Journal of Computational Physics 228 (2009) 6670–6681

Contents lists available at ScienceDirect

Journal of Computational Physics

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

How to improve the diagnosis of kinetic energy in δf PIC codes

S. Hess [b], F. Mottez [a,*]

[a] LUTH, Observatoire de Paris, CNRS, Université Paris Diderot, 5 Place Jules Janssen, 92190 Meudon Cedex, France
[b] Laboratory for Atmospheric and Space Physics, University of Colorado, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 16 September 2008
Received in revised form 12 May 2009
Accepted 13 May 2009
Available online 6 June 2009

Keywords:
Numerical methods
Plasma physics
Maxwell–Vlasov equations
Particle In cell
δf PIC codes

ABSTRACT

This paper propose to analyse the effect of the shape factor that is used in plasma PIC δf codes to make interpolations between the grid and the particles positions. In δf codes, the total density fluctuates, even when it should be conserved. We show that, in some cases, the computed non-physical part of the particle kinetic energy fluctuations is dependent on those of the total density. We deduce a method that can reduce drastically the statistical fluctuations in the diagnostics of the kinetic energy.

Published by Elsevier Inc.

1. Introduction

With particle codes, a plasma is considered as an assembly of *numerical* particles whose motion is dictated by the Lorentz force $qm^{-1}(\mathbf{E}_i + \mathbf{v}_i \times \mathbf{B}_i)$, where $\mathbf{E}_i$ and $\mathbf{B}_i$ represent the electromagnetic field at the position $\mathbf{x}_i$ of the particle i, and $\mathbf{v}_i$ is the particle velocity. The numerical particles are generally much less numerous than in a real plasma; this is why a careful analysis of particle codes requires to make the difference between *numerical* and *physical* particles.

There are at least two ways to compute the electromagnetic field seen by each numerical particle. The most direct method, that characterises N-body particle codes, consists of computing the electromagnetic field generated by any particle at the place of all the other particles. The computing time is proportional to N^2, where N is the number of numerical particles. These codes are too expensive when N becomes large, and methods based on tree-structured hierarchical algorithms have been designed to reduce this cost as $\sim N\log N$ [1] and more recently as $\sim N$ [2]. Tree codes are mainly used for stellar dynamics studies (the force is gravitation), and recent developments show that the concept is applicable to plasma physics [3].

The other way, that has been practiced with plasmas more extensively, is the Particle In Cell (PIC) algorithm. In PIC algorithms, the Maxwell equations are solved on a mesh, and the electromagnetic field at the position of each particle is interpolated from this mesh [4,5]. Most of the PIC codes are solved through a finite difference scheme. Finite elements PIC schemes have also been implemented; they give a better account of non-trivial geometrical constraints (such as toroidal tokamak geometries) [6].

The present paper deals with the case of finite difference particle PIC codes. With finite difference codes, the interpolation is characterised by "shape functions" S.

The interpolation of the electric field at the position of the particle i is given by

* Corresponding author. Tel.: +33 1 45077773.
E-mail addresses: sebastien.hess@obspm.fr (S. Hess), fabrice.mottez@obspm.fr (F. Mottez).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.05.035

$$\mathbf{E}(\mathbf{x}_i) = \sum_j S(\mathbf{X}_j - \mathbf{x}_i)\mathbf{E}(\mathbf{X}_j), \tag{1}$$

where $\mathbf{X}_j$ are the positions of the grid points j, and the sum is extended over the whole grid. (The interpolation of the magnetic field is similar.) Reciprocally, the resolution of the Maxwell equations requires the computation of charge and current densities. Those are deduced from the positions and velocities of the numerical particle trough the reciprocal interpolation formulas

$$\boldsymbol{\rho}(\mathbf{X}_j) = \sum_i q_i w_i S(\mathbf{X}_j - \mathbf{x}_i), \tag{2}$$

$$\mathbf{J}(\mathbf{X}_j) = \sum_i q_i \mathbf{v}_i w_i S(\mathbf{X}_j - \mathbf{x}_i). \tag{3}$$

where q_i is the electric charge. The number w_i is a statistical weight associated to each particle; it is discussed below. For any particle i, the shape function is normalised to one:

$$\sum_j S(\mathbf{X}_j - \mathbf{x}_i) = 1. \tag{4}$$

In order to avoid self-forces, the shape function used for the electromagnetic field and the shape function used for the charge and current densities are the same. The shape factor function is a uniform spline with a very small number of knots and a low degree. In 2D for instance, the shape function is usually related to four grid points (four knots) and is bi-linear (linear in each configuration space coordinate) [4].

In "classical" PIC codes, the numerical particles are markers of the whole particle distribution function. Then, the particle statistical weights are constant over time. When they are set to one ($\forall i, w_i = 1$) the value of the distribution function is proportional to the number of physical particles. Aydemir [7] has made a "Monte Carlo interpretation" of particle simulations. In that perspective, the numerical particle are called Lagrangian markers; the PIC algorithms with constant w_i's are called "simple" or "crude" Monte Carlo methods. When the w_i's are set to one, the algorithm is called "importance sampling".

As the number of numerical particles is much lower than in a real plasma, the numerical simulations are more noisy than the plasmas that they tend to modelize. But many numerical simulations are about cases where the modification of the particle distribution function is weak. Then, the total distribution function can be written $f(t,\mathbf{x},\mathbf{v}) = f_0(\mathbf{x},\mathbf{v}) + \delta f(t,\mathbf{x},\mathbf{v})$ where $f_0(\mathbf{x},\mathbf{v})$ is an equilibrium distribution function, and at any location and at any time considered in the simulation, $\delta f(t,\mathbf{x},\mathbf{v}) \ll f_0(\mathbf{x},\mathbf{v})$. In that case, it is better to use the numerical particles to modelize only the part δf of the distribution function. This is why "δf codes" (also called "perturbative codes") have been designed [8]. In the "Monte Carlo interpretation" of particle simulations, the δf algorithm is a "control variates" method [7].

In δf codes, the statistical weight w_i, deduced from the Vlasov equation must be recompiled at each time step. It represents the perturbed-to-equilibrium ratio of the distribution function. In spite of strongly increasing the signal to noise ratio, the δf approach does not allow for a totally realistic noise level. This can be seen through an analysis of the moments of the distribution function: (i) as with the nonperturbative PIC codes, because of the low number of numerical particles, the approximations of the moments of the distribution are less accurate as their order increase. In practice, with perturbative codes, it is difficult to get more than the third order. Many previous papers present methods to minimize this error, by the introduction of some additional closure equation deduced from the fluid theory [9], or by an optimized sampling of the phase space by the macro-particles [10]. (ii) As pointed out by Parker and Lee [8] the computation of the statistical weight itself introduces an error, showed by the non-conservation of the sum of the weights of all the particles. This facts causes the non-conservation of the particle number (that is conserved with nonperturbative PIC codes). Aydemir [7] has shown that the relative fluctuations of the density are then proportional to a second order moment of the statistical weight distribution (roughly $\sqrt{\overline{w_i^2}}$) divided by the square root of the numerical particle density.

