac{\sqrt{2n}}{4\alpha} \sum_{m=-N}^N (E_{k-m}^{t+1} + E_{k-m}^t) (C_{n-1,m}^{t+1} + C_{n-1,m}^t) = 0, \quad (17) \end{aligned}$$

where subscript t indicates the time step.

In summary, the problem is reduced to the numerical solution of the set of nonlinear equation (17). We have employed a Jacobian-Free Newton–Krylov solver, which algorithmically consists in defining Eq. (17) as a residual that is iteratively minimized at each time step [66].

2.1. Conservation properties

As noted in [50], the decomposition in the asymmetric Fourier–Hermite basis allows conservation of particle number and momentum exactly, irrespective of the chosen time discretization scheme. On the other hand, exact energy conservation was not possible with the operator splitting scheme in Ref. [50]. It is however possible to recover exact energy conservation by employing the fully-implicit time discretization proposed here. The simultaneous conservation of charge, momentum and energy is an important property of this technique, especially when compared with PIC codes. Indeed, it is important to emphasize that no existing PIC code can simultaneously conserve energy and momentum. In practice, this means that the non-conserved quantity must be monitored throughout the simulation in order to ensure that its error is somehow bounded.

Following [50], the total number of particles for each species is defined as

$$M_s = \int_0^L \int_{-\infty}^{\infty} f_s(x, v, t) dv dx = L \alpha_s C_{0,0}^s. \quad (18)$$

By inspection of Eq. (14) one can see that $\frac{dC_{0,0}^s}{dt} = 0$, from which the conservation of mass follows.

The single species linear momentum is defined as:

$$\begin{aligned} P_s &= \frac{m_s}{m_e} \int_0^L \int_{-\infty}^{\infty} v f_s(x, v, t) dv dx \\ &= \frac{m_s}{m_e} \left(\frac{L \alpha_s^2}{\sqrt{2}} C_{1,0}^s + u_s \alpha_s L C_{0,0}^s \right). \quad (19) \end{aligned}$$

In order to prove the conservation of momentum, one can calculate the time derivative of the coefficients $C_{1,0}^i$ and $C_{1,0}^e$:

$$\frac{dP_i}{dt} = \frac{m_i}{m_e} \frac{L \alpha_i^2}{\sqrt{2}} \frac{dC_{1,0}^i}{dt} = L \alpha_i \sum_{m=-N}^N E_{-m} C_{0,m}^i \quad (20)$$

$$\frac{dP_e}{dt} = \frac{L \alpha_e^2}{\sqrt{2}} \frac{dC_{1,0}^e}{dt} = -L \alpha_e \sum_{m=-N}^N E_{-m} C_{0,m}^e. \quad (21)$$

One can use Eq. (15) to substitute $C_{0,m}^e$ in Eq. (21) and obtain:

$$\begin{aligned} \frac{dP_e}{dt} &= -L \alpha_e \sum_{m=-N}^N E_{-m} \left(\frac{\alpha_i}{\alpha_e} C_{0,m}^i - \frac{2\pi m i}{L \alpha_e} E_m \right) \\ &= -L \sum_{m=-N}^N \left(\alpha_i E_{-m} C_{0,m}^i - \frac{2\pi m i}{L} |E_m|^2 \right). \quad (22) \end{aligned}$$

By taking into account that E_{-m} is the complex conjugate of E_m (since the electric field in physical space is a real quantity), it follows that $|E_{-m}|^2 = |E_m|^2$ and therefore the last term in parenthesis in Eq. (22) is zero. Hence $\frac{dP_i}{dt} + \frac{dP_e}{dt} = 0$.

The following proof of the conservation of energy for the FH scheme is, for simplicity, carried out for a single species. More general cases are straightforward, but with more cumbersome algebra. The kinetic and potential energy W_K and W_E are defined as:

$$\begin{aligned} W_K &= \frac{1}{2} \int_0^L \int_{-\infty}^{\infty} v^2 f(x, v, t) dv dx \\ &= \frac{\alpha^3 L}{4} (\sqrt{2} C_{2,0} + C_{0,0}) + \frac{u \alpha}{2} (\alpha \sqrt{2} C_{1,0} + u C_{0,0}) \quad (23) \end{aligned}$$

$$W_E = \frac{L}{2} \sum_k |E_k|^2. \quad (24)$$

The change of potential energy between time $t + 1$ and time t is

$$\begin{aligned} \Delta W_E &= \frac{L}{2} \sum_k |E_k^{t+1}|^2 - |E_k^t|^2 = \frac{L^3 \alpha^2}{8\pi^2} \sum_{k \neq 0} \frac{1}{k^2} (|C_{0,k}^{t+1}|^2 - |C_{0,k}^t|^2) \\ &= \frac{L^3 \alpha^2}{8\pi^2} \sum_{k \neq 0} \frac{1}{k^2} \left(|C_{0,k}^t - \frac{\Delta t \alpha \pi i k}{\sqrt{2} L} (C_{1,k}^{t+1} + C_{1,k}^t)|^2 - |C_{0,k}^t|^2 \right) \\ &= \sum_{k \neq 0} \frac{\alpha^4 L \Delta t^2}{16} |C_{1,k}^{t+1} + C_{1,k}^t|^2 \\ &\quad + \frac{\sqrt{2} L^2 \alpha^3 \Delta t}{8\pi k} (\text{Re}(C_{0,k}^t) \text{Im}(C_{1,k}^{t+1} + C_{1,k}^t) \\ &\quad - \text{Im}(C_{0,k}^t) \text{Re}(C_{1,k}^{t+1} + C_{1,k}^t)), \end{aligned}$$

where Eq. (17) has been used. Re and Im indicate the real and imaginary parts of a complex quantity. Similarly, one can calculate the change of kinetic energy between times $t + 1$ and time t , considering that $C_{0,0}^{t+1} = C_{0,0}^t$ and $C_{1,0}^{t+1} = C_{1,0}^t$.

$$\begin{aligned} \Delta W_K &= \frac{\alpha^3 L \sqrt{2}}{4} (C_{2,0}^{t+1} - C_{2,0}^t) \\ &= \frac{\alpha^3 L \sqrt{2}}{4} \left(-\frac{\Delta t}{2\alpha} \sum_{k \neq 0} (E_{-k}^{t+1} + E_{-k}^t) (C_{1,k}^{t+1} + C_{1,k}^t) \right) \\ &= \frac{\Delta t \alpha^3 L^2 i \sqrt{2}}{16\pi} \sum_{k \neq 0} \frac{1}{k} (C_{0,-k}^{t+1} + C_{0,-k}^t) (C_{1,k}^{t+1} + C_{1,k}^t) \\ &= \frac{\Delta t \alpha^3 L^2 i \sqrt{2}}{16\pi} \sum_{k \neq 0} \frac{1}{k} \left(2C_{0,-k}^t + \frac{\Delta t \alpha \pi i k}{\sqrt{2} L} (C_{1,-k}^{t+1} + C_{1,-k}^t) \right) \\ &\quad \times (C_{1,k}^{t+1} + C_{1,k}^t) \\ &= -\sum_{k \neq 0} \frac{\sqrt{2} L^2 \alpha^3 \Delta t}{8\pi k} (\text{Re}(C_{0,k}^t) \text{Im}(C_{1,k}^{t+1} + C_{1,k}^t) - \text{Im}(C_{0,k}^t) \\ &\quad \text{Re}(C_{1,k}^{t+1} + C_{1,k}^t)) + \frac{\alpha^4 L \Delta t^2}{16} |C_{1,k}^{t+1} + C_{1,k}^t|^2. \end{aligned}$$

Therefore, $\Delta W_E + \Delta W_K = 0$.

