ELSEVIER

Computer Physics Communications 150 (2003) 247–266

Computer Physics Communications

www.elsevier.com/locate/cpc

Comparison of Eulerian Vlasov solvers

F. Filbet [a,*], E. Sonnendrücker [b]

[a] *IECN, INRIA, Université Henri Poincaré, Nancy, France*
[b] *IRMA, Université Louis Pasteur, Strasbourg, France*

Received 10 August 2001; received in revised form 26 February 2002; accepted 29 June 2002

Abstract

Vlasov methods, which instead of following the particle trajectories, solve directly the Vlasov equation on a grid of phase space have proven to be an efficient alternative to the Particle-In-Cell method for some specific problems. Such methods are useful, in particular, to obtain high precision in regions where the distribution function is small.

Gridded Vlasov methods have the advantage of being completely free of numerical noise, however the discrete formulations contain some other numerical artifacts, like diffusion or dissipation. We shall compare in this paper different types of methods for solving the Vlasov equation on a grid in phase space: the semi-Lagrangian method, the finite volume method, the spectral method, and a method based on a finite difference scheme, conserving exactly several invariants of the system. Moreover, for each of those classes of methods, we shall first compare different interpolation or reconstruction procedures. Then we shall investigate the cost in memory as well as in CPU time which is a very important issue because of the size of the problem defined on the phase space.

PACS: 52.25.Dg; 52.65.-y

Keywords: Vlasov; Eulerian methods; Numerical simulation; Comparison

A model which can be used in many cases for the study of plasma as well as beam propagation is the Vlasov equation coupled with the Maxwell or Poisson equations to compute the self-consistent fields. It describes the evolution of a system of particles under the effects of external and self-consistent fields. The unknown $f(t, x, v)$, depending on the time t, the position x, and the velocity v, represents the distribution of particles in phase space for each species. The numerical resolution of the Vlasov equation is usually performed by Particle-In-Cell (PIC) methods which approximate the plasma by a finite number of particles. Trajectories of these particles are computed from characteristic curves given by the Vlasov equation, whereas self-consistent fields are computed on a mesh of the physical space. This method yields satisfying results with a relatively small number of particles. However, it is well known that the numerical noise inherent to the particle method becomes, in some cases, too important to get an accurate description of the distribution function. Moreover, the numerical noise only decreases in $1/\sqrt{N}$, when the number of particles N is increased.

* Corresponding author.
E-mail address: filbet@math.u-strabg.fr (F. Filbet).

0010-4655/02/$ – see front matter
PII: S0010-4655(02)00694-X

To remedy this problem, methods for discretizing the Vlasov equation on a mesh in phase space have been proposed. Among them, the semi-Lagrangian method consists of computing the distribution function at each grid point by following the characteristic curve ending there. Then to compute the value of the distribution function at the origin of the characteristic, a high order interpolation method is needed. A special case of this method, based on a time splitting which enables an exact computation of the characteristics, was first introduced by Cheng and Knorr [3]. A cubic spline interpolation is used. This algorithm was subsequently applied in many plasma physics papers, see for example [6,9] and references therein. This method was cast into the more general framework of semi-Lagrangian methods by E. Sonnendrücker et al. [12], and successfully adapted to beam physics problems, namely the simulation of space charge waves on an initially semi-Gaussian beam and halo development in a uniform focusing channel [13]. Another flavor of the semi-Lagrangian method was introduced by Nakamura and Yabe and called the Cubic Interpolated Propagation (CIP) method. It is based on a Hermite interpolation for which the gradients of the distribution function are also advanced along the characteristics [14]. It needs the storage of f, $\nabla_x f$, and $\nabla_v f$, therefore in order to reduce memory consumption, the mesh on which it is applied should be coarser.

Another type of scheme for the Vlasov equation is the finite volume type method (or flux balance method), where the discrete unknowns are averages of the distribution function on volumes paving the phase space. These unknowns are updated by considering incoming and outgoing fluxes leading to mass conservation. The first scheme of this type was introduced by Boris and Book [2]. It was further developed by Fijalkow [7]. We recently proposed an improved version of this scheme that we called the Positive and Flux Conservative method (PFC) [8], which is not only conservative, but also preserves the positivity and the maximum value of the distribution function. The scheme was implemented up to third order accuracy.

We shall also consider the Fourier–Fourier spectral method introduced by Klimas and Farell [4,5], based on a time splitting, using at each step a backward and forward Fourier transform for the shift. A filtration algorithm to limit small scale filamentation was also proposed.

One of the flaws common to all Vlasov solvers is their smearing of small structures and the associated non physical increase of entropy. However, this feature is necessary for the stability of Eulerian Vlasov solvers. Indeed, a finite difference scheme based on a method introduced by Arakawa [1], which exactly conserves $\int f^2 \,\mathrm{d}x\,\mathrm{d}v$, becomes unstable when filamentation phenomena occur. The method can still be useful when we stabilize it by a collision term which is chosen so as to still conserve desired moments of the distribution function.

In this paper, we review the methods introduced above, and propose, in particular for the semi-Lagrangian scheme, new interpolation techniques which are local ones and which should be more efficient for parallel computation as they need less communication between processors. Moreover, in order to get some insight on the behavior of these methods we compare them for some classical problems of plasma physics, with respect to numerical dissipation, phase errors and accuracy at lower resolution as well computational cost for one specific problem.

1. The Vlasov equation

The evolution of the density of particles $f(t,x,v)\,\mathrm{d}x\,\mathrm{d}v$ in the phase space $(x,v) \in \mathbb{R}^d \times \mathbb{R}^d$, $d = 1,\ldots,3$, is given by the Vlasov equation,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F(t,x,v) \cdot \nabla_v f = 0, \tag{1}$$

which can also be written in the following conservative form

$$\frac{\partial f}{\partial t} + \mathrm{div}_x(vf) + \mathrm{div}_v\big(F(t,x,v) f\big) = 0. \tag{2}$$

The force field $F(t,x,v)$ can consist of an applied external field and a self-consistent field which is coupled with the distribution function f giving a nonlinear system. Typically $F = \frac{q}{m}(E_{\text{self}} + E_{\text{appl}} + v \times (B_{\text{self}} + B_{\text{appl}}))$, where the self fields are solutions of Maxwell's equations with sources induced by the particles, or in simplified cases which we shall consider in this article, the magnetic field is neglected and the self electric field is computed using the Poisson equation, i.e.

$$E_{\text{self}}(t,x) = -\nabla_x \phi(t,x), \quad -\varepsilon_0 \Delta_x \phi = \rho, \tag{3}$$

where m represents the mass of one particle, q its charge and ρ is defined by

$$\rho(t,x) = q \int_{\mathbb{R}^d} f(t,x,v)\,\mathrm{d}v. \tag{4}$$

2. The Flux Conservative method

The starting point of our algorithm is the Flux Balance method [7]. Discretizing the Vlasov equation in conservative form, we first observe that using a time splitting scheme the algorithm boils down to one-dimensional problems which have the following form,

$$\partial_t f + \partial_x\big(\mathbf{u}(t,x) f\big) = 0. \tag{5}$$

The characteristic curves are the solutions of the differential system corresponding to the transport equation:

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}s}(s) = \mathbf{u}\big(s, X(s)\big), \\ X(t) = x. \end{cases} \tag{6}$$

We denote by $X(s,t,x)$ the solution of (6). Then the conservation of particles along the characteristic curves leads to

$$\int_K f(t,x)\,\mathrm{d}x = \int_{X(s,t,K)} f(s,x)\,\mathrm{d}x, \tag{7}$$

for any interval K, where

$$X(s,t,K) = \big\{ y \in \mathbb{R}:\ y = X(s,t,z);\ z \in K \big\}.$$

Note that this property remains true for dimensions $d \geqslant 1$. Now, let us introduce a finite set of mesh points $(x_{i+1/2})_{i\in I}$ of the computational domain $(x_{\min}, x_{\max})$. We will denote by $\Delta x = x_{i+1/2} - x_{i-1/2}$ the space step, and by $C_i = [x_{i-1/2}, x_{i+1/2}]$ the control volume. Assume the values of the distribution function are known at time $t^n = n\Delta t$. We find the new values at time t^{n+1} using the conservation of particles (7) on each interval C_i from time t^n to time t^{n+1}, we have

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1},x)\,\mathrm{d}x = \int_{X(t^n,t^{n+1},x_{i-1/2})}^{X(t^n,t^{n+1},x_{i+1/2})} f(t^n,x)\,\mathrm{d}x. \tag{8}$$

Denoting by

$$\Phi_{i+1/2}(t^n) = \int_{X(t^n,t^{n+1},x_{i+1/2})}^{x_{i+1/2}} f(t^n,x)\,\mathrm{d}x,$$

the flux conservation becomes

$$\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^{n+1},x)\,\mathrm{d}x = \Phi_{i-1/2}(t^n) + \int_{x_{i-1/2}}^{x_{i+1/2}} f(t^n,x)\,\mathrm{d}x - \Phi_{i+1/2}(t^n). \tag{9}$$

On the one hand, the evaluation of the average of the solution over $[x_{i-1/2}, x_{i+1/2}]$ smears out fine details of the exact solution. Indeed, in the filamentation process, these details will become finer than the size of a cell for any given grid for long time computations.