But Aydemir has not taken the effect of the interpolation procedure (summarized with Eqs. (1)–(3)) in his analysis of the particle non-conservation. The interpolation makes things worse. We devote this article to an analysis of the effects of the interpolation procedure on the non-physical fluctuations of the numerical particle density and of the particle kinetic energy in δf PIC codes.

Section 2 is a short presentation of the perturbative (δf) numerical scheme. In Section 3, we show that the interpolation scheme (Eqs. (1)–(3)) is equivalent to the consideration of finite-size particles with a fixed shape. In Section 4, we analyse the non-physical fluctuations of the density (or the total particle number) induced by the finite-size particles approximation. In Section 5 we analyse the case of the non-physical kinetic energy fluctuations, and their connection with the density fluctuations. We suggest a way of improving the accuracy of the kinetic energy diagnostics. In Sections 7–9, the improved diagnostics are applied to concrete numerical simulations of the Landau damping and of the two-streams instability. Then we conclude.

2. Perturbative (δf) model

The general method is inspired from Parker and Lee [8] and its implementation for the full dynamics particle code is derived from Sydora [11]. The particles of a collisionless plasma get a distribution function f for each species which evolves in time according to the Vlasov equation:

$$\mathrm{d}_t f = \partial_t f + \mathbf{v}\partial_{\mathbf{x}} f + \gamma(\mathbf{x},\mathbf{v})\partial_{\mathbf{v}} f = 0, \tag{5}$$

where $f = f(t,\mathbf{x},\mathbf{v})$ and $\gamma(\mathbf{x},\mathbf{v}) = q/m(\mathbf{E}+\mathbf{v}\times\mathbf{B})$ is the acceleration due to the electric $\mathbf{E}$ and magnetic $\mathbf{B}$ fields. If we assume that the distribution f differs from an equilibrium distribution f_0 (corresponding to the $\mathbf{E}_0$ and $\mathbf{B}_0$ equilibrium fields, and $\mathrm{d}_t\delta f_0 = 0$) by a weak perturbation δf, the Vlasov equation can be expressed as a time-derivative of the perturbation only. For the equilibrium part:

$$\mathrm{d}_t f_0 = \partial_t f_0 + \mathbf{v}\partial_{\mathbf{x}} f_0 + \gamma_0\partial_{\mathbf{v}} f_0 = 0, \tag{6}$$

where the acceleration γ is split in a stationary acceleration due to the equilibrium fields γ_0 and an acceleration γ_1 for the perturbed fields $\mathbf{E}-\mathbf{E}_0$ and $\mathbf{B}-\mathbf{B}_0$. As the function f_0 is in equilibrium with γ_0, it does not have to be recomputed at each time step. The evolution of the perturbation is given by:

$$\mathrm{d}_t\delta f = -\gamma_1\partial_{\mathbf{v}} f_0. \tag{7}$$

The f distribution is the distribution of the particles for a given species. The distribution of the numerical particles, that we note g, is different; it is often *defined as* a Klimontovich distribution [11]

$$g(\mathbf{x},\mathbf{v}) = \sum \delta(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i), \tag{8}$$

where $\mathbf{x}_i$ and $\mathbf{v}_i$ are the position and the velocity of the ith macro-particle, and $\delta(\mathbf{v}-\mathbf{v}_i)$ is the Dirac distribution centered on $\mathbf{v}_i$. (In the Aydemir [7] Monte Carlo interpretation, g is the distribution of the Lagrangian markers.) This definition is discussed and modified in the following sections. A statistical weight w_i is defined for each particle:

$$w_i = \frac{\delta f}{g}(\mathbf{x}_i,\mathbf{v}_i). \tag{9}$$

For a Klimontovich distribution of the macro-particles (Eq. (8)) the evolution of the g distribution is a solution of the Vlasov equation (Eq. (5)), thus the f/g ratio remains constant. Then the evolution of the weight is deduced from the perturbation evolution (Eq. (7)):

$$\mathrm{d}_t w_i = -\frac{\gamma_1}{g}\partial_{\mathbf{v}} f_0 = -\left(\frac{f}{g} - w_i\right)\frac{\gamma_1}{f_0}\partial_{\mathbf{v}} f_0. \tag{10}$$

As f/g is time-invariant, it can be computed initially and stored for each particle. Among many options, two opposite choices are often made: (1) setting $f/g = 1$, and a density of macro-particles proportional to the physical particle density (2) set a waterbag distribution g for the macro-particles (uniform probability between two values v_{min} and v_{max} and null outside) and compute a specific value $(f/g)_i$ for each particle. The first option requires less storage, but we shall see in Sections 7 and 8 that the second option is much less noisy.

3. Finite-size particles

The particle distribution itself is never explicitly computed in the δf PIC algorithm; only its first moments ρ and $\mathbf{J}$ are. Even the total particle number is computed only to check the accuracy of the conservation, but is not necessary to the time step advance. We show in this section that there are two different ways of writing the distribution functions. The moments ρ and $\mathbf{J}$, used in the time iteration, associated to these two different distributions are equal. But higher order moments (used in diagnostics) can be different.

The first way of writing the distribution function is to associate it to point-like particles. In that case,

$$\delta f(t,\mathbf{x},\mathbf{v}) = \sum_i w_i\delta(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i), \tag{11}$$

where the summation is done over the particles (of index i). Then, the charge density and current densities are

$$\boldsymbol{\rho}(\mathbf{x}) = \sum_i q_i w_i\delta(\mathbf{x}-\mathbf{x}_i), \tag{12}$$

$$\mathbf{J}(\mathbf{x}) = \sum_i q_i\mathbf{v}_i w_i\delta(\mathbf{x}-\mathbf{x}_i). \tag{13}$$

and the values on the grid computed with Eqs. (2) and (3) are considered as interpolations on the simulation grid points $\mathbf{X}_j$ of the values given in Eqs. (12) and (13) at the point $\mathbf{x} = \mathbf{X}_j$.

But we can also consider, as in the Parker and Lee paper (see their Eq. (8)), that the particles are not point like, but that they have a finite size in the phase space. Then,

$$\delta f(t,\mathbf{x},\mathbf{v}) = \sum_i w_i S(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i) \tag{14}$$

or more generally

$$\delta f(t,\mathbf{x},\mathbf{v}) = \sum_i w_i \mathcal{S}(\mathbf{x}-\mathbf{x}_i, \mathbf{v}-\mathbf{v}_i), \tag{15}$$

where $\mathcal{S}$ is a function defined on the phase space. In that case, a numerical particle is considered as an object of finite size in the phase space (by opposition to a point like particle), and the particle distribution is always regular. The local density at a given point is given by the value of the $\mathcal{S}$ function. Finite-size particles are sometimes referred as particle clouds or macro-particles. Then, their contribution to the charge and current densities is naturally given by the Eqs. (2) and (3), without the need of an interpolation procedure. The Eq. (4) garantee that the charge of a macro-particle is q_i. The interpolation is in fact hidden behind the finite size of the numerical particles; and sometimes, we write that the finite-size particles are defined through the interpolation function $\mathcal{S}$.