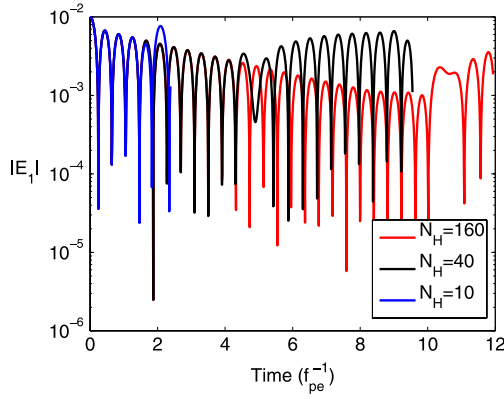


Fig. 1. Landau damping simulation with parameters: $\alpha_e = 1$, $\varepsilon = 0.01$. Recurrence phenomenon for the FH method due to filamentation. The plot shows the amplitude of E_1 (the mode in the electric field perturbed at $t = 0$). N_H is the number of Hermite modes employed: $N_H = 10$ (blue line); $N_H = 40$ (black line); $N_H = 160$ (Red line). The recurrence time is increased by increasing N_H , scaling approximately as $N_H^{1/2}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Filamentation and artificial collisionality

The development of increasingly smaller phase space structures in a collisionless plasma is very well known in plasma physics and typically referred to as the filamentation process. A classical and well-studied example where filamentation occurs is linear Landau damping, i.e., the damping in time of an initial electric field perturbation due to wave-particle resonances [67]. In this case filamentation is due to the presence of the factor $\exp[-ikvt]$ (k is the wavevector of the perturbation) in the perturbed distribution function that creates oscillations with smaller and smaller wavelengths in velocity space as time evolves. Clearly, any discretization of the velocity space is associated with a minimum wavelength that can be resolved, and, therefore, any numerical simulation of the filamentation process with fixed resolution is bound to fail after a certain time.

In the case of the FH expansion method, it is known that the time at which the simulation is unable to capture filamentation due to lack of resolution in velocity space scales approximately as the square root of the number of Hermite modes, N_H [48,68,50]. This time is known as the recurrence time: the effect of the lack of resolution is to reconstruct a distribution function similar to the initial one, from which the Landau damping starts again in a recurrent way. Fig. 1 shows the recurrence effect for the linear Landau damping studied with the FH code (this case will be studied in more detail in Section 4.2). Blue, black, and red lines show the amplitude of E_1 (the Fourier component of the electric field perturbed at time $T = 0$) for $N_H = 10, 40, 160$, respectively. One can appreciate that the time at which the simulation manifests the recurrence phenomenon approximately doubles when N_H is multiplied by a factor of 4, as obtained in [50].

Several fixes have been suggested in the literature in order to overcome the filamentation process in Vlasov codes. They involve some form of filtering or smoothing of the high order moments of the distribution function (see, e.g. [69,70]) or, equivalently, the introduction of a weakly-collisional operator [48].

In this work, we have tested the effect of a collisional operator. This is a purely numerical artifact, that does not represent physical collisions, and must be used by tuning it such that the physical results of interest are unaffected, while the small filamentation structures are damped. We have opted for a collision operator \mathcal{C} that acts on the coefficient $C_{n,k}$ as:

$$\mathcal{C}[C_{n,k}] = -\nu \frac{n(n-1)(n-2)}{(N_H-1)(N_H-2)(N_H-3)} C_{n,k}, \quad (25)$$

where ν is the collisional rate applied to the last Hermite coefficients $C_{N_H-1,k}$. The collision operator of Eq. (25) can be constructed as a nonlinear combination of terms involving the Lenard-Bernstein collision operator \mathcal{C}_{LB} [71]:

$$\mathcal{C}_{LB}[f] = -\nu \frac{\partial}{\partial v} \left(vf + \frac{1}{2} \frac{\partial f}{\partial v} \right), \quad (26)$$

which, in the Fourier-Hermite space, reads:

$$\mathcal{C}_{LB}[f] = -\nu n C_{n,k}. \quad (27)$$

An important point is that the Lenard-Bernstein operator transformed in the Fourier-Hermite space acts on all the coefficients $C_{n,k}$, including $n = 0, 1, 2$, while our operator is defined in such a way that it does not change the first three Hermite modes. The reason for choosing the form in Eq. (25) is that it conserves charge, momentum, and energy, by leaving $C_{0,k}$, $C_{1,k}$, $C_{2,k}$ unchanged. This result holds independently of the value of ν . The fact that this operator does not have a physical interpretation is not important, since our goal is to study collisionless plasmas. Also, the regime of validity of the simulation will be reduced to times for which the collisional rate is not dominant. Of course, other forms of the collision operator might be employed: for instance [72] proposes an iterative version of Eq. (26). The crucial feature, however, is that the damping rate applied to the coefficient $C_{n,k}$ must increase (in absolute value) with increasing Hermite index n .

The effect of the artificial collision operator is to damp the highest modes of the Hermite expansion. The convergence of the series implies that $|C_{n+1}|/|C_n| \rightarrow 0$ for large enough n . However, high n modes can grow due to the filamentation process or even just due to numerical errors. As we will show in Section 4.4, this can lead to numerical instabilities if the growth of the large n modes is not suppressed artificially.

The effect of collisionality on the Landau damping study is presented in Fig. 2. Here $N_H = 40$ and three values of ν have been used: $\nu = 0$, $\nu = 0.1$, $\nu = 1$. One can notice that the correct value of Landau damping (i.e. the one obtained before recurrence when $\nu = 0$) is recovered for a long time, when $\nu = 1$ and, therefore, in this case the use of collisionality is crucial to overcome the recurrence due to filamentation. The small box shows a zoom-in for time $T < 12$. The plot in Fig. 2 must be interpreted in light of the important result presented in [73], and rediscussed in [74]. It is well known that Landau damping in a collisionless plasma is due to the effect of the destructive interference of a continuous spectrum of singular eigenmodes (the Case-Van Kampen modes). [73] have shown that a Lenard-Bernstein collisional operator changes the spectrum of the linear Vlasov problem by replacing the singular continuous spectrum with a set of proper discrete eigenmodes, and that the Landau damping is recovered as a discrete mode (along with other modes). In this context, Fig. 2 clearly shows that the right damping rate (consistent with Landau damping) can be recovered. One has to keep in mind, however, that although the macroscopic nature of the plasma has been preserved, the microscopic information associated with the high order moments of the distribution function is irreversibly modified by applying the collisional operator. On the other hand, in most simulations of a kinetic collisionless plasma, the use of an artificial collisional operator is necessary because filamentation is an intrinsic feature. Once again, the underlying assumption (to be verified through a convergence study) is that the use of artificial collisions will not affect the macroscopic evolution of the system.

Obviously, a PIC code is not immune to filamentation problems: the use of a discrete number of super-particles implies a finite resolution in velocity space. However, the fact that any velocity value can be assigned to a single particle and hence that the discretized phase-space is not gridded in any standard way makes it difficult to quantify the relationship between the number of

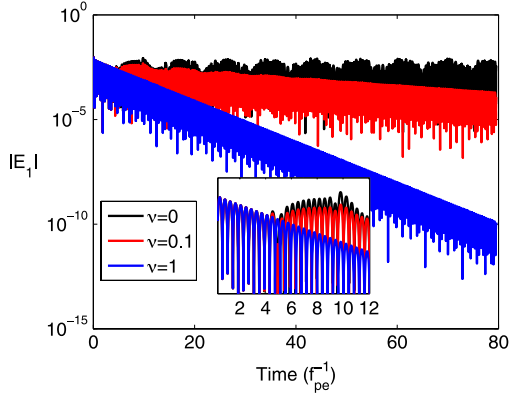


Fig. 2. Effect of the numerical collision term in the Landau damping simulation for the FH method, with parameters: $\alpha_e = 1$, $\varepsilon = 0.01$, $N_H = 40$. When the simulation is completely collisionless ($\nu = 0$, black line) the recurrence effect due to filamentation does not allow long-time simulations. The inset box shows a zoom-in at initial times. The correct damping is recovered for $\nu = 1$.