On the other hand, in order to get a high order scheme, an essential step is now to choose an efficient method to reconstruct the distribution function from the values on each cell C_i.

2.1. The Flux Balance method (FBM)

In [7], Fijalkow only used a linear interpolation

$$f_h(x) = f_i + (x - x_i)\frac{f_{i+1} - f_{i-1}}{2\Delta x}, \quad \forall x \in (x_{i-1/2}, x_{i+1/2}).$$

This method is very straightforward to implement. However, its drawbacks are that it does not give a positive approximation and does not control spurious oscillations.

2.2. The Positive and Flux Conservative method (PFC)

This method was introduced recently in [8]. It is based on a reconstruction via primitive function. Let $F(t^n,x)$ be a primitive of the distribution function $f(t^n,x)$. We will denote by

$$f_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} f(t^n,x)\,\mathrm{d}x,$$

then we have $F(t^n, x_{i+1/2}) - F(t^n, x_{i-1/2}) = \Delta x f_i^n$, and

$$F\big(t^n, x_{i+1/2}\big) = \Delta x \sum_{k=0}^{i} f_k^n = w_i^n.$$

On the interval $[x_{i-1/2}, x_{i+1/2}]$, we use the stencil $\{x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$ to approximate the primitive by a polynomial of degree three. By differentiation, we define a first approximation $\tilde{f}_h(t^n,x)$, which is a third-order approximation of the distribution function $f(t^n,x)$. However, it does not retain the property of the exact solution of the Vlasov equation that $0 \leqslant f(t,x) \leqslant f_\infty$ for all x and t, where f_∞ is the maximum value of the initial distribution function f_0. In order to enforce this property, we introduce slope correctors to obtain for all $x \in C_i$,

$$\begin{aligned} f_h(t^n,x) \;=\; & f_i^n + \frac{\epsilon_i^+}{6\Delta x^2}\big[2(x-x_i)(x-x_{i-3/2}) + (x-x_{i-1/2})(x-x_{i+1/2})\big]\big(f_{i+1}^n - f_i^n\big) \\ & - \frac{\epsilon_i^-}{6\Delta x^2}\big[2(x-x_i)(x-x_{i+3/2}) + (x-x_{i-1/2})(x-x_{i+1/2})\big]\big(f_i^n - f_{i-1}^n\big), \end{aligned}$$

with

$$\epsilon_i^+ = \begin{cases} \min(1;\, 2f_i^n/(f_{i+1}^n - f_i^n)) & \text{if } f_{i+1}^n - f_i^n > 0, \\ \min(1;\, -2(f_\infty - f_i^n)/(f_{i+1}^n - f_i^n)) & \text{if } f_{i+1}^n - f_i^n < 0, \end{cases}$$

and

$$\epsilon_i^- = \begin{cases} \min(1; 2(f_\infty - f_i^n)/(f_i^n - f_{i-1}^n)) & \text{if } f_i^n - f_{i-1}^n > 0, \\ \min(1; -2f_i^n/(f_i^n - f_{i-1}^n)) & \text{if } f_i^n - f_{i-1}^n < 0. \end{cases}$$

It is easy to check that the approximation of the distribution function $f_h(t^n, x)$ previously constructed satisfies

- Conservation of the average: for all $i \in I$, $\int_{x_{i-1/2}}^{x_{i+1/2}} f_h(t^n, x)\,\mathrm{d}x = \Delta x f_i^n$.
- Maximum principle: for all $x \in (x_{\min}, x_{\max})$, $0 \leqslant f_h(t^n, x) \leqslant f_\infty$.

From this reconstruction, we approximate the quantity $\Phi_{i+1/2}(t^n)$, by looking for the cell C_j such that $X(t^n, t^{n+1}, x_{i+1/2}) \in C_j$ and setting $\alpha_i = x_{j+1/2} - X(t^n, t^{n+1}, x_{i+1/2})$. Then for a positive $u(t, x)$, we obtain

$$\begin{aligned} \Phi_{i+1/2}(t^n) &= \int_{x_{j+1/2}-\alpha_i}^{x_{i+1/2}} f(t^n, x)\,\mathrm{d}x \\ &= \Delta x \sum_{k=j+1}^{i} f_k^n + \alpha_i \Bigg[f_j^n + \frac{\epsilon_j^+}{6}\left(1 - \frac{\alpha_i}{\Delta x}\right)\left(2 - \frac{\alpha_i}{\Delta x}\right)(f_{j+1}^n - f_j^n) \\ &\qquad + \frac{\epsilon_j^-}{6}\left(1 - \frac{\alpha_i}{\Delta x}\right)\left(1 + \frac{\alpha_i}{\Delta x}\right)(f_j^n - f_{j-1}^n)\Bigg], \end{aligned}$$

and when $\mathbf{u}(t, x)$ is negative, we set $\alpha_i = x_{j-1/2} - X(t^n, t^{n+1}, x_{i+1/2})$, then $-\Delta x \leqslant \alpha_i \leqslant 0$ and

$$\begin{aligned} \Phi_{i+1/2}(t^n) &= \int_{x_{j-1/2}-\alpha_i}^{x_{i+1/2}} f(t^n, x)\,\mathrm{d}x \\ &= \Delta x \sum_{k=i+1}^{j-1} f_k^n + \alpha_i \Bigg[f_j^n - \frac{\epsilon_j^+}{6}\left(1 - \frac{\alpha_i}{\Delta x}\right)\left(1 + \frac{\alpha_i}{\Delta x}\right)(f_{j+1}^n - f_j^n) \\ &\qquad - \frac{\epsilon_j^-}{6}\left(2 + \frac{\alpha_i}{\Delta x}\right)\left(1 + \frac{\alpha_i}{\Delta x}\right)(f_j^n - f_{j-1}^n)\Bigg]. \end{aligned}$$

3. The semi-Lagrangian method (SL)

This method is based on the usual advective form of the Vlasov equation, which reads

$$\frac{\partial f}{\partial t} + v.\nabla_x f + F(t, x).\nabla_v f = 0, \tag{10}$$

where F is the force field. The distribution function solution of the Vlasov equation is constant along the particle trajectories. So, assuming it is known at time $t^n = n\Delta t$, the solution at time t^{n+1} is given by

$$f(t^{n+1}, x, v) = f\big(t^n, X(t^n, t^{n+1}, x, v), V(t^n, t^{n+1}, x, v)\big), \tag{11}$$

where $(X(t^n, t^{n+1}, x, v), V(t^n, t^{n+1}, x, v))$ stands for the solution of the differential system defining the characteristic curves which reads

$$\frac{\mathrm{d}X}{\mathrm{d}t} = V(t),$$
$$\frac{\mathrm{d}V}{\mathrm{d}t} = F(X(t), t).$$

In the semi-Lagrangian method the distribution function is approximated at each grid point of the computational domain $(\mathbf{x}_i, \mathbf{v}_i)_{i\in I}$. It is updated at each time step from its value at the origin of the characteristic $(X(t^n, t^{n+1}, \mathbf{x}_i, \mathbf{v}_i), V(t^n, t^{n+1}, \mathbf{x}_i, \mathbf{v}_i))$, which is computed from values on the grid using a high order interpolation method. In previous works [3,12], a cubic spline interpolation was used. This gives very good results, but has the drawback of being non local which causes a higher communication overhead on parallel computers. We want here to compare its properties with those of local interpolation procedures.

Actually, the semi-Lagrangian method for the Vlasov equation can be simplified a lot when a splitting procedure is used, because in this case the feet of the characteristics can be computed explicitly at each split step. This leads to the following algorithm step t^n to t^{n+1} are the following:

$$\begin{aligned} f^*(x, v) &= f(t^n, x - v\Delta t/2, v), \\ f^{**}(x, v) &= f^*\big(x, v - E^*(x)\Delta t\big), \\ f(t^{n+1}, x, v) &= f^{**}(x - v\Delta t/2, v), \end{aligned} \tag{12}$$

where $E^*(x)$ is computed from f^*.

For simplicity, we will only consider one dimensional reconstruction, but it can be easily generalized to higher dimensions. In the following discussion, we will assume the distribution function is known at time t^n on the grid:

$$f\big(t^n, \mathbf{x}_i\big) = f_i^n, \quad \forall i \in I,$$

and will present two methods of reconstruction based on the Lagrange and Hermite interpolation.