Of course, the finite-size particle is not like a cloud of real particles, because real particles are not tied to each other by a relation specifying that their distances is phase space are constant. Studying the effect of the finite size of the particles when the distribution is given by Eq. (15) is equivalent to studying the effect of interpolation when the distribution is defined by Eq. (11).

It has been shown, since the begining of the δf PIC codes, that $\mathrm{d}_t \sum_i w_i \neq 0$ [8,7], and this fact is easily checked in practice. In the next section, we analyse the effect of the finite-size particles on the conservation of the momentums of the distribution function. We focus our analysis on the evaluation of the total density (that should remain constant) and on the particles kinetic energy (that is a part of the total energy that should remain constant).

4. Finite-size macro-particles and its consequence to the numerical solution

When the finite-size particle concept is applied, the g numerical particle distribution function is [8]:

$$g(\mathbf{x},\mathbf{v}) = \sum_i \mathcal{S}(\mathbf{x},\mathbf{x}_i,\mathbf{v},\mathbf{v}_i) = \sum_i S(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i), \tag{16}$$

where $\mathcal{S}(\mathbf{x},\mathbf{x}_i,\mathbf{v},\mathbf{v}_i) = \sum_i S(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i)$ is the interpolation function, i.e. the shape function. (Sometimes, $\mathcal{S}(\mathbf{x},\mathbf{x}_i,\mathbf{v},\mathbf{v}_i)$ is abbreviated more simply: $\mathcal{S}$.) The function $g(\mathbf{x},\mathbf{v})$ is time-varying, but $g(\mathbf{x}_i,\mathbf{v}_i) = \sum_j S(\mathbf{x}_i-\mathbf{x}_j)\delta(\mathbf{v}_i-\mathbf{v}_j) = S(0)$ is a constant, it is independent of time.

A single macro-particle defined by this interpolation function represents a part $h_i(\mathbf{x},\mathbf{v})$ of the plasma particle distribution perturbation f given by:

$$h_i(\mathbf{x},\mathbf{v}) = N_i f(\mathbf{x},\mathbf{v}) \mathcal{S}(\mathbf{x},\mathbf{x}_i,\mathbf{v},\mathbf{v}_i), \tag{17}$$

where N_i is a normalisation coefficient. In the code, the distribution function f cannot be computed directly, only h can be computed with the macro-particles algorithm. We want

$$\sum_i h_i(\mathbf{x}(t),\mathbf{v}(t)) = f(t,\mathbf{x},\mathbf{v}). \tag{18}$$

From Eq. (16), this becomes

$$\sum_i h_i(\mathbf{x},\mathbf{v}) = f(\mathbf{x},\mathbf{v}) \sum_i N_i \mathcal{S}(\mathbf{x},\mathbf{x}_i,\mathbf{v},\mathbf{v}_i), \tag{19}$$

with $N_i = g^{-1}(\mathbf{x},\mathbf{v})$. In Eq. (17), there is a discrepancy between the variables: N_i should depend only on $(\mathbf{x}_i,\mathbf{v}_i)$, not on $(\mathbf{x},\mathbf{v})$. In practice a factor

$$N_i = g^{-1}(\mathbf{x}_i,\mathbf{v}_i) = S^{-1}(0), \tag{20}$$

constant over time, is associated to each macro-particles at the beginning of the simulation. With such a value of N_i, the Eq. (18) can be satisfied only when $(\mathbf{x},\mathbf{v})$ coincide with the position of a macro-particle. (Hopefully, the support of the $\mathcal{S}$ function is restricted to a domain where $(\mathbf{x},\mathbf{v})$ is close to $(\mathbf{x}_i,\mathbf{v}_i)$.) We shall now compute the error introduced when we consider that this relations is valid for any value of $(\mathbf{x},\mathbf{v})$. Hence,

Let us consider a set of real particles whose initial distribution f_j is represented by a single macro-particle j, then $h_j(\mathbf{x}(t),\mathbf{v}(t)) = f_j(t,\mathbf{x},\mathbf{v})$. According to the Liouville theorem, $\mathrm{d}_t f_j(t,\mathbf{x},\mathbf{v}) = 0$, therefore, $\mathrm{d}_t h_j(\mathbf{x}(t),\mathbf{v}(t)) = 0$. This relation is appropriate for a set of "physical" particles: each of them follows its own trajectory, it is not bound to the others by any "non-physical relation".

However, things do not happen like this in a PIC code, because the particles represented by the same macro-particles are tied to each other. More precisely, they have the same velocity and the same acceleration. Therefore, we define a particular type of time-derivative for the macro-particle motion and for h, that we outline with a M exponent (as macro-particle), where all the particles corresponding to the same macro-particle i have the same velocity $\mathbf{v_i}$ and the same acceleration as the particle of coordinates $(\mathbf{x_i},\mathbf{v_i})$. Thus, for the macro-particle accelerations

$$\gamma^M(\mathbf{x},\mathbf{v_i}) = \gamma^M(\mathbf{x_i},\mathbf{v_i}), \tag{21}$$

when for "physical" untied particles, generally,

$$\gamma(\mathbf{x},\mathbf{v_i}) \neq \gamma(\mathbf{x_i},\mathbf{v_i}). \tag{22}$$

The relation between physical particles and particles tied in a macro-particle is given by the particle with the "reference" coordinate of the macro-particle:

$$\gamma^M(\mathbf{x_i},\mathbf{v_i}) = \gamma(\mathbf{x_i},\mathbf{v_i}) \quad \text{and} \quad \mathbf{v}_\mathbf{i}^M = \mathrm{d}_t^M \mathbf{x}_i = \mathrm{d}_t \mathbf{x}_i = \mathbf{v_i}. \tag{23}$$

It is possible to relate the physical derivative d_t to the macro-particle derivative d_t^M:

$$\mathrm{d}_t^M = \partial_t + \mathbf{v}\partial_x + \gamma^M \partial_v = (\gamma^M - \gamma)\partial_v + \mathrm{d}_t. \tag{24}$$

For particles tied in a finite-size macro-particle, the time-derivative $\mathrm{d}_t^M h$ involves the variations of $\mathcal{S}f$ (that depend on $\mathbf{x}$, $\mathbf{x}_i$, $\mathbf{v}$, and $\mathbf{v}_i$).

$$\mathrm{d}_t^M \int h_i(\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v} = N_i \int \left\{ f\mathrm{d}t^M \mathcal{S} + \mathcal{S}\mathrm{d}t^M f \right\} \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v} \tag{25}$$

It is shown in Appendix A that the variation of the sum of the weights $\mathrm{d}_t^M \sum_i w_i$ can be derived from the latter equation:

$$\mathrm{d}_t^M \sum_i w_i = \sum_i N_i \left\{ \int S(\mathbf{x}-\mathbf{x}_i)[\gamma(\mathbf{x}_i,\mathbf{v}_i) - \gamma(\mathbf{x},\mathbf{v}_i)]\partial_v f(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x} - \int S(\mathbf{x}-\mathbf{x}_i)[\gamma_0(\mathbf{x}_i,\mathbf{v}_i) - \gamma_0(\mathbf{x},\mathbf{v}_i)]\partial_v f_0(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x} \right\}. \tag{26}$$

This term represents the finite-size particle contribution to the error in the conservation of the particle number in the δf PIC simulations.