particles and the actual resolution in velocity space. On the other hand, it is very well-known that PIC simulations have difficulties in reproducing the fine details of a distribution function, such as high tails (contrary to Eulerian Vlasov codes, see, e.g., [4,75,76]). In Fig. 3, we show that although PIC simulations do not present recurrence similarly to Vlasov codes, the lack of resolution in velocity space still manifests itself and produces inaccurate results. Top, middle and bottom panels show PIC results (black line) for number of particles per cell $N_{pcel} = 1600$, 6400, and 25,600, respectively. The red line is a reference solution calculated with the FH code (with $N_H = 100$). For all three of these cases, the correct result is lost at some point, and the solution becomes essentially noise. A useful way of understanding this deviation from the correct solution is to look at the signal-to-noise ratio. In Fig. 4, we show the time evolution of the Landau damping test for different values of the initial perturbation ε (see Section 4 for the discussion of the initialization) from PIC simulations with 2,000,000 particles per cell. The black line indicates the noise level, which has been calculated as the maximum value of $|E_1|$ in a case without initial perturbation ($\varepsilon = 0$). Blue and red lines denote simulations with $\varepsilon = 0.001$ and $\varepsilon = 0.01$, respectively. The black dashed line shows the theoretical Landau damping rate. Although the two simulations have the same number of particles per cell, starting with a lower initial amplitude (blue line) (i.e. at a lower signal-to-noise level), clearly impacts the result: the Landau damping is almost immediately lost for $\varepsilon = 0.001$. In other words, the noise ground level sets the amplitude of a signal at which the simulation loses any physical interpretation. Of course, in the Hermite method, there is no noise ground level and the equivalent simulation (shown in Fig. 1) is independent of the value of the initial perturbation (given that the value is small enough to be in the linear regime).

4. Results

In this section we compare the performance of the Fourier-Hermite (FH) method with the PIC method, both implemented with fully-implicit time discretization. For the fully-implicit PIC method, we follow the approach of [54], that nonlinearly solves the Ampere equation discretized in time with a Crank-Nicolson scheme. The current density is self-consistently calculated from the particles. The only difference between the fully-implicit PIC used in this paper and the one employed in [54] is that we do not use a space filter (smoothing), for the following reason. We have calculated the numerical plasma frequency from a PIC simulation of Langmuir wave in a cold plasma, and we have verified that

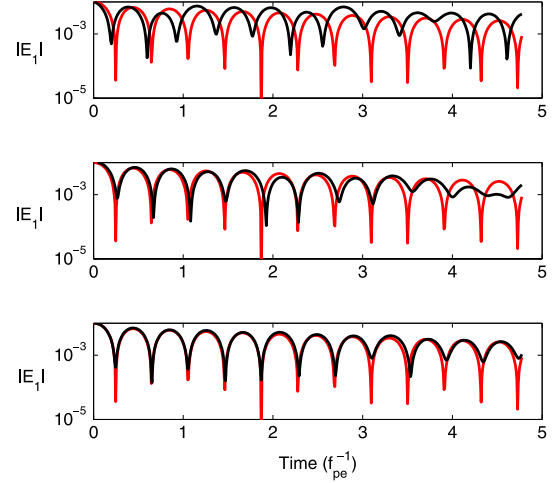


Fig. 3. PIC simulation of Landau damping with parameters $\alpha_e = 1$, $\varepsilon = 0.01$. The plot shows the amplitude of E_1 (the mode in the electric field perturbed at $t = 0$) for different number of particles per cell: $N_{pcel} = 1600$ (top panel), $N_{pcel} = 6400$ (middle panel), $N_{pcel} = 25,600$ (bottom panel). The black line is the PIC result and the red line is the reference solution (obtained with the Hermite code with $N_H = 100$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

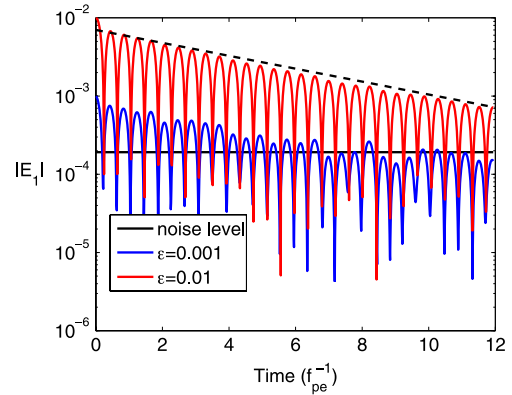


Fig. 4. PIC simulation of Landau damping. The number of particle per cell is $N_{pcel} = 2,000,000$. Different colors denote different initial amplitudes: $\varepsilon = 0.001$ (blue); $\varepsilon = 0.01$ (red). The black solid line indicates the average noise level. The dashed line indicates the theoretical Landau damping rate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

its deviation from the theoretical plasma frequency (equal to 1, in normalized units) is 4 times larger when using the binomial filter adopted in [54]. The PIC code employs linear interpolation, usually referred to as ‘Cloud-in-Cell’ (CIC) [1]. Note that higher order interpolation schemes have been proposed, for instance, in [77,78]. In order to make the comparison as fair as possible, we keep the same number of grid points for FH and PIC. Note that, by solving Ampere’s instead of Poisson’s equation, the fully-implicit PIC method does not require a space derivative operator. Hence, there is no equivalent of the Fourier discretization in the PIC. Of course, the grid spacing introduces a numerical error in the current accumulation and interpolation routines. Finally, although a preconditioned version of the fully-implicit PIC has been presented in [60], we use the unpreconditioned version here, so that both methods are unpreconditioned. Since the focus of this work is on the discretization in velocity space, i.e. the comparison between the spectral Hermite method and the use of super-particles, all simulations are performed for the same choice of timestep and grid size. The comparison is characterized by the following three metrics:

1. the error with respect to a ‘reference’ highly accurate solution as a function of CPU time and velocity discretization (number of Hermite modes N_H for FH and number of particles per cell N_{pcel} for PIC);
2. the error with respect to the ‘previous’ solution (i.e. less refined in velocity space);
3. the efficacy defined as the inverse of the product of CPU time and error.

The error used for all runs is calculated as the L_1 norm of the difference between two electric field solutions, averaged in time. The error in (2) is what is actually used by a user who is performing a convergence study to decide when the solution is accurate enough. The efficacy is a useful indicator of the cost-effectiveness of an algorithm. It measures whether an additional cost in terms of CPU time is compensated by a gain in terms of accuracy. Clearly, an algorithm performs well if the efficacy increases notably with increasing CPU time. In this regard, we note that the PIC algorithm, by construction, performs badly in terms of efficacy since the error scales as $N_{pcel}^{-1/2}$, while the computing time scales roughly linearly with N_{pcel} . Therefore, the efficacy scales as the inverse of the square root of the CPU time, i.e., it actually decreases with increasing CPU time. Hence, from a pure cost-effectiveness point of view, it is *never* advantageous to increase the number of particles in a PIC code to reduce the error. On the other hand, one is often forced to have a large number of particles such that the physical signal is above the noise level (see e.g. [7]). On the other hand the FH efficacy grows exponentially, as typical of spectral methods.

For all the cases discussed below we initialize the electrons (ions) with a Maxwellian distribution function with thermal velocity α_e (α_i). For the Langmuir wave, the Landau damping and the two-stream instability tests the ions constitute a fixed background, while for the study of the ion-acoustic wave they evolve. The initial electric field is initialized as:

$$E(x, t = 0) = \frac{L}{2\pi} \varepsilon \sin(2\pi x/L), \quad (28)$$

where ε is the amplitude of the initial perturbation. In Fourier space, such an initialization corresponds to:

$$E_{-1} = E_1 = -\frac{\varepsilon L}{4\pi} \quad (29)$$

and the density is initialized consistently. For all runs, the number of grid points (Fourier modes for FH) is equal to $2N + 1 = 33$ and the timestep is $\Delta t = 0.05$. We use the `nsoli` routine described in [66] for the JFNK solver. The Krylov solver is a non-preconditioned restarted GMRES. For all simulations reported in this paper, the Newton–Krylov absolute and relative tolerances are set to 10^{-8} . All the codes are written in MATLAB and run on an Intel Xeon 3.40 GHz Linux box.

