Journal of Computational Physics 229 (2010) 1927–1953

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Conservative semi-Lagrangian schemes for Vlasov equations

Nicolas Crouseilles [a,*], Michel Mehrenberger [b], Eric Sonnendrücker [b]

[a] INRIA-Nancy-Grand Est, CALVI Project and IRMA, Université de Strasbourg and CNRS, 7 Rue René Descartes, 67084 Strasbourg Cedex, France
[b] IRMA, Université de Strasbourg and CNRS and INRIA-Nancy-Grand Est, CALVI Project, 7 Rue René Descartes, 67084 Strasbourg Cedex, France

ARTICLE INFO

Article history:
Received 23 February 2009
Received in revised form 30 October 2009
Accepted 9 November 2009
Available online 26 November 2009

Keywords:
Vlasov equation
Semi-Lagrangian method
Conservative
Numerical methods

ABSTRACT

Conservative methods for the numerical solution of the Vlasov equation are developed in the context of the one-dimensional splitting. In the case of constant advection, these methods and the traditional semi-Lagrangian ones are proven to be equivalent, but the conservative methods offer the possibility to add adequate filters in order to ensure the positivity. In the non-constant advection case, they present an alternative to the traditional semi-Lagrangian schemes which can suffer from bad mass conservation, in this time splitting setting.

1. Introduction

To describe the dynamics of charged particles in a plasma or in a propagating beam, the Vlasov equation can be used to calculate the plasma response to the electromagnetic fields. The unknown $f(t,x,v)$ which depends on the time t, the space x and the velocity v represents the distribution function of the studied particles. The coupling with the self-consistent electromagnetic fields is taken into account through the Maxwell or Poisson equation.

Due to its nonlinear structure, analytical solutions are available only in few academic cases, and numerical simulations have to be performed to study realistic physical phenomena. Nowadays, mostly two classes of methods are used to investigate the behaviour of the numerical solution to the Vlasov equation. On the one hand, particle in cell (PIC) methods, which are the most widely used, approach the plasma by macro-particles, the trajectories of which follow the characteristic curves of the Vlasov equation whereas the electromagnetic fields are computed by gathering the charge and current densities particles on a grid of the physical space (see [2]). On the other hand, Eulerian methods consist in discretizing the Vlasov equation on a grid of the phase space using classical numerical schemes such as finite volumes or finite elements methods for example (see [5,11,26]).

Although PIC methods can theoretically and potentially resolve the whole 6 dimensional problem, it is well known that the inherent numerical noise makes difficult a precise description of low density regions, despite significant recent improvements. Hence, Eulerian methods offer a good alternative to overcome this lack of precision, even if problems of memory can arise when one deals with high dimensions. In particular, Vlasov codes seem to be appropriate to study nonlinear processes.

This last decade, gridded Vlasov solvers have been developed for $2D, 4D$ and even $5D$ phase space problems. Among them, the semi-Lagrangian method using a cubic spline interpolation (SPL) [26] and the positive flux conservative (PFC) method [11] have been implemented to deal with physical applications [14,13,29].

[*] Corresponding author.
E-mail addresses: crouseil@math.u-strasbg.fr (N. Crouseilles), mehrenbe@math.u-strasbg.fr (M. Mehrenberger), sonnen@math.u-strasbg.fr (E. Sonnendrücker).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.11.007

Recently, a parabolic spline method (PSM) has been introduced for transport equations arising in meteorology applications [32,33]. This method benefits from the best approximation property of the SPL method and from the conservation of mass and positivity (by applying a suitable filter) of the PFC method.

The aim of the present work is to study such a conservative method in the context of the Vlasov equation. Conservative methods present a lot of advantages. In addition to the inherent conservative property, slope limiters can be introduced in the reconstruction to ensure some specifical properties (positivity, monotonicity); moreover, since they solve the conservative form of the equation, multi-dimensional problems can be solved by a splitting procedure so that the solution of the full problem is reduced to a succession of solution to only one-dimensional problems. Obviously, this property is of great interest from an implementation and algorithmic point of view.

We will focus here essentially on PSM, which has never been applied to our knowledge to Vlasov simulations. This method is compared to other reconstructions like those used in PFC or PPM approaches. We will also introduce a new method based on a cubic splines approximation of the unknown; the characteristics curves are followed forwardly as in [27,9], but the unknown is reconstructed in a conservative way using its values on the transported non-uniform mesh.

In our numerical experiments, we first consider the special case of directional splitting with constant advection (like the Vlasov–Poisson system). In that case, when no filter is applied, we prove that the advective scheme (e.g. SPL) and the conservative one (e.g. PSM) are *equivalent*. Note that in this setting, a mathematical proof of the convergence has been performed in [1]. We also propose a unified reconstruction which enables to recover different methods available in the literature. These approaches can be coupled with filters which can be applied to preserve the positivity or monotonicity. Different filters are discussed and compared: non-oscillatoring filter or maxima preserving filters. Various numerical results are given to emphasize the differences of the methods and of the filters for transport problems and for classical plasma test cases: the strong Landau damping, the bump-on-tail instability and the two stream instability.