5. Kinetic energy computation

Let us now examine the error between the derivatives of the kinetic energy as computed with point particles and as in the δf PIC code. We have chosen to compute, not only the perturbation, but the whole kinetic energy K, including those associated to the equilibrium distribution function. Considering the general definition of the kinetic energy and Eq. (18)

$$K = \frac{m}{2}\int v^2 f(t,\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v} = \sum_i K_i. \tag{27}$$

With Eq. (16), we can write

$$K_i = \frac{m}{2}\int v_i^2 \frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)} S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}. \tag{28}$$

The ratio $(f/g)_i$ is constant in x and t, it is computed initially in the simulation, and not changed for subsequent times. The derivative of the kinetic energy is therefore

$$\mathrm{d}_t K_i = m\int v_i \gamma(\mathbf{x},\mathbf{v}_i)\frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)} S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}. \tag{29}$$

It is shown in Appendix B that

$$\mathrm{d}_t K_i = \mathrm{d}_t(K_1 + K_2), \tag{30}$$

where

$$K_1 = \frac{m}{2}\sum_i v_i^2 \frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)} \tag{31}$$

is a first approximation of the kinetic energy. (Here, the indices 1 and 2 refer to the decomposition of K given above, and not to specific particle numbers. As we do not specify any explicit particle number, the two kinds of indices, i or 1 and 2, can be used without confusion.) The kinetic energy K_1 is the one found with point-like particles, as shown by Hatzky et al. [10]. The correction is given by

$$\mathrm{d}_t K_2 = m\sum_i \int v_i[\gamma(\mathbf{x},\mathbf{v}_i) - \gamma(\mathbf{x}_i,\mathbf{v}_i)]\frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)} S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}. \tag{32}$$

In a few cases, this correction can be estimated. Let us consider the case of a uniform Maxwellian distribution function f_0. If $\delta f \ll f_0$, then, to the first order,

$$\frac{\partial f(\mathbf{x},\mathbf{v}_i)}{\partial \mathbf{v}} = -2\frac{\mathbf{v}_i}{v_T^2} f(\mathbf{x},\mathbf{v}_i). \tag{33}$$

Eq. (32) becomes:

$$\mathrm{d}_t K_2 = -\frac{m}{2}\sum_i N_i \int v_T^2[\gamma(\mathbf{x},\mathbf{v}_i) - \gamma(\mathbf{x}_i,\mathbf{v}_i)]\partial_v f(t,\mathbf{x},\mathbf{v}_i) S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}. \tag{34}$$

We have used the definition of N_i with the same approximation as in Eq. (20). Thus the correction on energy at a time t can be related to the error on the weight conservation given in Eq. (26),

$$K_2 = -\frac{mv_T^2}{2}\sum_i w_i - \frac{mv_T^2}{2}\int\sum_i N_i \int S(\mathbf{x}-\mathbf{x}_i)(\gamma_0(\mathbf{x}_i,\mathbf{v}_i)-\gamma_0(\mathbf{x},\mathbf{v}_i))\partial_v f_0(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}\mathrm{d}t. \tag{35}$$

The first term of the right-hand side can be evaluated directly in the numerical simulation, as w_i is a part of the data associated to every macro-particle. The second term depends only on the equilibrium configuration and the positions of the macro-particles. It can be computed. In the case of uniform initial conditions (then γ_0 does not depend on x), this term is null:

$$K_2 = -\frac{mv_T^2}{2}\sum_i w_i. \tag{36}$$

6. Three ways of computing the kinetic energy

There are actually three ways of computing the electron kinetic energy in a δf PIC code. For each of the following numerical simulations, we shall compare these three method of estimate, that are not equivalent.

The two first ways have been explained above, they consist of computing K_1 (marked with crosses on Figs. 2, 4, and 6) or $K_e = K_1 + K_2$ (stars). The third way (labelled E_{kine} and marked with triangles on the same figures) is defined by

$$E_{kine} = \frac{1}{n_0}\sum_i \frac{m}{2} v_i^2 w_i + \mathrm{C}. \tag{37}$$

The value of C is the part of the energy associated to the analytical presentation of the unperturbed distribution function f_0; it is time-invariant. (As we are only interested in the energy fluctuations, we don't compute C explicitly, we just manage to fit the initial values of E_{kine} and K_e.)

In the following section, we shall also compare the values of the total energy. It can be estimated as $E_{total} = E_{kine} + E_{elec}$ (circles), where E_{elec} is the electric energy. An another estimate is $E_T = K_e + E_{elec}$ (squares).

As we have said in Section 2, the initial conditions of the simulations can be set with a uniform value $f/g = 1$ for each particle, or with a waterbag macro-particle distribution g. The first two examples are numerical simulation of the Landau damping of a Langmuir wave. The first (Section 7) is set with a constant ratio f/g, the second (Section 8) with a waterbag distribution g. We shall see that the choice of these initial values has an influence on the quality of the kinetic energy estimates.

The third simulation is devoted to the beam-plasma instability and is set with a waterbag distribution g.

In the simulations, reduced variables are used. The unit of velocity is c, the unit of time is the inverse plasma frequency ω_{pe}^{-1}, the unit of electric charge is the electron charge absolute value e, the unit of mass is the electron mass m_e. In this system, the unit of length is $c\omega_{pe}^{-1}$ and the unit of electric field is $cm_e\omega_{pe}/e$. The unit of energy is $m_e c^2$.

7. Landau damping with uniform f/g ratio

We perform a simulation of the Landau damping with a δf PIC code. In this section, $f/g = 1$ for each particle. The physical initial distribution function is proportional to those of the macro-particles.

The electrons i are initialized with a statistic weight $w_i = 0$ The initial perturbation is given by an electric potential

$$E_1(x, t=0) = -\widehat{E}_1 \cos(kx+\phi) \tag{38}$$

and a perturbation of the macro-particle positions

$$\delta x = -E_1(x, t=0) \tag{39}$$

in order to set a density perturbation that is the corresponding solution of the Poisson equation [4]. We have not introduced any velocity perturbation. With such initial conditions, two waves of equal amplitudes are supposed to propagate in opposite directions Numerically, $\widehat{E}_1 = 0.1cm_e\omega_{pe}/e$, The electron thermal velocity is $v_{te} = 0.075c$. According to the linear theory, the frequency is $\omega = 1.275600\omega_{pe}$, the damping rate is $\gamma = -0.090220\omega_{pe}$. The perturbation has a wave vector $k\lambda_D = 0.392699$, where λ_D is the Debye length, and the phase velocity is $v_\phi = 0.243622c$. The computation is performed on a quasi-1D numerical grid of size 32×4. The initial conditions are uniform in y. Each grid cell has a size $\Delta X = 0.075c/\omega_{pe}$ and there are 2000 electrons per cell. The time step is $\Delta t\omega_{pe} = 0.05$.