3.1. The Lagrange interpolation method

We look for a continuous approximation f of $f(t^n, .)$ such that

$$\forall i \in I, \quad f(\mathbf{x}_i) = f_i^n, \quad \text{and} \quad \forall x \in [\mathbf{x}_i, \mathbf{x}_{i+1}], \quad f(x) = q_m(x),$$

where $q_m(x)$ belongs to $P_{2m+1}[\mathbf{x}_i, \mathbf{x}_{i+1}]$, i.e. the set of polynomial of degree $2m+1$ on the interval $[\mathbf{x}_i, \mathbf{x}_{i+1}]$. We only choose polynomials of odd degree to have a centered approximation, indeed the set of points used to construct the polynomial $q_m(x)$ on the interval $[\mathbf{x}_i, \mathbf{x}_{i+1}]$ is

$$\{\mathbf{x}_{i-m}, \ldots, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+1+m}\},$$

and $q_m(x)$ is in the following form

$$q_m(x) = f_{i-m}^n + \sum_{k=1}^{2m+1} f[\mathbf{x}_{i-m}, \ldots, \mathbf{x}_{i-m+k}] \prod_{l=0}^{k} (x - \mathbf{x}_{i-m+l}),$$

where $f[\mathbf{x}_{i-m}, \ldots, \mathbf{x}_{i-m+k}]$ is given by the divided difference formula

$$\begin{aligned} f[\mathbf{x}_i, \ldots, \mathbf{x}_{i+p}] &= \frac{1}{p!} \frac{f[\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+p}] - f[\mathbf{x}_i, \ldots, \mathbf{x}_{i+p-1}]}{\mathbf{x}_{i+p} - \mathbf{x}_i}, \\ f[\mathbf{x}_i] &= f(\mathbf{x}_i). \end{aligned} \tag{13}$$

This interpolation method only gives a continuous function, then we have to consider a high degree polynomial (m greater than 2) to obtain an accurate approximation of the distribution function. It has been implemented until $m = 4$.

From this reconstruction, we define the approximation of the distribution function $f(t^{n+1}, \mathbf{x}_i)$ at time t^{n+1} at each grid point as follows:

for simplicity take $m = 2$, assume $X(t^n, t^{n+1}, \mathbf{x}_i)$ is known and belongs to the interval $[\mathbf{x}_j, \mathbf{x}_{j+1}]$, then let us denote by $\alpha_i = [X(t^n, t^{n+1}, \mathbf{x}_i) - \mathbf{x}_j]/\Delta x$, with $\Delta x = \mathbf{x}_{j+1} - \mathbf{x}_j$,

$$\begin{aligned} f(t^{n+1}, \mathbf{x}_i) &= q_2\big(X(t^n, t^{n+1}, \mathbf{x}_i)\big) \\ &= f_j^n + \alpha_i\big[f_{j+1}^n - f_j^n\big] - \frac{1}{2}\alpha_i(1-\alpha_i)\big[f_{j+1}^n - 2f_j^n + f_{j-1}^n\big] \\ &\quad - \frac{1}{6}\alpha_i(1-\alpha_i)(1+\alpha_i)\big[f_{j+2}^n - 3f_{j+1}^n + 3f_j^n - f_{j-1}^n\big] \\ &\quad + \frac{1}{24}\alpha_i(1-\alpha_i)(1+\alpha_i)(2-\alpha_i)\big[f_{j+2}^n - 4f_{j+1}^n + 6f_j^n - 4f_{j-1}^n + f_{j-2}^n\big] \\ &\quad + \frac{1}{120}\alpha_i(1-\alpha_i)(1+\alpha_i)(2-\alpha_i)(2+\alpha_i) \\ &\quad \times \big[f_{j+3}^n - 5f_{j+2}^n + 10f_{j+1}^n - 20f_j^n + 5f_{j-1}^n - f_{j-2}^n\big]. \end{aligned}$$

In the general situation, the semi-Lagrangian method does not conserve global mass, but for linear advection with constant coefficients, the use of a centered approximation ensures the conservation of global mass: for simplicity, assume the propagation velocity u is positive,

$$f\big(t^{n+1}, \mathbf{x}_i\big) = f\big(t^n, \mathbf{x}_i - u\Delta t\big),$$

we set $j = [\frac{u\Delta t}{\Delta x}]$, where [.] represents the integer part, and $0 \leqslant \alpha = u\Delta t - \mathbf{x}_{i-j} \leqslant \Delta x$, then for the previous scheme, we have

$$\sum_i f_i^{n+1} = \sum_i f_{i-j}^n + \alpha \sum_i \big[f_{i-j+1}^n - f_{i-j}^n\big] + \frac{1}{2}\alpha(1-\alpha)\sum_i \big[f_{i-j+1}^n - 2f_{i-j}^n + f_{i-j-1}^n\big] + \cdots,$$

using the divided difference formula, we obtain the result

$$\sum_i f_i^{n+1} = \sum_i f_{i-j}^n = \sum_i f_i^n.$$

3.2. The Hermite interpolation method

In this section, we will only consider cubic polynomials to construct a C^1 approximation $f(t^n, .)$ using a Hermite interpolation, which needs to estimate the derivative $\partial_x f(x)$. In [14], the authors treated the case when the propagating velocity is constant and proposed to approximate the profile of the derivative by differentiating the equation. They finally obtained a transport equation for f, and $\partial_x f$, but the memory cost is increased to treat the Vlasov equation in the (x, v) space. Here, we propose to approximate the derivative by a fourth-order accurate finite difference formula:

$$\partial_x f_i^n = \frac{1}{12\Delta x}\big[8\big[f_{i+1}^n - f_{i-1}^n\big] - \big[f_{i+2}^n - f_{i-2}^n\big]\big].$$

Then, for all $x \in [\mathbf{x}_i, \mathbf{x}_{i+1}]$, $f(x)$ is given by the cubic polynomial $p_3(x)$ such that

$$\begin{aligned} &p_3(\mathbf{x}_i) = f_i^n, && \partial_x p_3(\mathbf{x}_i) = \partial_x f_i^n, \\ &p_3(\mathbf{x}_{i+1}) = f_{i+1}^n, && \partial_x p_3(\mathbf{x}_{i+1}) = \partial_x f_{i+1}^n. \end{aligned}$$

Setting $\alpha_i = [X(t^n, t^{n+1}, \mathbf{x}_i) - \mathbf{x}_j]/\Delta x$, where $X(t^n, t^{n+1}, \mathbf{x}_i)$ belongs to $[\mathbf{x}_j, \mathbf{x}_{j+1}]$, the value at time t^{n+1} is determined by

$$\begin{aligned} f\big(t^{n+1},\mathbf{x}_i\big) &= p_3\big(X(t^n,t^{n+1},\mathbf{x}_i)\big) \\ &= f_i^n+\alpha_i\big[f_{i+1}^n-f_i^n\big]+\alpha_i^2\big[3\big[f_{i+1}^n-f_i^n\big]-\Delta x\big[2\partial_x f_i^n-\partial_x f_{i+1}^n\big]\big] \\ &\quad +\alpha_i^3\big[\Delta x\big[\partial_x f_{i+1}^n+\partial_x f_i^n\big]-2\big[f_{i+1}^n-f_i^n\big]\big]. \end{aligned}$$

Let us note that using the centered approximation of the derivative, we also prove the conservation of global mass for the linear advection with constant coefficients.

3.3. The CIP method

This method, which is a variant of the semi-Lagrangian method, was developed by Nakamura and Yabe [14]. It is based on a splitting procedure into one-dimensional advection equations. The interpolation step in the semi-Lagrangian algorithm is performed using a cubic Hermite interpolation, i.e. an interpolation using on each interval the values of the functions and its derivatives at the endpoints of the interval. Its specificity is that the derivatives needed for such an interpolation procedure are not computed numerically but advanced themselves along the one-dimensional characteristics. For example, the first split step of the Vlasov–Poisson approximation consists in solving the following system,

$$\frac{\partial f}{\partial t}+v\frac{\partial f}{\partial x}=0, \tag{14}$$

$$\frac{\partial(\partial_x f)}{\partial t}+v\frac{\partial(\partial_x f)}{\partial x}=0, \tag{15}$$

$$\frac{\partial(\partial_v f)}{\partial t}+\frac{\partial(v\partial_v f)}{\partial x}=0, \tag{16}$$

where the unknowns are $(f,\partial_x f,\partial_v f)$.

Steps (14) and (15) are linear advections of f and $\partial_x f$ which can be solved exactly. However step (16) requires a numerical procedure which needs to be performed with great care so as not to ruin the accuracy and conservation properties of the algorithm [14]. Once f and its derivatives are advanced, the reconstruction step of the semi-Lagrangian algorithm can be performed using a Hermite interpolation.

This scheme has the advantage of being local, which is a nice feature for parallel computations, as it involves less inter-processor communications, however it has a higher memory cost as it needs storing all the derivatives as well as the values of distribution function at each grid point.

4. A spectral method

This method was proposed by Klimas and Farell to approximate the one-dimensional Vlasov–Poisson and Vlasov–Maxwell system [4,5].

The distribution function is approximated by a partial sum of a Fourier series

$$f_N(t,x,v)=\sum_{k=-N}^{N}\hat{f}_k(t,v)\exp(-\mathrm{i}2\pi k\cdot x/L),\quad \forall x\in(0,L);$$

and the Fourier coefficients are given by

$$\hat{f}_k(t,v)=\sum_{k=-N}^{N}f(t,x_j,v)\exp(\mathrm{i}2\pi k\cdot x_j/L);\quad \forall k\in\{-N,\dots,N\}.$$

Then, the first shift of (12) is equivalent to the application of a phase shift to the expansion coefficients

$$\hat{f}_k^*(v)=\hat{f}_k(t^n,v)\exp(-\mathrm{i}2\pi k\cdot v\Delta t/2L).$$

In short, each split step consists in performing a forward FFT, a phase shift, and a backward FFT.