We then focus on the case where we do not have a constant advection, as is the case for the guiding-center model. In [26], a $2D$ interpolation was proposed to approximate this model to overcome the poor density conservation of the advective splitting procedure (see [17]). The time splitting in the conservative form has been successfully tested for the PFC scheme [3], which appears to be too diffusive. Different works have also been devoted to the study of the constrained interpolation profile (CIP) method in its conservative form (see [21,22,28] and references therein). The time splitting in the advective form is often discarded since it can lead to bad mass conservation, especially for long time simulations [17,21]. We see here that with the conservative spline formulation (PSM), we can perform simulations with directional splitting not as diffusive as PFC, while maintaining the mass conservation. The time step is nevertheless limited by a condition which imposes that the characteristic curves are enough accurately computed. In particular, they should not cross. In the constant advection case, this never occurs so that there is no condition on the time step; in non-constant advection case, the condition is generally less restrictive than the CFL standard condition (see [25]), and this has been checked in our simulations.

The paper is organized as follows: first, semi-Lagrangian conservative methods are recalled and also introduced for one-dimensional general problems. Then, the constant advection case is investigated, and it is proved that, in this case, a conservative method and its advective counterparts are equivalent when no filter are considered; numerical results applied to the Vlasov–Poisson model are then discussed. Finally, we focus on the more interesting non-constant advection case for which numerical results illustrate the good behaviour of the new approaches.

2. Conservative methods

We are interested in the approximation of multi-dimensional transport equations of the form

$$\frac{\partial g}{\partial t} + \nabla_x \cdot (ag) = 0, \quad x \in \Omega \subset \mathbb{R}^n, \tag{2.1}$$

where the unknown g depends on time and on the multi-dimensional spatial direction x and a is a divergence free vector field $\nabla_x \cdot a = 0$ which can depend on time. The so-called conservative form (2.1) is then equivalent to the advective form

$$\frac{\partial g}{\partial t} + a \cdot \nabla_x g = 0, \quad x \in \Omega \subset \mathbb{R}^n. \tag{2.2}$$

In Vlasov-type equations which enter in the class of equation of the form (2.1), Ω is a subset of the phase space which has up to 6 dimensions.

Splitting the components of x into x_1 and x_2, Eq. (2.1) can be written in the form

$$\frac{\partial g}{\partial t} + \nabla_{x_1} \cdot (a_1 g) + \nabla_{x_2} \cdot (a_2 g) = 0,$$

where a_1 and a_2 denote the component of the field a corresponding to x_1 and x_2. It is well known (see [5]) that a splitting procedure involves a successive solution of

$$\frac{\partial g}{\partial t} + \nabla_{x_1} \cdot (a_1 g) = 0, \quad \frac{\partial g}{\partial t} + \nabla_{x_2} \cdot (a_2 g) = 0, \tag{2.3}$$

keeping high order accuracy in time for the whole Eq. (2.1). However, the traditional semi-Lagrangian methods described in [26] for example do not resolve the conservative form but the non-conservative form of the equations (2.2). Then, by solving only the advective form of (2.3), the corrective terms $g\nabla_{x_1}\cdot a_1$ and $g\nabla_{x_2}\cdot a_2$ are omitted and can lead to an important lack of accuracy in long time simulations (see [17,21]). An alternative way would be to solve the conservative form so that the solution of (2.1) can be performed by solving a succession of one-dimensional problems. Hence in the sequel, we propose different conservative methods to solve one-dimensional problems; the methods are first presented in a general context but practical examples will be detailed in the next sections.

2.1. Conservative backward semi-Lagrangian methods for one-dimensional problems

The conservative methods enable to solve the conservative terms separately so that we restrict ourselves to the following one-dimensional problem

$$\frac{\partial g}{\partial t}+\frac{\partial(ag)}{\partial x}=0,\quad x\in I\subset\mathbb{R}, \tag{2.4}$$

For $N\in\mathbb{N}^*$, we define the grid points

$$x_i=x_{\min}+i\Delta x,\quad i\in\frac{1}{2}\mathbb{Z},\ \text{with } \Delta x=(x_{\max}-x_{\min})/N \text{ and } I=[x_{\min},x_{\max}].$$

We consider the average quantity for a given time s

$$\bar{g}_i(s)=\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}}g(s,x)dx,\ i=0,\dots,N-1. \tag{2.5}$$