4.1. Langmuir wave

The parameters are chosen as follows: $L = 2\pi$, $\alpha_e = 0.1\sqrt{2}$, $\varepsilon = 0.01$, $\nu = 0$. Fig. 5 shows the comparison between FH and PIC. The top panels show the error with respect to the ‘reference’ solution (red circles) and with respect to the ‘previous’ less refined solution (black circles). The results for PIC are on the left, as a function of number of particles N_{pcel} , and the results for FH are on the right as a function of number of Hermite polynomials N_H . The reference solutions are calculated with $N_{pcel} = 102,400$ and $N_H = 100$, respectively for PIC and FH. The PIC result recovers the theoretical scaling with $N_{pcel}^{-1/2}$ (black line). For this case both PIC and FH reach a low error with a relatively low number of particles per cell and Hermite coefficients, respectively. This would not be the case with an explicit PIC (not shown), and thus one can infer

that the conservation of energy is beneficial, in this case, to assure a faster convergence of the solution. The two bottom panels of Fig. 5 show the error (left) and the efficacy (right) as a function of the CPU time. Black circles are for PIC and red circles are for FH. Here the error is with respect to the reference solution. Although both methods are extremely accurate, as a figure of merit, in order to reach an error of the order of 10^{-11} , the FH method takes about 30 s, while the PIC takes about 3300 s, i.e. more than 100 times longer. This is reflected in computing the efficacy (right-bottom panel). As anticipated, the PIC efficacy scales as the inverse of the square root of the CPU time (black line), i.e. it decreases with increasing CPU time. In contrast, the FH efficacy increases by about 4 orders of magnitude when the CPU time increases by a factor of 4 (from 15 to 60 s).

4.2. Landau damping

The Landau damping case is run with the following parameters: $L = 4\pi$, $\alpha_e = \sqrt{2}$, $\varepsilon = 0.05$, $\nu = 0$. The errors are averaged in the time window from $T = 0$ to $T = 3$. After $T = 3$ the PIC reference solution (with $N_{pcel} = 102,400$) is affected by a low noise-to-signal ratio. This is shown in Fig. 6. Here the black line is the PIC reference solution and the blue line is the result from FH with $N_H = 100$. Clearly both results do not suffer from recurrence/noise up to $T = 3$. Additionally, we show with red circles the result from FH when a collisionality $\nu = 2$ is used, with $N_H = 20$. Adding a collisional term allows the elimination of the recurrence problem for FH, even using a low number of Hermite polynomials. Fig. 7 presents the comparison between FH and PIC with the same format of Fig. 5. Once again the correct scaling with the inverse of the number of particles per cell is recovered for PIC. The difference in performance between the two methods is greater than that for the Langmuir wave case. For instance, an error equal to $6 \cdot 10^{-4}$ corresponds to a CPU time of 26 s for FH and about 6400 s for the PIC, with a ratio between the two times approximately equal to 250. Conversely a simulation that takes 200 s achieves errors equal to 10^{-3} and 10^{-9} for PIC and FH, respectively, i.e. FH is 6 orders of magnitude more accurate for equal CPU time. On this test, we have also tried the space filtering (smoothing) described in [54], and verified that the results do not change qualitatively. In Fig. 8, we show the energy (left) and momentum (right) conservation for this test, for different Newton–Krylov absolute and relative tolerances. Red and black lines represent $N_H = 20$ and $N_H = 40$, respectively. The energy and momentum conservation is calculated as the difference between the time-averaged and the initial values for a simulation run with 1000 timesteps. The plots are presented as function of CPU time (note that this does not scale linearly with decreasing tolerance), and the range of tolerances tested is $[10^{-5}, \dots, 10^{-12}]$. For convenience, the absolute and relative tolerances in the Newton–Krylov solver are equal.

4.3. Ion-acoustic wave

The ion-acoustic wave is often used as a benchmark test since it involves multi-scale physics. In fact, it is generated by a perturbation in the ion density only, and has a frequency much lower than the electron plasma frequency, but it still requires accurate representation of the electron dynamics. The parameters for this case are the following: The mass ratio between ions and electrons is equal to 1836; the ratio between electron and ion temperature is equal to 10, $L = 10$, $\alpha_e = \sqrt{2}$, $\varepsilon = 0.2$. The collisionality is $\nu = 1$. The initial perturbation is large enough to drive nonlinear interactions. Fig. 9 shows the time evolution of the amplitude of the first 4 Fourier modes. E_1 is the mode initially excited, and its higher harmonics (E_2 , E_3 , etc.) are excited via wave–wave interactions. The

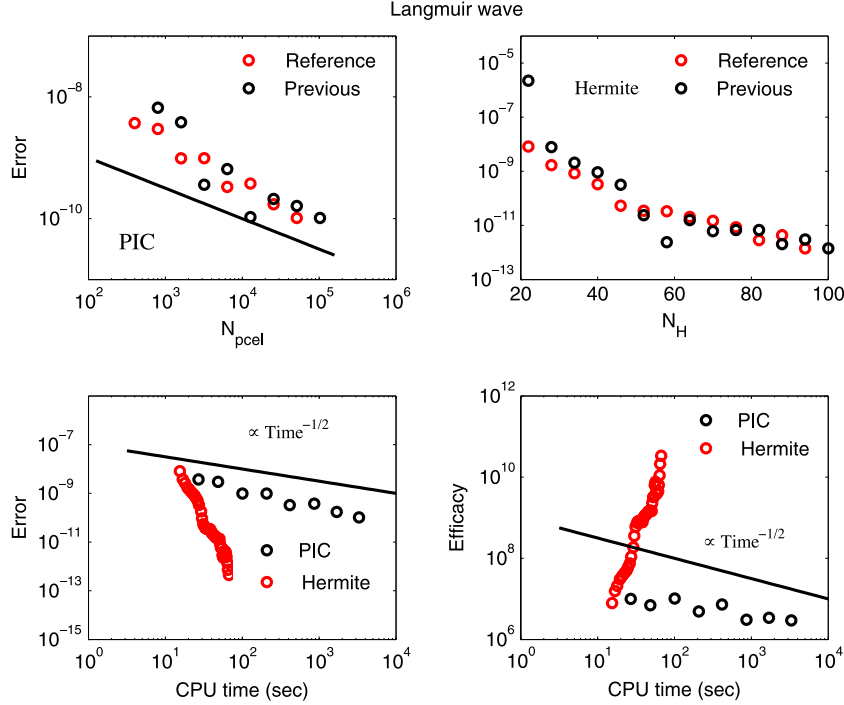


Fig. 5. Langmuir wave with parameters $L = 2\pi$, $\alpha_e = 0.1\sqrt{2}$, $\varepsilon = 0.01$, $\nu = 0$. Top left: PIC simulation; error as a function of number of particles per cell N_{pcel} . Red and black circles represent the error calculated with respect to a reference solution (with $N_{pcel} = 102,400$) and previous less accurate solution, respectively. The black solid line indicates the scaling $N_{pcel}^{-1/2}$. Top right: Hermite simulation; error as a function of number of Hermite modes N_H . Red and black circles represent the error calculated with respect to a reference solution (with $N_H = 100$) and previous less accurate solution, respectively. Bottom left: error as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. Bottom right: efficacy as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