The development of filamentation in the velocity distribution, with its related propagation to large Fourier modes in v in the Fourier–Fourier transformed distribution, can lead to serious computational difficulty. Indeed high frequencies induce strong numerical oscillations. To remedy this problem, a filtered solution can be computed by eliminating high frequencies of the velocity distribution, which gives a smoother approximation [4,5].

5. A Finite Difference Method (FDM)

In 1966 Arakawa [1] introduced a finite difference method for the integration of the Euler equation for two-dimensional fluid flow which can, in particular, achieve conservation of mean kinetic energy and mean square vorticity. This scheme can be easily adapted to the Vlasov–Poisson equations. In 1D, it reads in a dimensionless form

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + \frac{\partial \varphi}{\partial x}\frac{\partial f}{\partial v} = 0, \qquad \Delta\varphi = \int_{\mathbb{R}} f\,\mathrm{d}v - 1. \tag{17}$$

Setting

$$\psi = \varphi - \frac{v^2}{2} \quad \text{and} \quad J(\psi, f) = \frac{\partial \psi}{\partial x}\frac{\partial f}{\partial v} - \frac{\partial \psi}{\partial v}\frac{\partial f}{\partial x},$$

the Vlasov equation reads

$$\frac{\partial f}{\partial t} + J(\psi, f) = 0,$$

which is the form used by Arakawa who gives a second and fourth order discretization of $J(\psi, f)$, with the following conservation properties:

- Particle conservation:
$$\int_{\mathbb{R}^2} J_h(\psi, f)\,\mathrm{d}x\,\mathrm{d}v = 0 \Longrightarrow \int_{\mathbb{R}^2} f(t)\,\mathrm{d}x\,\mathrm{d}v = \int_{\mathbb{R}^2} f_0\,\mathrm{d}x\,\mathrm{d}v.$$
- Energy conservation:
$$\int_{\mathbb{R}^2} J_h(\psi, f)\psi\,\mathrm{d}x\,\mathrm{d}v = 0 \Longrightarrow \int_{\mathbb{R}^2} f(t)\psi(t)\,\mathrm{d}x\,\mathrm{d}v = \int_{\mathbb{R}^2} f_0\psi(0)\,\mathrm{d}x\,\mathrm{d}v.$$
- Conservation of the mean square of f:
$$\int_{\mathbb{R}^2} J_h(\psi, f) f\,\mathrm{d}x\,\mathrm{d}v = 0 \Longrightarrow \int_{\mathbb{R}^2} f^2(t)\,\mathrm{d}x\,\mathrm{d}v = \int_{\mathbb{R}^2} f_0^2\,\mathrm{d}x\,\mathrm{d}v.$$

Then, we first compute three approximations of $J(\psi, f)$ on the grid $(x_i, y_j)_{i,j}$. Denoting by $h = x_{i+1} - x_i = y_{j+1} - y_j$, we obtain

$$\begin{aligned} J^1_{i,j}(\psi, f) &= \frac{1}{4h^2}\big[(\psi_{i+1,j} - \psi_{i-1,j})(f_{i,j+1} - f_{i,j-1}) - (\psi_{i,j+1} - \psi_{i,j-1})(f_{i+1,j} - f_{i-1,j})\big], \\ J^2_{i,j}(\psi, f) &= \frac{1}{4h^2}\big[\psi_{i+1,j}(f_{i+1,j+1} - f_{i+1,j-1}) - \psi_{i-1,j}(f_{i-1,j+1} - f_{i-1,j-1}) \\ &\qquad - \psi_{i,j+1}(f_{i+1,j+1} - f_{i-1,j+1}) + \psi_{i,j-1}(f_{i+1,j-1} - f_{i-1,j-1})\big], \end{aligned}$$

$$J^3_{i,j}(\psi, f) = \frac{1}{4h^2}\big[\psi_{i+1,j+1}(f_{i,j+1} - f_{i+1,j}) - \psi_{i-1,j-1}(f_{i-1,j} - f_{i,j-1}) - \psi_{i-1,j+1}(f_{i,j+1} - f_{i-1,j}) + \psi_{i+1,j-1}(f_{i+1,j} - f_{i,j-1})\big].$$

Finally, the approximation $J_h(\psi, f)$ is given by computing the average of the above three approximations, allowing to conserve the total mass, the total energy and the L^2 norm of f. However, it does not preserve positivity. Moreover the scheme becomes oscillatory when filaments develop on the order of the cell size. The previously introduced Vlasov solvers, rely on the interpolation procedures to numerically smear out the thin filament and thus damp the oscillations. We can here achieve the same effect by adding to the Vlasov equation a collision term, the collision frequency being of the same order as the cell size. Following Robert and Someria [11], we can compute this term so that it maximizes local entropy and conserves moments up to any desired order. In the actual code, we have implemented the conservation of moments up to the second order, i.e. mass, impulsion and kinetic energy. More precisely, we look for a collision model of the form

$$\frac{\partial f}{\partial t} = \frac{\partial \mathcal{J}}{\partial v},$$

where $\mathcal{J}$ should be chosen such that at each point in x: collisions maximize entropy

$$S(t,x) = \int_{\mathbb{R}} f \log f \, \mathrm{d}v,$$

conserve total number of particles, impulsion, kinetic energy:

$$\int_{\mathbb{R}} \frac{\partial \mathcal{J}}{\partial v} \begin{pmatrix} 1 \\ v \\ v^2 \\ \vdots \\ v^K \end{pmatrix} \mathrm{d}v = 0.$$

Moreover, for a given norm

$$\|\mathcal{J}\| = \int_{\mathbb{R}} \frac{\mathcal{J}^2}{f} \, \mathrm{d}v,$$

we require $\|\mathcal{J}\| = \alpha$, where the parameter α is linked to the collision frequency which is here of order to the grid size. Using a Lagrange multiplier technique, these constraints yield

$$\mathcal{J} = \alpha\left(\frac{\partial f}{\partial v} + \sum_{k=1}^{K} A_k k f v^{k-1}\right), \quad \text{with } A_k = (m-1)\frac{\int_{\mathbb{R}} f v^{m-2} \, \mathrm{d}v}{\int_{\mathbb{R}} f v^{m+k-2} \, \mathrm{d}v}.$$

In particular for $K = 2$, our model reads

$$\frac{\partial f}{\partial t} + J(\psi, f) = \alpha \frac{\partial}{\partial v}\left(\frac{\partial f}{\partial v} + A_1 f - A_2 f v\right),$$

with $A_1 = \frac{u_0}{\epsilon - u_0^2/n}$, and $A_2 = \frac{n}{\epsilon - u_0^2/n}$, where

$$n = \int_{\mathbb{R}} f \, \mathrm{d}v, \quad u_0 = \frac{1}{n}\int_{\mathbb{R}} f v \, \mathrm{d}v, \quad \epsilon = \int_{\mathbb{R}} f v^2 \, \mathrm{d}v.$$

6. Numerical tests

6.1. The linear advection

Let us first consider the problem of linear advection:

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = 0, \quad \forall x \in [-\pi, \pi], \text{ and } f(t, -\pi) = f(t, \pi). \tag{18}$$

One can analyse the schemes using a discrete Fourier transform

$$f_j^n = \sum_{k=0}^{N-1} \hat{f}_k^n \mathrm{e}^{\mathrm{i}kx_j}, \quad \text{where } \hat{f}_k^n = \sum_{j=0}^{N-1} f_j^n \mathrm{e}^{-\mathrm{i}kx_j}.$$

Then, the solution in the Fourier space is given by

$$\hat{f}_k^n = \hat{f}_k^0 \mathrm{e}^{\mathrm{i}kvt^n}. \tag{19}$$

In the general situation, Eq. (19) is not satisfied by the algorithm, then it is valuable to give the types of numerical errors which can occur,

- the amplitude error $|\hat{f}_k^n / \hat{f}_k^0|$: the harmonic must decay to stabilize the algorithm, which introduces numerical diffusion. These errors are usually most important for short wave length harmonics.
- the phase error $|vt^n - \mathrm{Arg}(\hat{f}_k^n / \hat{f}_k^0)|$: it is generally called dispersion and describes the error of harmonics which propagate at the wrong speed. The errors are usually increasing with the wave number k.

Let us first consider the amplification factor for the different methods (see Fig. 1). We observe that methods using a smooth reconstruction (Hermite or spline) are less dissipative than those using only a continuous interpolation. To obtain a similar amplification factor with the Lagrange interpolation as with the spline interpolation, a polynomial of degree nine is required. The dissipation of the conservative method with a quadratic polynomial is identical to the one using cubic Lagrange interpolation. The linear reconstruction used in the (FBM) is the most dissipative. Consider now the phase errors (see Fig. 2). It is the most important for the semi-Lagrangian method using a Hermite reconstruction with a second or fourth order approximation of the derivative. The spline reconstruction is also less accurate than the Lagrange interpolation of degree nine.