The ion are considered as a neutralizing background. (This initialization has the virtue of simplicity, and has been practiced in many occasions (see for instance [11]) but it is not exact. Rigourously, we should set an initial value of the statistical weights w_i that correspond to $(f - f_0)/g$ and that is not equal to zero. Then, during the first time step of the simulation, there

is a transitory phase during which w_i get a non-null value, at the expense of the energy initially injected in the simulation. In the next section, where initial conditions are cleaner, similar waves are initialized with a smaller value of $\widehat{E}_1$.)

Fig. 1 shows the electric field energy displayed in logarithmic scale, as a function of time. The high energy oscillations correspond to the Langmuir wave frequency. These oscillations are due to the beating between the forward and backward propagating waves of similar amplitudes. The straight line superimposed to the energy corresponds to the damping predicted by the linear theory (given above). For times $t\omega_{pe} < 20$, the damping is linear. For times $t\omega_{pe} > 20$, nonlinear trapping effect, already described by O'Niel [12,13] occurs. Our goal is not to check the validity of their theories, but to analyse the energy balance. We have plot, in Fig. 2 various energies. The evaluation based on E_{kine} (triangles) do not show any correlation with the electric energy, and the fluctuations of the associated total energy E_{total} (that should be time-invariant) are larger than those of the electric energy. Therefore, using E_{kine} for the evaluation of the kinetic energy does not allow for a relevant analysis of the energy balance.

The line labelled K_e (marked with stars) is the sum of K_1 and K_2 (diamonds). It is the evaluation of the kinetic energy described in Section 5. We can see that K_2 cannot be neglected: its value is smaller but of the same order of magnitude than K_1. The finite-size particle effect is therefore important. But $K_e = K_1 + K_2$ is anti-correlated with the electric energy. The associated total energy fluctuations E_T (squares) are smaller (for any time interval) than those of the electric energy, and an energy balance analysis is possible.

The fluctuations of the total energy evaluation E_T (derived from the method described in the present paper) are three orders of magnitude smaller that those of the E_{total} evaluation. Nevertheless, the total energy E_T conservation is not perfect either. This can be due to the fluctuations already evaluated by Aydemir [7].

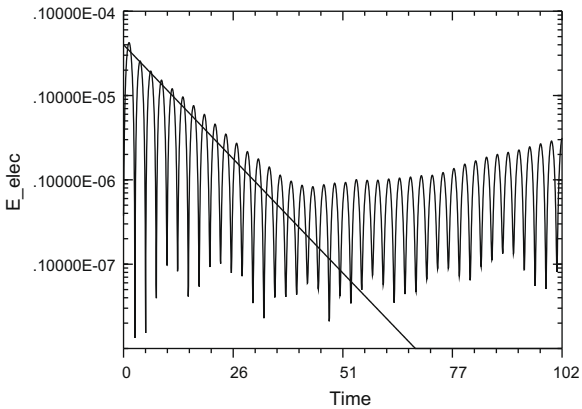

Fig. 1. Electric energy over the whole simulation box. As in the next figures, the unit of time is ω_{pe}^{-1}, the unit of energy is $m_e c^2$.

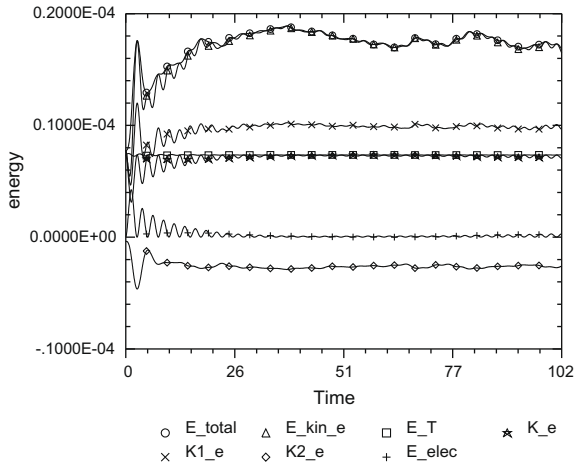

Fig. 2. Electron 0 and W kinetic energy over the whole simulation box.

8. Landau damping with waterbag macro-particle distribution

We perform a simulation of the Landau damping with a δf PIC code. The macro-particles have a waterbag velocity distribution: the probability is uniform between two boundaries $\pm 6 v_{the}$, and null outside. The initial distribution f is given through the ratio $(f/g)_i$ that is specific to each particle i. The initial conditions are similar to those of the "reference Landau simulation" developed by Belmont et al. [14], but the wave amplitude is higher (by a factor 100) and nonlinear effects occur. The wave initially set in the simulation is supposed to propagate in only one direction. The electrons i are initialized with a statistic weight w_i that corresponds to the initial perturbation of the distribution function:

$$\delta f(v,x,t=0) = \frac{N_{1,0}(x)}{\sqrt{2\pi}v_t}\exp\left\{-\frac{[v-V_{1,0}(x)]^2}{2v_t^2}\right\}. \tag{40}$$

The functions $N_{1,0}(x), V_{1,0}(x)$, are deduced from the cold plasma theory of the Langmuir wave. (see [14] for explanations and comments):

$$E_1(x,t=0) = -\widehat{E}_1\cos(kx+\phi), \tag{41}$$

$$n_1(x,t=0) = N_{1,0}(x) = -\frac{k\widehat{E}_1}{\omega_L^2}\sin(kx+\phi), \tag{42}$$

$$v_1(x,t=0) = V_{1,0}(x) = -\frac{\widehat{E}_1}{\omega_L}\sin(kx+\phi), \tag{43}$$

Numerically, $\widehat{E}_1 = 0.01 c m_e \omega_{pe}/e$, the amplitude of $N_{1,0}$ is $0.0322(\omega_{pe}/c)^3$, and the amplitude of $V_{1,0}$ is $0.0078c$. The electron thermal velocity is $v_{te} = 0.075c$. According to the linear theory, the frequency is $\omega = 1.2756\omega_{pe}$, the damping rate is $\gamma = -0.090221\omega_{pe}$. The perturbation has a wave vector $k\lambda_D = 0.3927$, where λ_D is the Debye length, and the phase velocity is $v_\phi = 0.243622c$. The computation is performed on a numerical grid of size 32×4. Each grid cell has a size $\Delta X = 0.075c/\omega_{pe}$ and there are 2000 electrons per cell. The time step is $\Delta t\omega_{pe} = 0.05$.

The ion are considered as a neutralizing background.