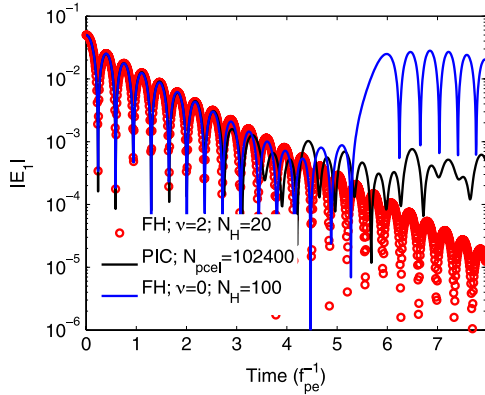


Fig. 6. Landau damping simulation. Black line: reference solution for PIC with $N_{pcel} = 102,400$; blue line: solution for FH with $\nu = 0$, $N_H = 100$; red circle: solution for FH with $\nu = 2$, $N_H = 20$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

red lines are the results from FH with $N_H = 300$ and the black lines are results from the PIC reference solution (with $N_{pcel} = 102,400$). For the fundamental mode (top panel), there is good agreement between FH and PIC, although one can notice a certain amount of noise in the PIC results. The first harmonic E_2 is quite noisy in the PIC results, but there is still a qualitative agreement between the two methods, with the PIC solution having the correct order of magnitude and approximately the correct frequency. However, for the PIC solution, harmonics higher than the second (E_3 and E_4 plotted respectively in the third and last panel) are at the noise ground level and completely miss the correct physical evolution. This is an example where, although the fundamental mode has been excited at a large (nonlinear) amplitude, the PIC method is able to capture only the largest scale fluctuations, completely missing the lower amplitude interactions. We have described a similar shortcoming

of PIC elsewhere [79], for a very different physical phenomenon (the plasma echo). As we anticipated, it is not surprising that the numerical noise known to plague PIC plays a similar role in explicit, semi-implicit and fully-implicit versions of the algorithm, even though one expects the latter to be the most accurate of the three methods.

The performance/efficacy study for the ion-acoustic test is presented in Fig. 10. One can see that this is a much harder test, and in general the errors are larger than in the previous examples for similar N_{pcel} and N_H . For this case, the PIC method recovers the theoretical scaling with N_{pcel} only for the first few data points (top-left panel). This is an indication that even the most accurate solution ($N_{pcel} = 102,400$) taken as a ‘reference’ is, in reality still far from convergence. The FH method is also not yet in the full spectral convergence regime. With regards to the comparison between PIC and FH performance and efficacy, conclusions similar to previous cases hold. Note that this case has a generally longer CPU time with respect, for instance, to the Landau damping case, simply because we have run the simulations for many more timesteps (the ion-acoustic wave frequency is about 60 times lower than the electron plasma frequency). The errors of PIC and FH are of similar orders of magnitude for CPU times smaller than 1000 s (left-bottom panel). However, the gap in errors between the two methods becomes as large as two orders of magnitude for CPU times of about 20,000 s.

4.4. Two-stream instability

The two-stream instability is a classical textbook study often used as a benchmark for kinetic plasma codes. It is a linear instability that is excited when the plasma consists of two populations of particles counter-streaming with a large enough relative speed. We initialize the electron distribution function as

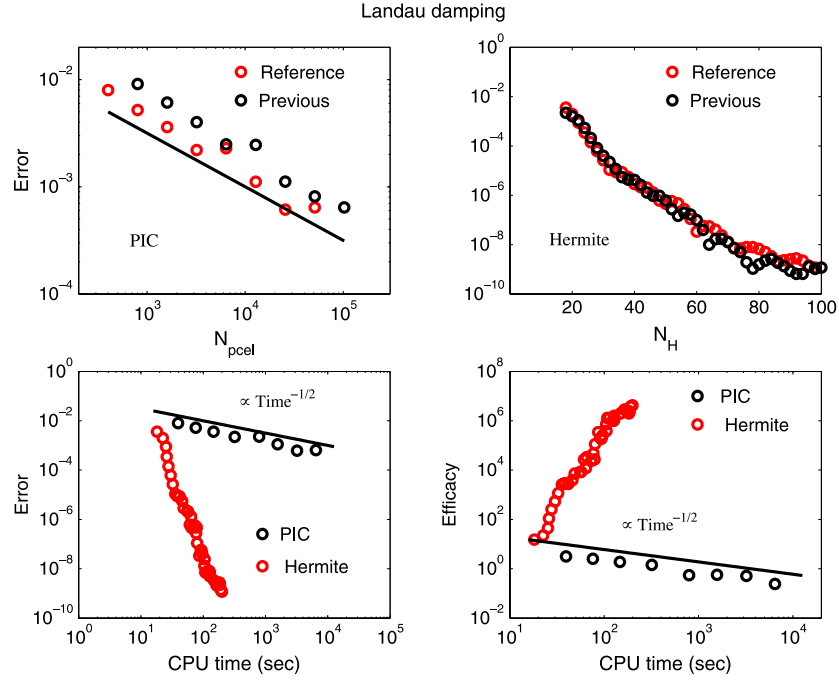


Fig. 7. Landau damping with parameters $L = 4\pi$, $\alpha_e = \sqrt{2}$, $\varepsilon = 0.05$, $\nu = 0$. Top left: PIC simulation; error as a function of number of particles per cell N_{pcel} . Red and black circles represent the error calculated with respect to a reference solution (with $N_{pcel} = 102,400$) and previous less accurate solution, respectively. The black solid line indicates the scaling $N_{pcel}^{-1/2}$. Top right: Hermite simulation; error as a function of number of Hermite modes N_H . Red and black circles represent the error calculated with respect to a reference solution (with $N_H = 100$) and previous less accurate solution, respectively. Bottom left: error as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. Bottom right: efficacy as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

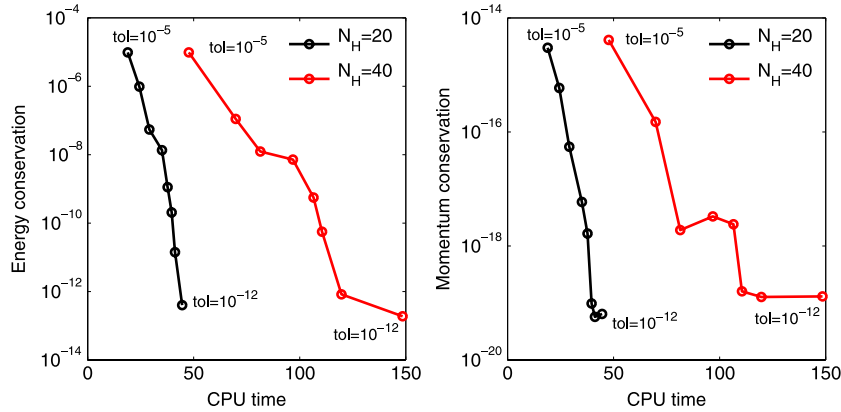


Fig. 8. Energy and momentum conservation for Landau damping with parameters $L = 4\pi$, $\alpha_e = \sqrt{2}$, $\varepsilon = 0.05$, $\nu = 2$. Left: Energy conservation vs. CPU time for tolerances in the range $[10^{-5}, \dots, 10^{-12}]$. Right: Momentum conservation vs. CPU time for tolerances in the range $[10^{-5}, \dots, 10^{-12}]$. Red and black lines are for $N_H = 20$ and 40, respectively. The quantities are calculated as the difference between the time-averaged and the initial values, for a simulation run for 1000 timesteps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

two drifting Maxwellians with equal temperature:

$$f_e = 0.5e^{-\left(\frac{v-U}{\alpha_e}\right)^2} + 0.5e^{-\left(\frac{v+U}{\alpha_e}\right)^2}, \quad (30)$$

with U the drift velocity.

The distribution function in Eq. (30) can be efficiently described in the FH method as two distinct electron populations by setting $u_e = \pm U$ and solving Eq. (17) separately for each drifting Maxwellian. The parameters are as follows: $L = 4\pi$, $\alpha_e = \sqrt{2}/2$, $U = 1$, $\varepsilon = 0.001$, $\nu = 2$. Fig. 11 shows the time evolution of E_1 for PIC solutions with different numbers of particles per cell (black: $N_{pcel} = 400$, blue: $N_{pcel} = 6400$, red: $N_{pcel} = 12,800$, magenta: $N_{pcel} = 102,400$). The onset of the two-stream instability is very sensitive to the initial condition. Although the same linear growth rate is approximately recovered in all cases, the solutions for times

$T < 2$ are completely different. Fig. 12 shows the comparison of performance between FH and PIC, with the same format of previous figures. For a CPU time around 550 s, the FH solution is about 10^7 times more accurate than the PIC solution. Also, the most accurate PIC solution ($N_{pcel} = 51,200$) is less accurate than the least accurate FH solution.