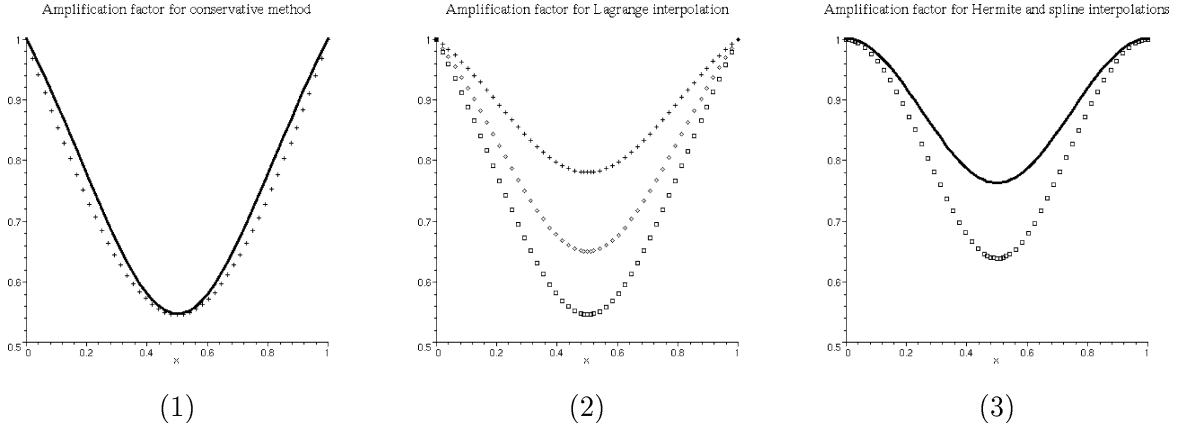

Fig. 1. The amplification factor with respect to α for a fixed mode k. (1) The FBM (cross) and third order reconstruction without slope corrector (line); (2) the semi-Lagrangian method with a Lagrange interpolation of degree 3 (box), 5 (diamond), and 9 (cross); and (3) with cubic Hermite polynomial with a fourth-order approximation of the derivative (box), and cubic spline interpolation (line).

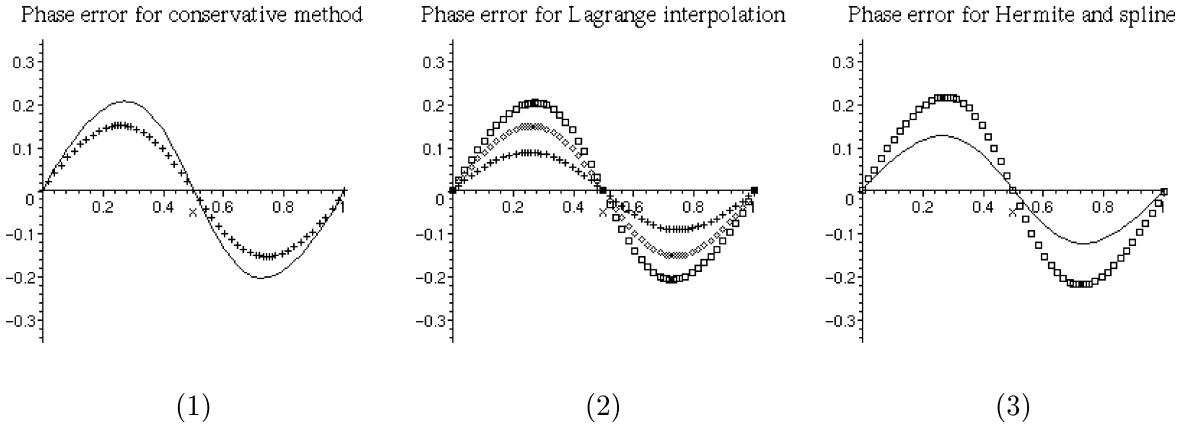

Fig. 2. The phase error with respect to α for a fixed mode k. (1) The conservative method for the FBM (cross) and third order reconstruction without slope corrector (line); (2) the semi-Lagrangian method with a Lagrange interpolation of degree 3 (box), 5 (diamond), and 9 (cross); and (3) with cubic Hermite polynomial with a fourth-order approximation of the derivative (box), and cubic spline interpolation (line).

6.2. The one-dimensional Vlasov–Poisson system

6.2.1. The plasma echo

Following the work by Manfredi et al. [10], we consider an initial data representing an homogeneous Maxwellian distribution

$$f(0,x,v)=\frac{1}{\sqrt{2\pi}}\exp(-v^2/2),\quad \forall (x,v)\in(0,L)\times\mathbb{R},$$

with $L=2\pi/k$ and $k=0.483$. At time $t=0$, we excite an external electric field in the plasma, of the form

$$E_0(x)=\alpha\cos(kx),$$

where $\alpha=0.1$. This field induces a velocity modulation, and right after a density modulation, which eventually decays by Landau damping. After the first has damped away, we launch a second wave at time $t=30\omega_p^{-1}$ of the form

$$E_1(x)=\alpha\cos(2kx).$$

The density modulation induced by this second pulse also fades away. However, after a time much longer than the inverse Landau damping rate of the first two pulses, a third wave appears (the echo) as a modulation of the density at the wave number $k_{\text{echo}}=2k-k=k$. The echo is due to the nonlinear interaction between the two pulses and is essentially a phenomenon of beating between two waves. Fig. 3 shows the electrostatic energy as a function of time. The damping of the two pulses and the subsequent echo are accurately reproduced with the different schemes. The echo wave number is indeed $k_{\text{echo}}=k$ as predicted by the theory. The Landau damping rate for the first pulse is in good agreement with the theoretical value $\gamma_L=0.4\omega_p^{-1}$ and even larger for the second pulse. The echo time is

$$t_{\text{echo}}=\frac{2k}{2k-k}30\omega_p^{-1}=60\omega_p^{-1},$$

which corresponds very well with the numerical value. From time $t=30\omega_p^{-1}$ to $t\simeq 60\omega_p^{-1}$, the second wave has no effect on the first mode of the electric field, but at time $t=60\omega_p^{-1}$, it is strongly perturbed by the echo effect.

We report the results of a simulation using a number of cells $N_x=32$ in the x-direction, and $N_v=64$, 128 in the v-direction with $v_{\max}=6.5$, and $\Delta t=1/8$ for conservative, spectral and semi-Lagrangian methods which are

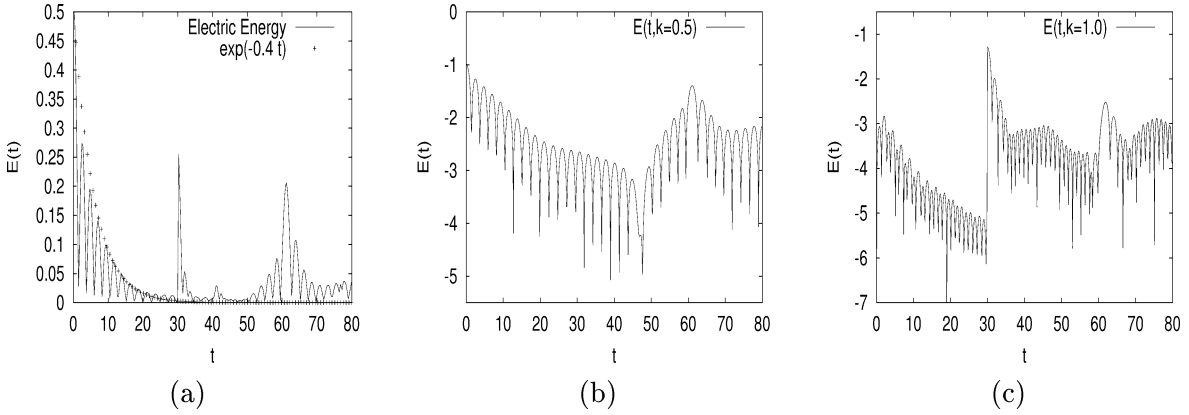

Fig. 3. Plasma echo: time development of (a) the electric energy, (b) the first mode of the electric field (in log scale), (c) the second mode of the electric field (in log scale).

Table 1
Plasma echo: relative error norm for different methods for 32×64 and 32×128 points. The approximation is compared to a reference solution computed on a fine grid (512×1024 points)

	FBM	PFC	FDM	Spectral	CIP	SL Spline	SL Hermite
32×32	0.05	0.045	0.09	0.056	0.071	0.078	0.070
32×128	0.0065	0.0036	0.035	0.0035	0.0035	0.0014	0.0034

not restricted by a CFL condition, whereas $\Delta t = 1/40$ for the Finite Difference Method. The numerical solution remains positive for all schemes and the relative error norms of variations of kinetic entropy, L^2-norm, and total energy always stay less than 10^{-3} for semi-Lagrangian and conservative schemes. The evolution of the electric field (until $t = 30\omega_p^{-1}$) obtained by different schemes is compared to a reference solution computed on a fine grid (512×1024 points) in Table 1. The conservative methods seem to be more accurate on a coarse grid, whereas the semi-Lagrangian method with a cubic spline interpolation gives the best result with 32×128 points. The order of convergence for the different methods agrees very well with the order of the reconstruction. Finally, let us mention that the approximation obtained by the Finite Difference Method strongly depends on the numerical collision frequency and the spectral algorithm on the truncation of high frequencies. It is inconvenient for the robustness of the methods, since they are very sensitive to these parameters, which are not determined by physical considerations.