Fig. 3 shows the electric field energy displayed in logarithmic scale, as a function of time. The high energy oscillations correspond to the Langmuir wave frequency. These oscillations are due to the beating between the forward propagating wave and a residual backward propagating one. (See [14] for further analysis.) The straight line superimposed to the energy corresponds to the damping predicted by the linear theory (given above). As in the previous simulation, for times $t\omega_{pe} < 20$, the damping is linear; then, the nonlinear trapping effect occurs. We have plot various energies estimates in Fig. 4. As in the previous simulation, the total energy fluctuations E_{total} associated to E_{kine} are larger than those of the electric energy. Thus, the estimate E_{kine} does not allow for an analysis of the energy balance. As before, the estimate E_T of the total energy associated to $K_e = K_1 + K_2$ is much better and allow for a correct energy balance. But in the present case, where the initial conditions are more "clean" (but more difficult to design), the correction K_2 is much smaller than K_1 and E_{elec}, and a correct energy balance could be reached only with K_1, instead of K_e.

We have also made a simulation of the Landau damping, but with a much lower initial amplitude $\widehat{E}_1 = 10^{-4} c m_e \omega_{pe}/e$. The result (not displayed in a figure) shows that the correction becomes negligible and our method does not improve sensitively the diagnostics of energy. Other effects than those induced by finite size numerical particles, such as the finite particle number [7] maybe dominant.

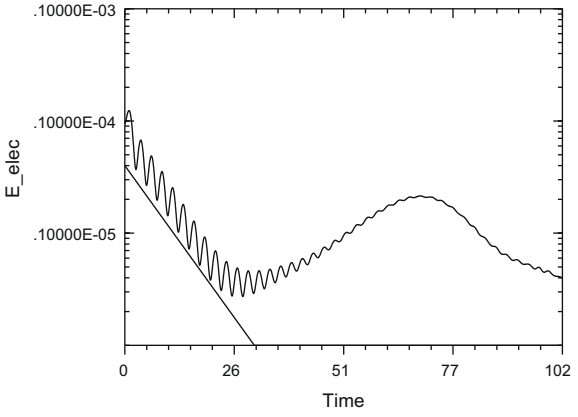

Fig. 3. The electric energy E_{elec} as a function of time, displayed in logaritmic scale. The straight line corresponds to the damping rate $\gamma = -0.090221\omega_{pe}$ given by the linear theory of the Langmuir wave.

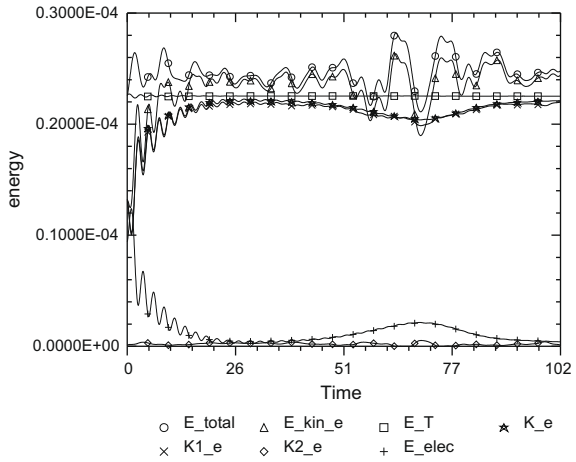

Fig. 4. Various energies, integrated in the whole simulation domain, as a function of time, displayed in linear scale. The energy labelled E_{elec} (marked with pluses +) is the electric energy displayed in Fig. 3. The energy labelled E_{kin_e} (marked with triangles △) is the electron kinetic energy computed according to the classical method. The energy labelled E_{total} (marked with circles ○) is the sum of the electric and kinetic E_{kin_e} energies. The energy labelled K_e (marked with stars ☆) is the kinetic energy evaluated according to the method described in Section 5; it is the sum of K_1 (crosses ×) and K_2 (diamonds ◇). The corresponding total energy is labelled E_T (marked with squares □) and is much better conserved than E_{total}.

9. Bump in tail instability

This section is based on a simulation of the electron beam-plasma instability.

The initial electron distribution function consists of the addition of a beam of electrons with a speed v_b to an uniform Maxwellian electrons distribution:

$$f = \exp\left(\frac{-\mathbf{v}^2}{2v_t^2}\right) + \epsilon \exp\left[\frac{-(\mathbf{v}-\mathbf{v}_b)^2}{2v_{tb}^2}\right]. \tag{44}$$

The equilibrium function f_0 is set to $f_0 = \exp\left(\frac{-\mathbf{v}^2}{2v_t^2}\right)$ and the rest of the initial distribution is carried by the numerical particles.

The electrons i are initialized with a statistic weight w_i that corresponds to an initial perturbation of the distribution function that is a core Maxwellian function of relative amplitude 1, and a beam (of gaussian profile) of thermal velocity $v_{tb} = 0.0075c$, of mean velocity $v_{mb} = 0.1875c$, and of relative amplitude $\epsilon = 0.5$. The core electron thermal velocity is $v_{te} = 0.075c$. The computation is performed on a numerical grid of size 32×4. Each grid cell has a size $\Delta X = 0.075c/\omega_{pe}$ and there are 500 electrons per cell. The time step is $\Delta t\omega_{pe} = 0.05$.

The ion are considered as a neutralizing background. We can see on Fig. 5 that the energy has a phase of linear growth ($t\omega_{pe} < 30$) then a phase of saturation and oscillations. Without entering into the details of this already well documented case, we can see that this behavior is qualitatively conform to the classical results concerning the beam-plasma instability. Fig. 6 shows the kinetic E_{kin_e} and total E_{total} energies computed following the "classical method" and the kinetic K_e and total E_T energies with the modified method. We are only interested on their variations. Ideally, both E_{total} and E_T should be constant.

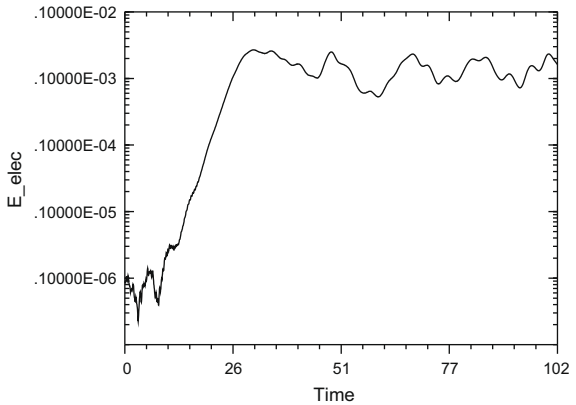

Fig. 5. The electric energy E_{elec} as a function of time, displayed in logaritmic scale.

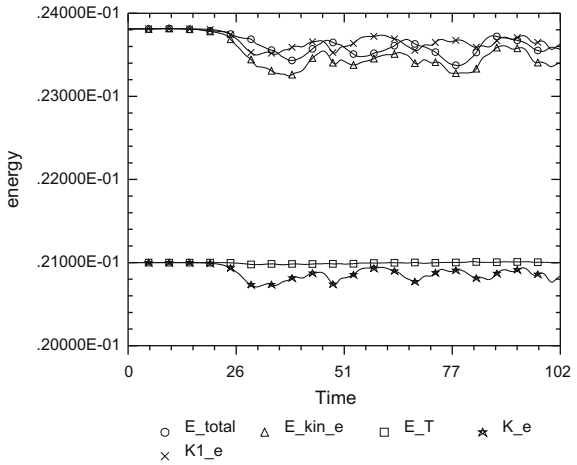

Fig. 6. A few energies, as in Fig. 4 but for the bump in tail instability simulation. The energy labelled E_{kin_e} (marked with triangles △) is the electron kinetic energy computed according to the classical method. The energy labelled E_{total} (marked with circles ○) is the sum of the electric and kinetic E_{kin_e} energies. The energy labelled K_e (marked with stars ☆) is the kinetic energy evaluated according to the method described in Section 5. The corresponding total energy is labelled E_T (marked with squares □) and is much better conserved than E_{total}.