5. Conclusions

We have described a spectral method to numerically solve the Vlasov–Poisson equations that describe the evolution of a collisionless plasma. The velocity and configuration spaces are discretized by means of an Hermite and Fourier basis, respectively. In this paper, we have introduced an implicit Crank–Nicolson discretization in time, which is charge, momentum, and energy conserving. The

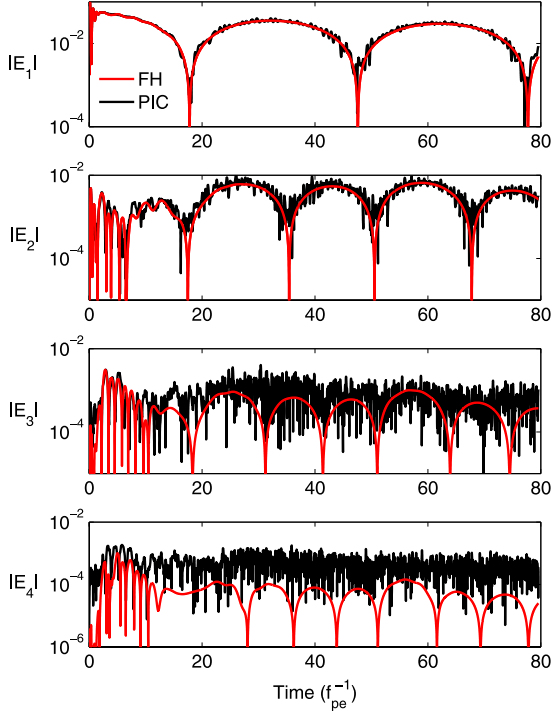


Fig. 9. Ion-acoustic wave simulation with parameters $L = 10$, $\alpha_e = \sqrt{2}$, $\alpha_i = 0.0074\alpha_e$, $\varepsilon = 0.2$, $\nu = 1$. The mass ratio between ion and electron is equal to 1836. The temperature ratio between electrons and ions is equal to 10. Red lines are results from FH with $N_H = 300$. Black lines are results from PIC with $N_{pcel} = 102,400$. The panels from top to bottom represent the amplitude of the modes E_1 – E_4 . E_1 is the mode that is perturbed at $T = 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

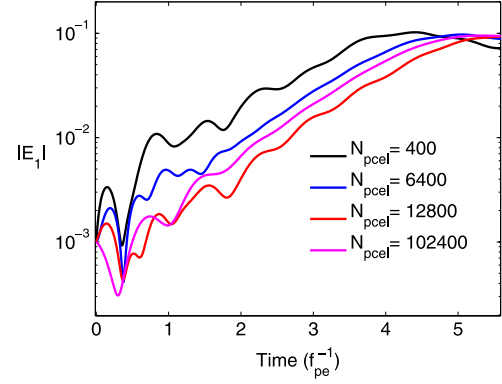


Fig. 11. Two-stream instability simulation with PIC. Black, blue, red, and magenta lines are for $N_{pcel} = 400$, 6400, 12,800, 102,400, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

simultaneous conservation of these three quantities is a very important property of this method, especially when compared with PIC methods. Indeed, no PIC code is able to simultaneously conserve momentum and energy and thus require monitoring of the non-conserved quantity. Of course, other Vlasov schemes focus on the positivity preservation, which is not guaranteed in the FH scheme, but usually at the expenses of energy conservation.

We have compared the performance of the implicit FH method with the recently proposed implicit PIC code for the case of one-dimensional, electrostatic simulations. The comparison results for Langmuir wave, Landau damping, ion-acoustic wave, and two-stream instability are summarized in Figs. 5, 7 and 10, 12, respectively. The two metrics that we have considered in order to fairly assess which method is computationally more advantageous are errors and efficacy as a function of CPU time. In particular, we

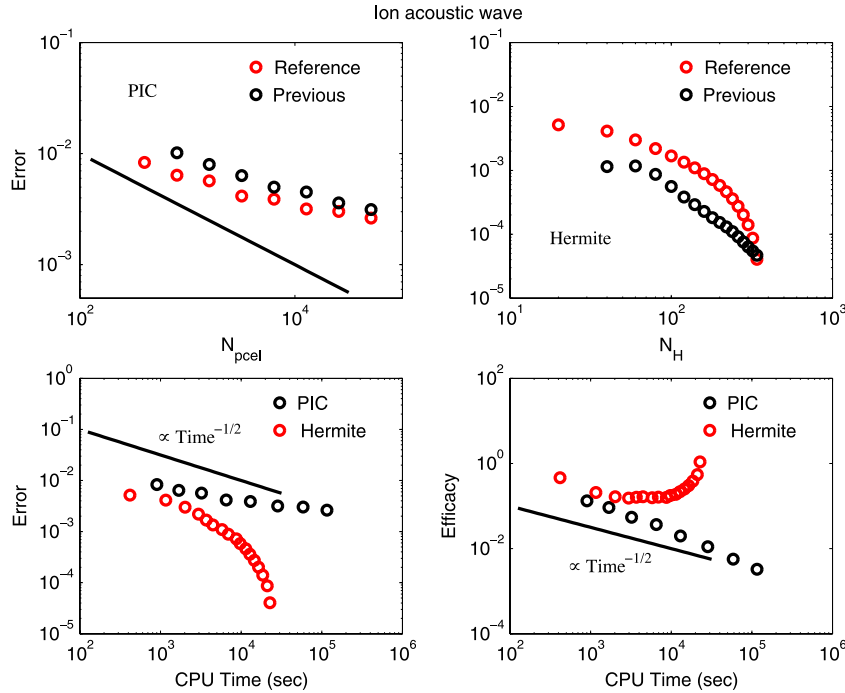


Fig. 10. Ion-acoustic wave with parameters $L = 10$, $\alpha_e = \sqrt{2}$, $\alpha_i = 0.0074\alpha_e$, $\varepsilon = 0.2$, $\nu = 1$. The mass ratio between ion and electron is equal to 1836. The temperature ratio between electrons and ions is equal to 10. Top left: PIC simulation; error as a function of number of particles per cell N_{pcel} . Red and black circles represent the error calculated with respect to a reference solution (with $N_{pcel} = 102,400$) and previous less accurate solution, respectively. The black solid line indicates the scaling $N_{pcel}^{-1/2}$. Top right: Hermite simulation; error as a function of number of Hermite modes N_H . Red and black circles represent the error calculated with respect to a reference solution (with $N_H = 400$) and previous less accurate solution, respectively. Bottom left: error as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. Bottom right: efficacy as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