6.2.2. The nonlinear Landau damping

In this case, the initial data is given by

$$f(0, x, v) = \frac{1}{\sqrt{2\pi}}\big(1 + \alpha \cos(kx)\big) \exp(-v^2/2),$$

with $\alpha = 0.5$, and $L = 2\pi/k$. We are using a number of cells $N_x = 32$ in the x-direction, and $N_v = 64$ in the v-direction with $v_{\max} = 6.5$, and $\Delta t = 1/8$ for conservative and semi-Lagrangian methods which are not restricted by a CFL condition, whereas $\Delta t = 1/40$ for the Finite Difference Method.

The linear Landau damping theory is valid as long as $t < \alpha^{-1/2}$; for longer times the problem is inherently nonlinear. Here, the Landau theory cannot be applied because nonlinear effects are too important, but this test has been studied numerically by many authors [4,10,14]. The electric energy first decays exponentially and is next periodically oscillating. In Fig. 4, the electrical energy, obtained by the different methods using 32×64 points, is plotted in logarithmic scales. It can be compared with a reference solution (512×1024) for which L^p norms and kinetic entropy are well conserved. The evolution obtained by the PFC scheme clearly appears like the best

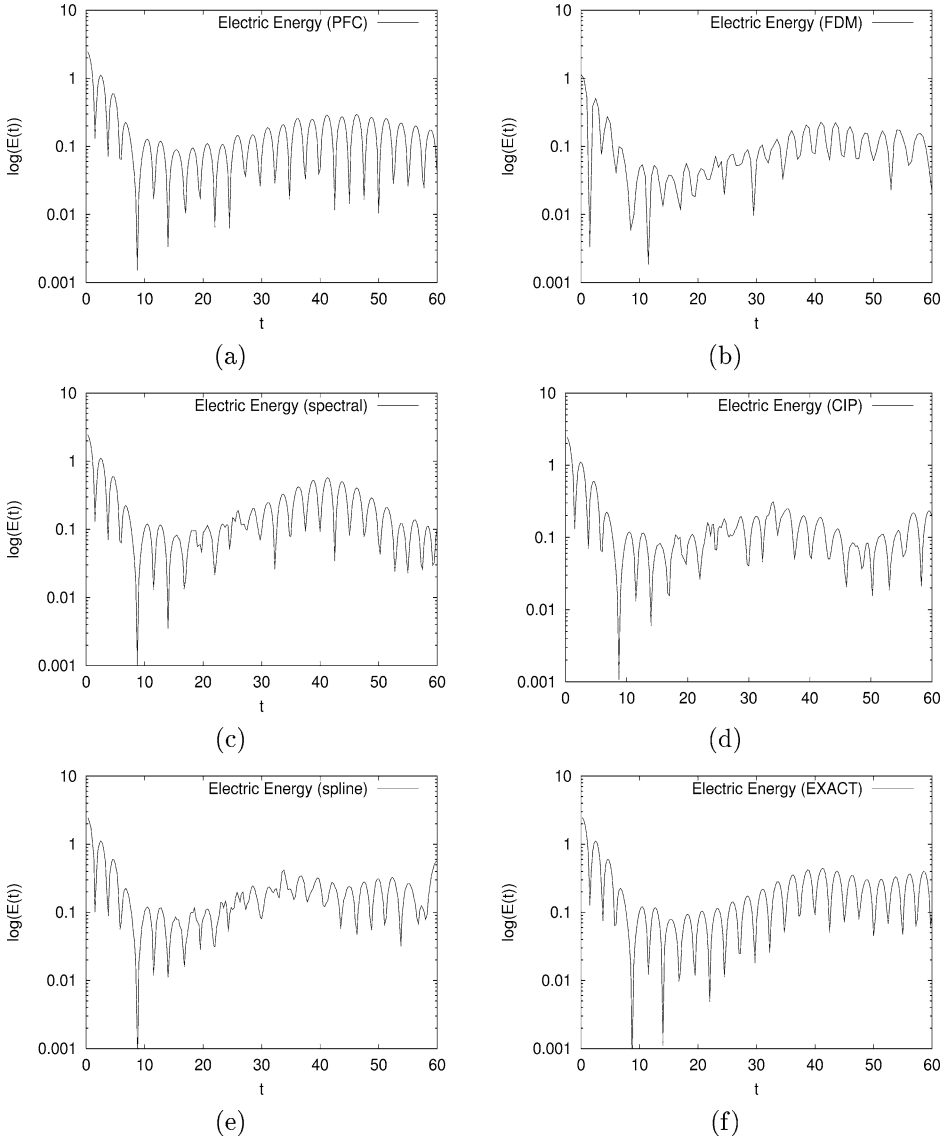

Fig. 4. Nonlinear Landau damping: time development of the electric energy using 32×64 points obtained by (a) the PFC scheme, (b) the FDM, (c) the spectral algorithm, (d) the CIP method, (e) the semi-Lagrangian method with a Cubic spline interpolation, (f) the reference solution (512×1024).

approximation. Nonlinear effects are so important that it is necessary to control spurious oscillations. The evolution of L^p norms of $f_h(t)$, $\sum |f_i(t)|^p$ for $p = 1, 2$, are reported in Fig. 5. The PFC scheme conserves the total mass and also positivity, the L^1 norm of $f_h(t)$ is then conserved along time, whereas strong spurious oscillations occur for the different semi-Lagrangian methods and for the spectral method, which do not have as efficient mechanism to eliminate numerical instabilities and rely on sampling effects for that. The use of slope correctors in the PFC scheme enhances the decay of the discrete L^2 norm, but when oscillations, due to the nonlinearity, are damped or averaged by the projection on the grid, the L^2 norm is well stabilized.

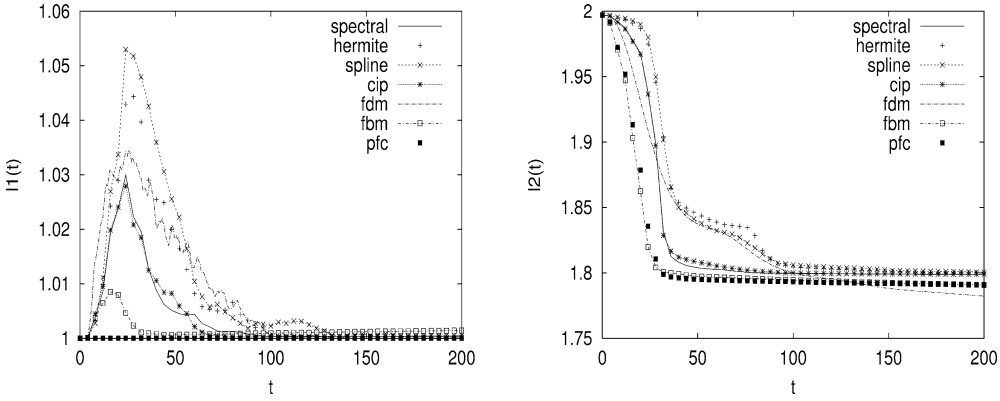

Fig. 5. Nonlinear Landau damping: time development of L^1 and L^2 norms of $f(t)$.

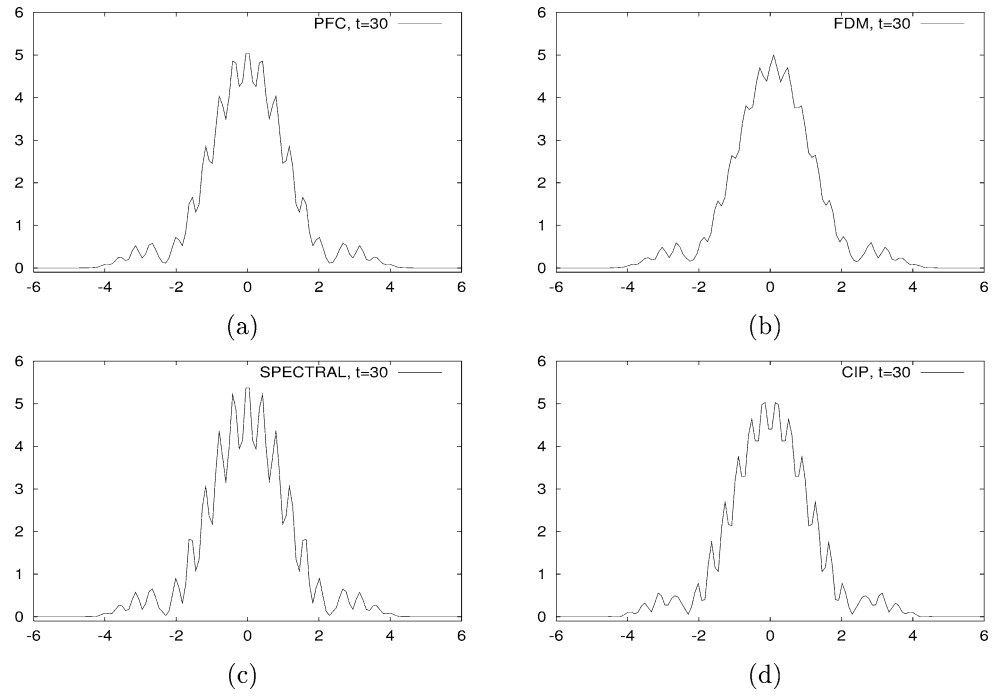

Fig. 6. Nonlinear Landau damping: the space integrated distribution function $f(v)$ at time $t = 30\omega_p^{-1}$ obtained with (a) the PFC scheme, (b) the FDM, (c) the spectral algorithm, (d) the CIP method.