We can see in practice that the fluctuations of E_T are much lower than those of E_{total}, especially in the nonlinear phase when the exchanges between the kinetic and electric energy become significant.

It is important to notice that in spite of a relatively important beam, our correction of energy is based on the Eq. (33) where only f_0 (that represents only the core distribution) has been taken into account. In spite of this approximation, the correction of kinetic energy is satisfactory. We can also notice that our correction of energy is significant only when the amplitude of the signal (at the end of the linear growth phase) is high enough. For a lower signal, other causes of error (such as those studied by Aydemir [7]) are dominant.

10. Conclusion

Generally, with δf codes, the evaluation of the perturbations of the moments of the particle distribution function are based on formulas of the type

$$\langle \delta A \rangle = \frac{1}{n_0} \sum_i A w_i. \tag{45}$$

We have shown that the finite size of the macro-particles induces some non-physical effects that limit the accuracy of such diagnostics. For instance, the density, i.e. the moment of order 0, that would be $\sum_i w_i/n_0$ is not conserved. We have shown an analytical expression (Eq. (26)) of the error caused by the finite particle size effect (or more precisely, we can express the time-derivative of this error). Then, we have evaluated the error introduced by the particle finite size effect into the evaluation of the kinetic energy; this is given in the Eq. (32). We have not based our energy computation on a formula like Eq. (45) that involves only δf, but on an estimate of the total kinetic energy, Eq. (31) that involves the ratio f/g associated to each particle. In general, the formula (31) does not provide a better accuracy than with Eq. (45). But in the particular case of a Maxwellian equilibrium function f_0, there is a simple and practical way of relating the error K_2 on the kinetic energy to the error on the particle number $\sum_i w_i$, (Eq. (36)). Using both Eqs. (31) and (36) gives a much more accurate energy estimate than with Eq. (45).

In order to test this way of improving the diagnostics of energy, we have made simulations of Landau damping and beam-plasma instability where the three ways of computing the kinetic energy are compared: those with Eqs. (45) and (31) and the correction Eq. (36). We have based our tests of accuracy on the conservation of the total energy. We have shown that the first methods is the worst. The second method can give good results when the initial condition are set carefully. The third method is the best, the numerical fluctuations of the total energy are three order of magnitude lower than with the first method. In this last case, the level of fluctuations of the total energy is well below the level of fluctuations of its electric and kinetic components; this fact is not so common in the world of PIC numerical simulations.

Nevertheless, there are other causes of error on the diagnostics of energy, such as the statistical fluctuations induced by the low number of numerical particles. The numerical particle finite size effect, and our proposed solution to counteract it, have shown in our simulations to be more effective when the amplitude of the physical signal is higher. This fact can be explained: the fluctuations of density scale as $\sqrt{\bar{w}_i^2}/N$ when caused by the finite number of numerical particles [7]. They do not depend on the amplitude of the forces acting on the particles. In the case of the finite size effect, the fluctuations of the den-

sity are proportional to the heterogeneity of the particle acceleration $\gamma(\mathbf{x},\mathbf{v}_i)-\gamma(\mathbf{x}_i,\mathbf{v}_i)$. In the case of a stronger signal (for instance waves with the same wavelength but with larger amplitudes), this term, mainly associated to the wave electric field, is higher and contributes more effectively to the numerical fluctuations of the density. Therefore, in the case of stronger forces acting physically on the particles, the finite size effect and our method to counteract it, are the most significant terms affecting the precisions of the diagnostics.

It is important to mention an important restriction: this method requires a Maxwellian equilibrium distribution function f_0. In the case of counterstreaming electron beams [15,16], for instance, the distribution f_0 is composed of two shifted Maxwellian distributions (with opposite mean velocities) of similar densities; in that case; Eq. (33) is not valid anymore and our improved diagnostic cannot be applied as is. In the case of the beam-plasma instability shown in Section 9, our correction worked because the beam density is much lower than the core density, and we could neglect it in Eq. (33), keeping in it only the core Maxwellian distribution. It is certainly possible to find other useful cases where a similar method can be applied: the fundamental need is to provide a relation between $\partial_v f_0(\mathbf{x_i},\mathbf{v_i})$ and $v_i f_0(\mathbf{x_i},\mathbf{v_i})$ that is (almost) independent of v_i. This is possible for instance when the particle distribution function is a sum of Maxwellian distributions of different temperatures (in this case, each "family" of electrons can be treated separately) or in the case of a Maxwellian with an anisotropy of temperature.

The present analysis of the finite macro-particle size errors might have other applications in δf PIC code algorithms. For instance, it might help to design a new method of numerical evaluation of the pressure tensor components.

As the expression of the force is not explicitly required in our computations, the present analysis of finite size effects is also valid in the case of finite difference PIC δf simulations of gravitational systems.

Appendix A. Error on density conservation

Considering that $\mathbf{v}^M=\mathrm{d}_t^M\mathbf{x}$, $\mathbf{v}_i^M=\mathrm{d}_t^M\mathbf{x}_i=\mathbf{v}_i$, $\gamma^M=\mathrm{d}_t^M\mathbf{v}$, $\gamma_i^M=\mathrm{d}_t^M\mathbf{v}_i$, considering the peculiar form of $\mathcal{S}$ given in Eq. (16), noting $S'=\partial_x S(x-x_i)$, we have $\partial_x\mathcal{S}=-\partial_{x_i}\mathcal{S}=S'\delta(v-v_i)$, $\partial_v\mathcal{S}=-\partial_{v_i}\mathcal{S}=S\delta'(v-v_i)$, and the first term on the right-hand side of Eq. (25), can be written

$$\begin{aligned}\int f\mathrm{d}_t^M\mathcal{S}\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}&=\int \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}f(\mathbf{x},\mathbf{v})\times\left[\mathbf{v}^M\partial_x+\mathbf{v}_i^M\partial_{x_i}+\gamma^M(\mathbf{x},\mathbf{v})\partial_v+\gamma^M(\mathbf{x}_i,\mathbf{v}_i)\partial_{v_i}\right]S(\mathbf{x}-\mathbf{x}_i)\delta\left(\mathbf{v}^M-\mathbf{v}_i^M\right)\\&=\int S(\mathbf{x}-\mathbf{x}_i)\left[\gamma^M(\mathbf{x},\mathbf{v}_i)-\gamma^M(\mathbf{x}_i,\mathbf{v}_i)\right]\partial_v f(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}=0.\end{aligned}\tag{A.1}$$