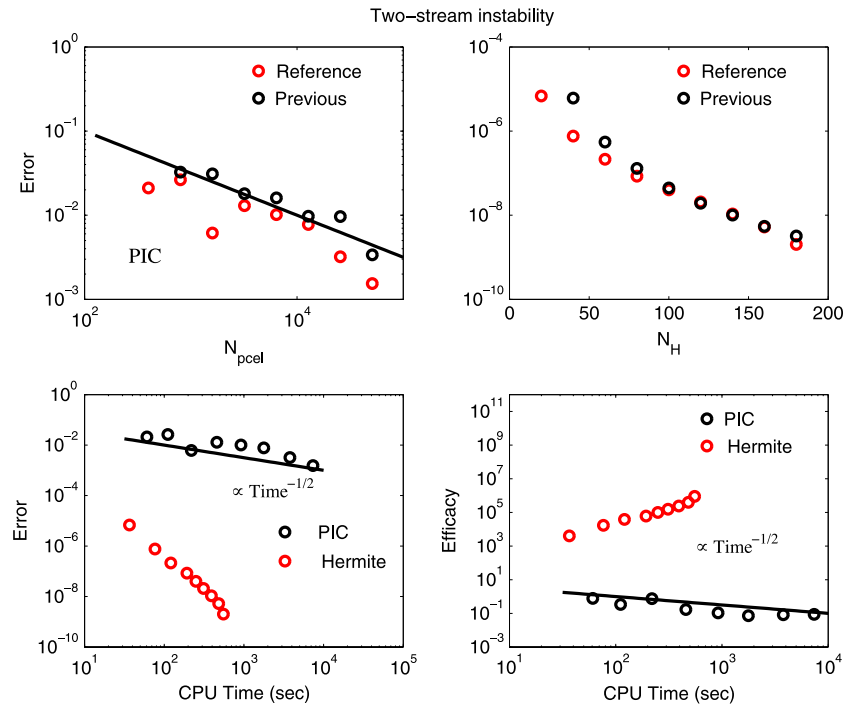


Fig. 12. Two-stream instability with parameters $L = 4\pi$, $\alpha_e = \sqrt{2}/2$, $U = 1$, $\varepsilon = 0.001$, $\nu = 2$. Top left: PIC simulation; error as a function of number of particles per cell N_{pcel} . Red and black circles represent the error calculated with respect to a reference solution (with $N_{pcel} = 102,400$) and previous less accurate solution, respectively. The black solid line indicates the scaling $N_{pcel}^{-1/2}$. Top right: Hermite simulation; error as a function of number of Hermite modes N_H . Red and black circles represent the error calculated with respect to a reference solution (with $N_H = 200$) and previous less accurate solution, respectively. Bottom left: error as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. Bottom right: efficacy as a function of CPU time (in seconds); black circles for PIC, red circles for Hermite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

evaluate the efficacy of a simulation, computed as the inverse of the product of its error (relative to an accurate ‘reference’ solution) and the CPU time to measure the cost-effectiveness of a method or, in other words, how much a larger CPU time is paid off in terms of better accuracy.

Although each simulation is quantitatively different, they all share the same conclusion that the FH method is orders of magnitude more efficient than PIC, at least for the examples considered. We have also shown, for the test case of the ion-acoustic wave, that the PIC code is unable to correctly capture the higher-order harmonics of the excited dominant mode (excited at nonlinear amplitude), making any analysis of wave-wave coupling and energy transfer impossible. This is an important problem that can impact PIC applications in areas such as plasma turbulence (see, e.g. [6,80]), suggesting that denoising techniques are mandatory for PIC there.

We notice that a multidimensional extension of the Hermite basis has been already successfully employed in [51], in a linear context. Further developments of the FH method in terms of optimization of the Hermite basis will be critical for multidimensional applications of this technique, and will be considered in future work. Finally, it is fair to mention that the application of this method to large size real physics problems would likely require some sort of parallelization. It is known that the parallelization of a Fourier basis can badly affect performance. In this respect, local spatial discretizations, such as spectral elements, should also be investigated (and their effect on the conservation properties).

Acknowledgments

The authors would like to thank L. Chacon, S. Markidis, V. Roytershteyn, X. Tang for useful conversations. This work was funded by the Laboratory Directed Research and Development program (LDRD) under the auspices of the National Nuclear

Security Administration of the U.S. Department of Energy by Los Alamos National Laboratory, operated by Los Alamos National Security LLC under contract DE-AC52-06NA25396.

References

- [1] C. Birdsall, A.B. Langdon, *Plasma Physics Via Computer Simulation*, CRC Press, 2005.
- [2] R. Hockney, J. Eastwood, *Computer Simulation Using Particles*, CRC Press, 2010.
- [3] J.P. Verboncoeur, *Plasma Phys. Control. Fusion* 47 (2005) A231.
- [4] F. Valentini, P. Veltri, F. Califano, A. Mangeney, *Phys. Rev. Lett.* 101 (2008) 025006.
- [5] L. Matteini, P. Hellinger, S. Landi, P. Trávníček, M. Velli, *Space Sci. Rev.* 172 (2012) 373–396.
- [6] C.T. Haynes, D. Burgess, E. Camporeale, *Astrophys. J.* 783 (2014) 38.
- [7] E. Camporeale, D. Burgess, *Astrophys. J.* 730 (2011) 114.
- [8] W. Daughton, V. Roytershteyn, H. Karimabadi, L. Yin, B. Albright, B. Bergen, K. Bowers, *Nat. Phys.* 7 (2011) 539–542.
- [9] K. Bowers, B. Albright, L. Yin, W. Daughton, V. Roytershteyn, B. Bergen, T. Kwan, *Journal of Physics: Conference Series*, vol. 180, IOP Publishing, p. 012055.
- [10] J. Chehab, A. Cohen, D. Jennequin, J. Nieto, C. Roland, J. Roche, *Numerical methods for hyperbolic and kinetic problems*, IRMA Lect. Math. Theor. Phys. 7 (2005) 29–42.
- [11] B. Wang, G. Miller, P. Colella, *SIAM J. Sci. Comput.* 33 (2011) 3509–3537.
- [12] W. Deng, G.-Y. Fu, *Comput. Phys. Comm.* 185 (2014) 96–105.
- [13] J. Byers, in: J.P. Boris, R.A. Shanny (Eds.), *Fourth Conference on Numerical Simulation of Plasmas*.
- [14] J. Denavit, J. Walsh, *Comments Plasma Phys. Control. Fusion* 6 (1981) 209–223.
- [15] R. Sydora, *J. Comput. Appl. Math.* 109 (1999) 243–259.
- [16] W. Nevins, G. Hammett, A. Dimits, W. Dorland, D. Shumaker, *Phys. Plasmas* 12 (2005) 122305–122305.
- [17] D. Batchelor, M. Beck, A. Becoulet, R.V. Budny, C.S. Chang, P.H. Diamond, J.Q. Dong, G.Y. Fu, A. Fukuyama, T.S. Hahn, D.E. Keyes, Y. Kishimoto, S. Klasky, L.L. Lao, K. Li, Z. Lin, B. Ludaescher, J. Manickam, N. Nakajima, T. Ozeki, N. Podhorszki, W.M. Tang, M.A. Vouk, R.E. Waltz, S.J. Wang, H.R. Wilson, X.Q. Xu, M. Yagi, F. Zonca, *Plasma Sci. Technol.* 9 (2007) 312. <http://phoenix.ps.uci.edu/zlin/bib/batchelor07.pdf>.
- [18] W. Tang, *Journal of Physics: Conference Series*, vol. 125, IOP Publishing, p. 012047.
- [19] E. Walker, *Computer* 42 (2009) 35–41.
- [20] J. Whealton, R. McGaffey, P. Meszaros, *J. Comput. Phys.* 63 (1986) 20–32.
- [21] W. Guo, J.-M. Qiu, *J. Comput. Phys.* (2012).