For the distribution function in the (x, v) space, small bumps appear around the phase velocity $v_\phi = \omega/k$. These bumps represent particles which are trapped by electrostatic waves (see Fig. 6). As a consequence of the entropy decay, the distribution function is smoothed when filaments become smaller than the phase space grid size. Nevertheless, this smooth approximation seems to give a good description of macroscopic values (physics quantities obtained by the integration of moments of the distribution function with respect to v). Indeed, the evolution of the electric energy is more accurate than the one obtained from the semi-Lagrangian method using the cubic spline interpolation.

Table 2
Nonlinear Landau damping: total time of computation for different methods with respect to the number of points points

Numerical method	32×32 points	32×64 points	32×128 points
FBM	03.33 s	05.39 s	10.80 s
PFC	03.56 s	06.28 s	11.20 s
FDM	17.22 s	35.27 s	71.20 s
SPECTRAL	04.10 s	08.25 s	16.90 s
CIP	13.83 s	21.40 s	43.24 s
SL SPLINE	06.12 s	10.55 s	20.90 s
SL HERMITE	03.60 s	06.90 s	11.00 s

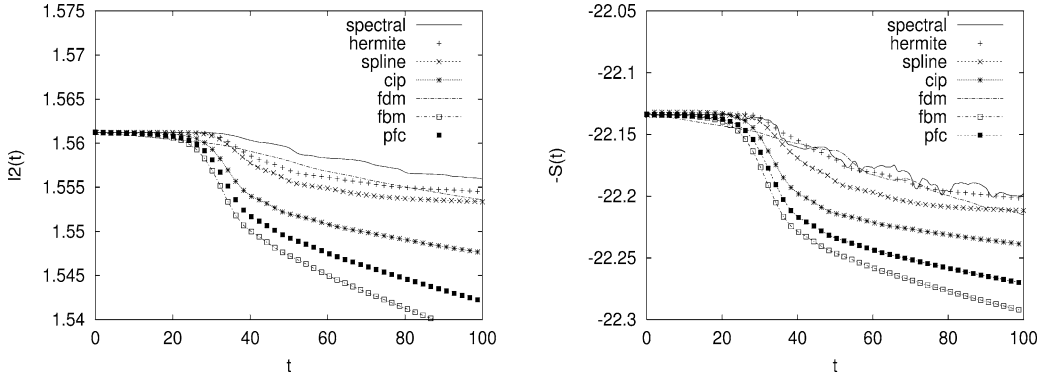

Fig. 7. Two stream instability: time development of numerical L^2 norm and entropy of $f(t)$.

Computational cost of the different methods implemented. For the test case of the nonlinear Landau Damping, we give for each method the total time of computation with respect to the number of points in Table 2. We notice that the numerical schemes using a local reconstruction are faster than ones using a global interpolation. The FDM is penalized by a CFL condition on the time step, which increases the computation time on finer grids. For the other methods, the largest time step giving the same accuracy was chosen. The use of the Fast Fourier Transforms for the spectral algorithm induces an amount of work of $N\log(N)$, where N is the number of unknowns, whereas the computational cost of Flux Conservative and semi-Lagrangian methods is linear with respect to the number of unknowns.

6.2.3. The two stream instability

We consider the symmetric two stream instability with initial condition

$$f(0,x,v) = \frac{2}{7\sqrt{2\pi}}(1+5v^2)\big(1+\alpha\big((\cos(2kx)+\cos(3kx))/1.2+\cos(kx)\big)\big)\exp(-v^2/2),$$

with $\alpha = 0.01$, $k = 0.5$, and $L = 2\pi/k$. We are using a number of cells $N_x = 64$ in the x-direction, and $N_v = 64, 128$ in the v-direction with $v_{\max} = 5$, and $\Delta t = 1/8$ for conservative, spectral and semi-Lagrangian methods which are not restricted by a CFL condition, whereas $\Delta t = 1/40$ for the Finite Difference Method. From time $t \simeq 20\omega_p^{-1}$ to $t \simeq 40\omega_p^{-1}$, the instability grows rapidly and a hole structure appears. After $t = 45\omega_p^{-1}$ until the end of the simulation, trapped particles oscillate in the electric field and the vortex rotates. Fig. 7 shows the time development of the discrete L^2 norm and of the kinetic entropy for the different methods. For this test, nonlinearities are less important than for the previous ones and the decay of the kinetic entropy is smaller. The variations of $H(t) = -\sum f_i(t)\ln(f_i(t))$ for FDM and semi-Lagrangian methods with a Hermite interpolation are

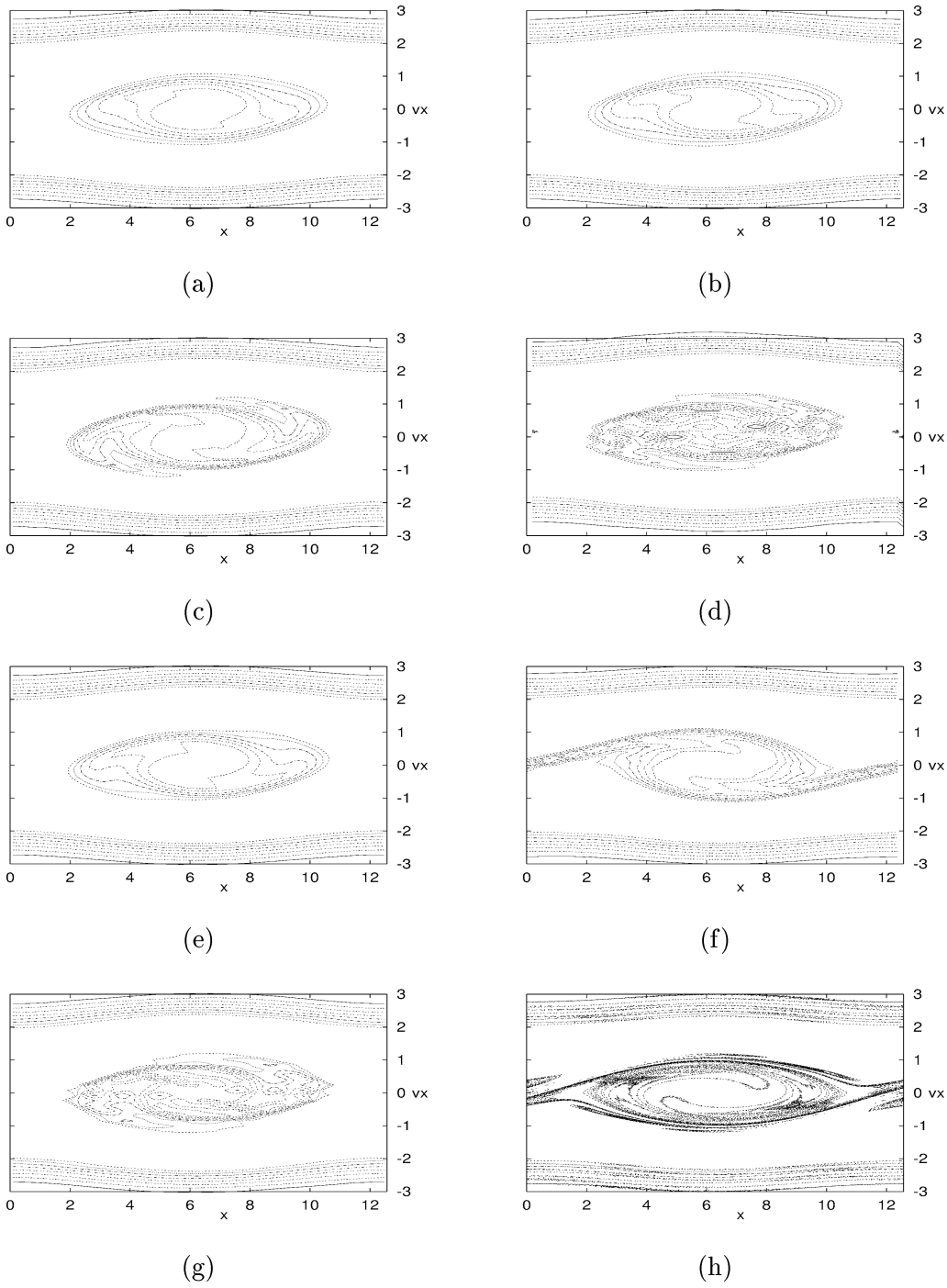

Fig. 8. Two stream instability: the x–v_x projection of the distribution function (32×64 points) at time $t = 75\omega_p^{-1}$ obtained with (a) the FBM, (b) the PFC scheme, (c) the spectral algorithm, (d) the FDM, (e) the CIP method, (f) the semi-Lagrangian method with a cubic spline reconstruction, (g) the semi-Lagrangian method with a Hermite interpolation, (h) an almost "exact" solution (512×1024 points).

less important than for the other methods, but the x–v_x projection, plotted in Fig. 8, shows that strong instabilities take place for such methods. The approximation obtained by the different schemes using 64×64 points is compared with a reference solution computed using 512×1024 points. Of course, the grid size is too coarse to detail thin filaments developed by the solution until the end of the simulation. The Flux Conservative methods seem to give a good approximation of the average of the solution on the mesh, whereas the semi-Lagrangian method with a cubic spline interpolation follows thin details of the solution for longer times. Let us mention that in their paper [14], Nakamura and Yabe have compared the CIP method with the well known PIC method. In this case, Eulerian schemes give a better approximation than particle methods.