The second term of Eq. (25) is

$$\int\mathcal{S}\mathrm{d}_t^M f\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}=\int S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}_t^M f(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}=\int S(\mathbf{x}-\mathbf{x}_i)\left\{\mathrm{d}_t f(t,\mathbf{x},\mathbf{v}_i)+\left[\gamma^M(\mathbf{x},\mathbf{v}_i)-\gamma(\mathbf{x},\mathbf{v}_i)\right]\partial_v f(t,\mathbf{x},\mathbf{v}_i)\right\}\mathrm{d}\mathbf{x}.\tag{A.2}$$

The two above terms are sumed up in Eq. (25). The Vlasov–Liouville theorem implies that $\mathrm{d}_t f(\mathbf{x}_i,\mathbf{v}_i)=0$ (this is a "physical" derivative), therefore,

$$\mathrm{d}_t^M\int h_i(\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}=N_i\int S(\mathbf{x}-\mathbf{x}_i)[\gamma(\mathbf{x}_i,\mathbf{v}_i)-\gamma(\mathbf{x},\mathbf{v}_i)]\partial_v f(t,\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}.\tag{A.3}$$

Coming back to the definition in Eq. (17),

$$\sum_i\mathrm{d}_t^M\int h_i(\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}=\sum_i N_i\left[\mathrm{d}_t^M\int f_0\mathcal{S}\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}+\mathrm{d}_t^M\int\delta f\mathcal{S}\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}\right].\tag{A.4}$$

The first term on the right-hand side of Eq. (A.4) can be treated the same way as in Eqs. (A.1) and (A.2) replacing f by f_0, therefore

$$\sum_i N_i\mathrm{d}_t^M\int f_0\mathcal{S}\mathrm{d}\mathbf{x}=\int S(\mathbf{x}-\mathbf{x}_i)[\gamma_0(\mathbf{x}_i,\mathbf{v}_i)-\gamma_0(\mathbf{x},\mathbf{v}_i)]\partial_v f_0(\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}.\tag{A.5}$$

We can develop the second term on the right-hand side of Eq. (A.4). Considering the Eqs. (9), (20), and (4),

$$\sum_i N_i\mathrm{d}_t^M\int\delta f\mathcal{S}\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}=\mathrm{d}_t^M\sum_i w_i+\int N_i[\delta f(t,\mathbf{x},\mathbf{v}_i)-\delta f(t,\mathbf{x}_i,\mathbf{v}_i)]S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}.\tag{A.6}$$

As the domain where S is not null is small (usually a grid cell), usually, $[\delta f(t,\mathbf{x},\mathbf{v}_i)-\delta f(t,\mathbf{x}_i,\mathbf{v}_i)]\ll\delta f(t,\mathbf{x}_i,\mathbf{v}_i)$, and to the first order, the second term in the right-hand side can be neglected. Then, Eq. (A.4) reduces to

$$\sum_i\mathrm{d}_t^M\int h_i(\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}=\mathrm{d}_t^M\sum_i w_i+\int S(\mathbf{x}-\mathbf{x}_i)[\gamma_0(\mathbf{x}_i,\mathbf{v}_i)-\gamma_0(\mathbf{x},\mathbf{v}_i)]\partial_v f_0(\mathbf{x},\mathbf{v}_i)\mathrm{d}\mathbf{x}.\tag{A.7}$$

Eq. (26) is found when h_i is eliminated from Eqs. (A.3) and (A.7).

Appendix B. Correction on the kinetic energy

As we have seen, $\gamma(\mathbf{x},\mathbf{v}_i)$ is not computed with a δf PIC code. Therefore, we decompose the above term as follows

$$\mathrm{d}_t K_i = \frac{m}{2}\int v_i\gamma(\mathbf{x}_i,\mathbf{v}_i)\frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)}S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}+\frac{m}{2}\int v_i[\gamma(\mathbf{x},\mathbf{v}_i)-\gamma(\mathbf{x}_i,\mathbf{v}_i)]\frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)}S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}. \tag{B.1}$$

The first term on the right-hand side is simply $v_i\gamma(\mathbf{x}_i,\mathbf{v}_i)\frac{f(t,\mathbf{x},\mathbf{v}_i)}{g(\mathbf{x},\mathbf{v}_i)}$, because $\int S(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x}=1$. The decomposition $K=K_1+K_2$ given in Eqs. (31) and (32) can be deduced from the above relation.

References

[1] J. Barnes, P. Hut, A hierarchical O(*N*log*N*) force-calculation algorithm, Nature 324 (1986) 446–449.
[2] W. Dehnen, A hierarchical O(*N*log*N*) force calculation algorithm, Journal of Computational Physics 179 (2002) 27–42.
[3] A. Beck, F. Pantellini, Spherical expansion of a collisionless plasma into vacuum: self-similar solution and ab initio simulations, Plasma Physics and Controlled Fusion 51 (1) (2009) 015004.
[4] C. Birdsall, A. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, 1985.
[5] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Hilger, Bristol, 1988.
[6] M. Fivaz, S. Brunner, G. de Ridder, O. Sauter, T.M. Tran, J. Vaclavik, L. Villard, K. Appert, Finite element approach to global gyrokinetic particle-in-cell simulations using magnetic coordinates, Computer Physics Communications 111 (1998) 27–47.
[7] A.Y. Aydemir, A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas, Physics of Plasmas 1 (1994) 822–831.
[8] S.E. Parker, W.W. Lee, A fully nonlinear characteristic method for gyrokinetic simulation, Physics of Fluids B 5 (1993) 77–86.
[9] Y. Todo, A complementary fluid method in δf particle simulation, Journal of Plasma and Fusion Research 81 (2007) 944–948.
[10] R. Hatzky, T.M. Tran, A. Könies, R. Kleiber, S.J. Allfrey, Energy conservation in a nonlinear gyrokinetic particle-in-cell code for ion-temperature-gradient-driven modes in θ-pinch geometry, Physics of Plasmas 9 (2002) 898–912.
[11] R.D. Sydora, Low noise electromagnetic δf particle-in-cell plasma simulation model and applications, in: J. Büchner, C.T. Dum, M. Scholer (Eds.), Space Plasma Simulation: Proceedings of the Sixth International School/Symposium, 2001, p. 5.
[12] T. O'Neil, Collisionless damping of nonlinear plasma oscillations, Physics of Fluids 8 (1965) 2255–2262.
[13] G. Laval, R. Pellat, A. Roux, Amortissement non lineaire d'une onde monochromatique d'amplitude finie, Physics Letters A 29 (1969) 159–160.
[14] G. Belmont, F. Mottez, T. Chust, S. Hess, Existence of non-Landau solutions for Langmuir waves, Physics of Plasmas 15 (5) (2008) 052310-1–052310-13.
[15] F. Mottez, S. Perraut, A. Roux, P. Louarn, Coherent structures in the magnetotail triggered by counterstreaming electron beams, Journal of Geophysical Research 102 (1997) 11399–11408.
[16] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space, Computer Physics Communications 120 (1999) 122–154.