- [22] N. Besse, E. Sonnendrücker, *J. Comput. Phys.* 191 (2003) 341–376. [http://dx.doi.org/10.1016/S0021-9991\(03\)00318-8](http://dx.doi.org/10.1016/S0021-9991(03)00318-8).
- [23] R. Heath, I. Gamba, P. Morrison, C. Michler, *J. Comput. Phys.* 231 (2012) 1140–1174.
- [24] J. Banks, J. Hittinger, *IEEE Trans. Plasma Sci.* 38 (2010) 2198–2207.
- [25] R. Duclous, B. Dubroca, F. Filbet, et al., *Discrete Contin. Dyn. Syst. Ser. S* 5 (2012) 283–305.
- [26] M. Shoucri, *Commun. Nonlinear Sci. Numer. Simul.* 13 (2008) 174–182. <http://dx.doi.org/10.1016/j.cnsns.2007.04.004>.
- [27] F. Califano, A. Mangeney, Eulerian Vlasov codes for the Numerical Solution of the Kinetic Equations of Plasmas, Nova Publisher Inc., New York, 2010.
- [28] M. Shoucri, G. Knorr, *J. Comput. Phys.* 14 (1974) 84. [http://dx.doi.org/10.1016/0021-9991\(74\)90006-0](http://dx.doi.org/10.1016/0021-9991(74)90006-0).
- [29] C.Z. Cheng, G. Knorr, *J. Comput. Phys.* 22 (1976) 330–351. [http://dx.doi.org/10.1016/0021-9991\(76\)90053-X](http://dx.doi.org/10.1016/0021-9991(76)90053-X).
- [30] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, *J. Comput. Phys.* 149 (1999) 201–220. <http://dx.doi.org/10.1006/jcph.1998.6148>.
- [31] T. Umeda, M. Ashour-Abdalla, D. Schriver, *J. Plasma Phys.* 72 (2006) 1057. <http://dx.doi.org/10.1017/S0022377806005228>.
- [32] J.A. Carrillo, F. Vecil, *SIAM J. Sci. Comput.* 29 (2007) 1179–1206. <http://dx.doi.org/10.1137/050644549>.
- [33] G. Crouseilles, N. and Latu, E. Sonnendrücker, *Int. J. Appl. Math. Comput. Sci.* 17 (2007) 335–349.
- [34] K. Imadera, Y. Kishimoto, D. Saito, J. Li, T. Utsumi, *J. Comput. Phys.* 228 (2009) 8919–8943. <http://dx.doi.org/10.1016/j.jcp.2009.09.008>.
- [35] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, *J. Comput. Phys.* 229 (2010) 1927–1953. <http://dx.doi.org/10.1016/j.jcp.2009.11.007>.
- [36] J.-M. Qiu, A. Christlieb, *J. Comput. Phys.* 229 (2010) 1130–1149. <http://dx.doi.org/10.1016/j.jcp.2009.10.016>.
- [37] J.-M. Qiu, C.-W. Shu, *J. Comput. Phys.* 230 (2011) 8386–8409. <http://dx.doi.org/10.1016/j.jcp.2011.07.018>.
- [38] J.A. Rossmann, D.C. Seal, *J. Comput. Phys.* 230 (2011) 6203–6232.
- [39] F. Filbet, E. Sonnendrücker, P. Bertrand, *J. Comput. Phys.* 172 (2001) 166–187. <http://dx.doi.org/10.1006/jcph.2001.6818>.
- [40] F. Engelmann, M. Feix, E. Minardi, J. Oxenius, *Phys. Fluids* 6 (1963) 266–275. <http://dx.doi.org/10.1063/1.1706724>.
- [41] T. Armstrong, R. Harding, G. Knorr, D. Montgomery, *Methods Comput. Phys.* 9 (1970) 29.
- [42] J. Denavit, W. Kruer, *Phys. Fluids* 14 (1971) 1782.
- [43] A.J. Klimas, *J. Comput. Phys.* 50 (1983) 270–306.
- [44] B. Eliasson, *J. Comput. Phys.* 190 (2003) 501–522.
- [45] S. Le Bourdieu, F. De Vuyst, L. Jacquet, *Comput. Phys. Comm.* 175 (2006) 528–544.
- [46] A. Mangeney, F. Califano, C. Cavazzoni, P. Travnicek, *J. Comput. Phys.* 179 (2002) 495–538.
- [47] H. Grad, *Comm. Pure Appl. Math.* 2 (1949) 331–407.
- [48] F.C. Grant, M.R. Feix, *Phys. Fluids* 10 (1967) 696–702. <http://dx.doi.org/10.1063/1.1762177>.
- [49] J.P. Holloway, *Transp. Theory Stat. Phys.* 25 (1996) 1–32. <http://dx.doi.org/10.1080/00411459608204828>.
- [50] J. Schumer, J. Holloway, *J. Comput. Phys.* 144 (1998) 626–661.
- [51] E. Camporeale, G.L. Delzanno, G. Lapenta, W. Doughton, *Phys. Plasmas* 13 (2006) 092110. <http://dx.doi.org/10.1063/1.2345358>.
- [52] E. Siminos, D. Bénisti, L. Gremillet, *Phys. Rev. E* 83 (2011) 056402. <http://dx.doi.org/10.1103/PhysRevE.83.056402>.
- [53] J. Vencels, G.L. Delzanno, A. Johnson, I.B. Peng, E. Laure, S. Markidis, *Procedia Comput. Sci.* 51 (2015) 1148–1157.
- [54] G. Chen, L. Chacón, D. Barnes, *J. Comput. Phys.* 230 (2011) 7018–7036.
- [55] R. Mason, *J. Comput. Phys.* 41 (1981) 233–244.
- [56] J. Brackbill, D. Forslund, *J. Comput. Phys.* 46 (1982) 271–308.
- [57] S. Markidis, E. Camporeale, D. Burgess, G. Lapenta, Numerical Modeling of Space Plasma Flows: ASTRONUM-2008, vol. 406, p. 237.
- [58] B. Cohen, A.B. Langdon, A. Friedman, *J. Comput. Phys.* 46 (1982) 15–38.
- [59] A.B. Langdon, B. Cohen, A. Friedman, *J. Comput. Phys.* 51 (1983) 107–138.
- [60] G. Chen, L. Chacon, C. Leibs, D. Knoll, W. Taitano, arXiv:1309.6243 (2013). arXiv preprint.
- [61] S. Markidis, G. Lapenta, *J. Comput. Phys.* 230 (2011) 7037–7052.
- [62] G. Chen, L. Chacon, *Comput. Phys. Comm.* 185 (2014) 2391–2402.
- [63] G. Lapenta, L. Chacón, *J. Comput. Phys.* 219 (2006) 86–103.
- [64] T. Tang, *SIAM J. Sci. Comput.* 14 (1993) 594–606.
- [65] J. Crank, P. Nicolson, *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 43, Cambridge Univ Press, pp. 50–67.
- [66] C. Kelley, *Solving Nonlinear Equations with Newton's Method*, vol. 1, Siam, 2003.
- [67] L. Landau, *J. Phys. (USSR)* 10 (1946) 25–34.
- [68] J. Canosa, J. Gazdag, J. Fromm, *J. Comput. Phys.* 15 (1974) 34–45.
- [69] A.J. Klimas, *J. Comput. Phys.* 68 (1987) 202–226.
- [70] A.J. Klimas, *J. Comput. Phys.* 110 (1994) 150–163.
- [71] A. Lenard, I. Bernstein, *Phys. Rev.* 112 (1958) 1456–1459. <http://link.aps.org/doi/10.1103/PhysRev.112.1456> <http://dx.doi.org/10.1103/PhysRev.112.1456>.
- [72] J. Parker, P. Dellar, *Bull. Am. Phys. Soc.* 57 (2012).
- [73] C. Ng, A. Bhattacharjee, F. Skiff, *Phys. Rev. Lett.* 83 (1999) 1974.
- [74] P. Hilscher, K. Imadera, J. Li, Y. Kishimoto, *Phys. Plasmas* 20 (2013) 082127.
- [75] F. Valentini, F. Califano, P. Veltri, *Phys. Rev. Lett.* 104 (2010) 205002.
- [76] S. Servidio, K. Osman, F. Valentini, D. Perrone, F. Califano, S. Chapman, W. Matthaeus, P. Veltri, *Astrophys. J. Lett.* 781 (2014) L27.
- [77] H.R. Lewis, *J. Comput. Phys.* 6 (1970) 136.
- [78] E. Evstatiev, B. Shadwick, *J. Comput. Phys.* 245 (2013) 376.
- [79] E. Camporeale, G.L. Delzanno, B.K. Bergen, J.D. Moulton, arXiv:1311.2098 (2013).
- [80] S. Servidio, F. Valentini, D. Perrone, A. Greco, F. Califano, W. Matthaeus, P. Veltri, *J. Plasma Phys.* 81 (2014) 1–35.