6.3. The two-dimensional Vlasov–Poisson system

6.3.1. The nonlinear Landau damping in 2D

The initial condition is set to

$$f_0(x, y, v_x, v_y) = \frac{1}{2\pi} \exp\bigl(-(v_x^2 + v_y^2)/2\bigr)\bigl(1 + \alpha\bigl(\cos(k_x x) + \cos(k_y y)\bigr)\bigr),$$

with $\alpha = 0.5$, the velocity space is truncated at $v_{\max} = 6$, the wave numbers are $k_x = k_y = 0.5$, and the length of the periodic box in the physical space is $L_x = L_y = 4\pi$. Finally, the four-dimensional grid contains 32 points per direction and the time step is set to $\Delta t = 1/8$. From the symmetry of the initial data, the evolution of two components of the electric field are identical.

The numerical simulation of nonlinear Landau damping in the four-dimensional phase space is a difficult problem since the number of grid points is strongly limited by computer memory, and examples of simulations are

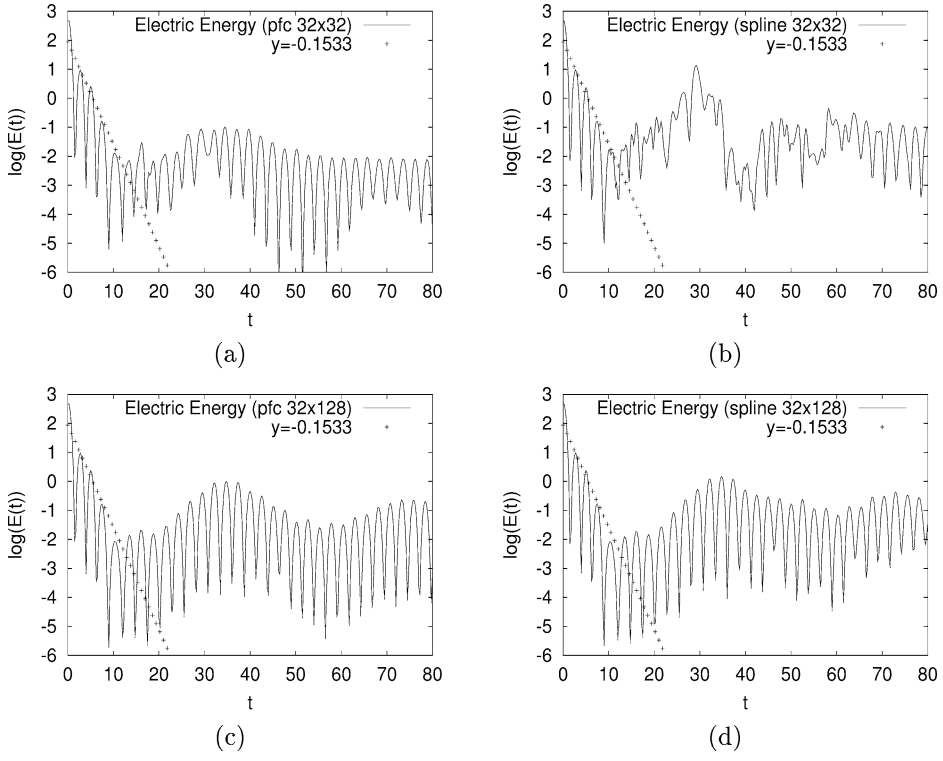

Fig. 9. Nonlinear Landau damping 2D: time development of the electric energy in logarithm scale obtained with (a) the PFC scheme (32×32), (b) the SL Spline (32×32), (c) PFC scheme (32×128), (d) the SL Spline (32×128).

Table 3
Nonlinear Landau damping 2D: total computation time for PFC and SL Spline methods with respect to the number of processors for a grid size $32 \times 32 \times 128 \times 128$ points

Number of processors	PFC scheme	SL Spline method
2 processors	9 h 03 min	12 h 42 min
4 processors	4 h 24 min	6 h 16 min
8 processors	1 h 51 min	3 h 07 min
16 processors	0 h 59 min	1 h 41 min

not frequent in the literature. Indeed, it is necessary to use a high order scheme, which kills spurious oscillations in order to obtain accurate results. Fig. 9 shows the evolution of the electrical energy obtained by the PFC scheme and the semi-Lagrangian method using cubic spline interpolation. On the one hand, the local reconstruction highly reduces the computational cost and on the other hand slope correctors avoid to introduce numerical instabilities. In Table 3, the total CPU time is presented with respect to the number of processors. The amount of work of the PFC algorithm is less important than for the cubic spline interpolation. Therefore, it will be more efficient in high dimensions.

7. Conclusions

At the beginning of numerical simulation of plasmas where computer power was such that only 1D models could be simulated, Particle In Cell methods coexisted with direct Vlasov solvers, and many such solvers can be found in the literature of the 70 s. Then in the 80 s and 90 s as people could perform 2-dimensional and sometimes even 3-dimensional simulation with PIC codes, and get useful information out of it, direct Vlasov solvers lost their interest for most people. Even though it is a fact that Monte Carlo methods and in particular PIC simulations become numerically the more interesting as the dimension increases, there is room again for Eulerian Vlasov solvers as very powerful, as well in CPU speed as in memory size, parallel computers are now available. It is easy nowadays to perform a realistic 2D kinetic plasma simulation with a direct Vlasov solver and some toy 3D simulations have also been performed. The numerics on which such methods are based are very different from those of PIC simulations. This fact alone, makes it interesting to have such a solver in one's simulation tool-box in order to benchmark once and again a PIC code against it. And for some specific problems needing high accuracy and low noise, they should be the preferred method. As we mentioned, statistical noise is absent from direct Vlasov computations, but they have other flaws which we presented and analyzed here. Among the methods we presented there is no clear winner, each method having its pros and cons. And depending on the problem being solved, one might want to use one or another method. We hope that the information given in this paper will help making this choice. It seems in particular for beam halo problems, for which we need a good precision of the distribution function in regions where its values are small, the PFC algorithm appears to be more efficient since it does not introduce any numerical oscillation and preserves positivity. However, when the distribution function remains smooth, for example for the linear Landau damping, a very high order method, like the cubic spline interpolation, can be used.

References

[1] A. Arakawa, Computational design for long-term numerical integration of the equation of fluid motion: Two-dimensional incompressible flow. Part 1, J. Comput. Phys. 1 (1) (1966) 119–143; Reprinted in J. Comput. Phys. 135 (1997) 103–114.
[2] J.P. Boris, D.L. Book, Solution of continuity equations by the method of flux-corrected transport, J. Comput. Phys. 20 (1976) 397–431.
[3] Cheng, G. Knorr, J. Comput. Phys. 22 (1976) 330–348.

[4] A.J. Klimas, A method for overcoming the velocity space filamentation problem in collisionless plasma model solutions, J. Comput. Phys. 68 (1987) 202–226.
[5] A. Klimas, W.M. Farrell, A splitting algorithm for Vlasov simulation with filamentation filtration, J. Comput. Phys. 110 (1994) 150–163.
[6] M.R. Feix, P. Bertrand, A. Ghizzo, Eulerian codes for the Vlasov equation, in: Kinetic Theory and Computing, in: Series on Advances in Mathematics for Applied Sciences, Vol. 22, 1994, p. 45.
[7] E. Fijalkow, A numerical solution to the Vlasov equation, Comput. Phys. Comm. 116 (1999) 319–328.
[8] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, J. Comput. Phys. 172 (2001) 166–187.
[9] A. Ghizzo, P. Bertrand, M. Shoucri, T.W. Johnston, E. Filjakow, M.R. Feix, A Vlasov code for the numerical simulation of stimulated Raman scattering, J. Comput. Phys. 90 (1990) 431.
[10] G. Manfredi, Long time behavior of non linear Landau damping, Phys. Rev. Lett. 79 (15) (1997) 2815–2818.
[11] R. Robert, J. Sommeria, Statistical equilibrium states for two-dimensional flows, J. Fluid. Mech. 229 (1991) 291–310.
[12] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of Vlasov equations, J. Comput. Phys. 149 (1998) 201–220.
[13] E. Sonnendrücker, J.J. Barnard, A. Friedman, D.P. Grote, S.M. Lund, Simulation of heavy ion beams with a semi-Lagrangian Vlasov solver, Nucl. Instrum. Methods in Phys. Res. A 464 (1–3) (2001) 653–661.
[14] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space, Comput. Phys. Comm. 120 (1999) 122–154.