Now for another time t, thanks to the conservation of the volume, we can write the following equality

$$\int_{x_{i-1/2}}^{x_{i+1/2}}g(t,x)dx=\int_{x_{i-1/2}(s)}^{x_{i+1/2}(s)}g(s,x)dx, \tag{2.6}$$

where $x_{i-1/2}$ and $x_{i-1/2}(s)$ belong to the same characteristic curve defined by

$$\frac{dX(\tau)}{d\tau}=a(\tau,X(\tau)),\quad X(t)=x_{i-1/2},\quad X(s)=x_{i-1/2}(s),\quad i=0,\dots,N. \tag{2.7}$$

Assuming that the values $\bar{g}_i(s),\ i=0,\dots,N-1$ are known, we can reconstruct the primitive function $G(s,x)=\frac{1}{\Delta x}\int_{x_{-1/2}}^{x}g(s,y)dy$ on the grid points as a cumulative function

$$G(s,x_{i-1/2})=\sum_{k=0}^{i-1}\bar{g}_k(s),\quad i=1,\dots,N,\quad G(s,x_{-1/2})=0. \tag{2.8}$$

Using (2.6), we then have

$$\bar{g}_i(t)=\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}}g(t,x)dx=\frac{1}{\Delta x}\int_{x_{i-1/2}(s)}^{x_{i+1/2}(s)}g(s,x)dx=G(s,x_{i+1/2}(s))-G(s,x_{i-1/2}(s)). \tag{2.9}$$

Thanks to this equality, for going from time s to time t, we need to

- compute at least numerically the values $x_{i-1/2}(s), i=0,\dots,N$,
- reconstruct numerically a primitive function (satisfying the interpolation constraints (2.8)) on $x_{i-1/2}(s), i=0,\dots,N$.

Hence, as in the pointwise semi-Lagrangian method, the algorithm of conservative methods is composed of two main steps: the computation of the characteristic curves, and the reconstruction step.

2.1.1. Computation of the characteristic curves

In the semi-Lagrangian method, we have to compute the characteristic curves between two consecutive time steps. In the case where a is constant, the integration of (2.7) is straightforward. Note that in the general case no information on a is known at any given time. Typically a may depend on g through a Poisson equation. To overcome this difficulty, we can use a two time step scheme as in [26] which is second order accurate.

We consider a time discretization

$$t^n=n\Delta t,\quad n\in\mathbb{N},\quad \Delta t>0,$$

and introduce $\bar{g}_i^n\approx\bar{g}_i(t^n)$ defined by (2.5). If we assume that $\bar{g}_i^{n-1}$ and $\bar{g}_i^n$ are known for $i=0,\dots,N-1$, we reconstruct the advection term $a^n(x)\approx a(t^n,x)$, which generally depends on $\bar{g}_i^n,\ i=0,\dots,N-1$. The value $x_{i+1/2}(t^{n-1})$ is then approximated by $X_{i+1/2}^{n-1}\approx X(t^{n-1})$ which is given by

$$\frac{dX(t)}{dt} = a^n(X(t)), \quad X(t^{n+1}) = x_{i+1/2}, \quad t \in [t^{n-1}, t^{n+1}] \tag{2.10}$$

and $\bar{g}_i^{n+1}$ can then be computed by formula (2.9), with $t = t^{n+1}$ and $s = t^{n-1}$:

$$\bar{g}_i^{n+1} = G^{n-1}\left(X_{i+1/2}^{n-1}\right) - G^{n-1}\left(X_{i-1/2}^{n-1}\right), \tag{2.11}$$

with $G^{n-1} \approx G(t^{n-1}, \cdot)$ computed with the values $\bar{g}_i^{n-1}, i = 0, \ldots, N-1$.

To compute $X_{i+1/2}^{n-1}$, we can either compute directly the feet of the characteristic ending at the interfaces $x_{i+1/2}$ as suggested by (2.10). We can also solve the same equation with final condition $X(t^{n+1}) = x_i$ to get X_i^{n-1} and then interpolate to obtain $X_{i+1/2}^{n-1}$. Practically, we use the latter approach with the approximation $X_{i+1/2}^{n-1} = \left(X_i^{n-1} + X_{i+1}^{n-1}\right)/2$, which remains second order accurate. In the sequel, we will present some numerical ways to compute the approximated solution X_i^{n-1}.

2.1.1.1. Midpoint formula. As in [26,14], a midpoint formula can be employed:

$$x_i - X_i^{n-1} = 2\Delta t a^n\left(\frac{x_i + X_i^{n-1}}{2}\right).$$

By writing $X_i^{n-1} = x_i - 2\alpha_i$, the displacement α_i can be computed at second order by solving the following one-dimensional fixed-point

$$\alpha_i = \Delta t\, a^n(x_i - \alpha_i). \tag{2.12}$$

In [26], a Newton algorithm is used but every iterative methods can be employed. We also mention [14] in which a Taylor expansion of the right hand side of (2.12) is performed; this strategy is equivalent to a Newton algorithm in which two iterations are imposed. However, the drawback of these algorithms is that they require the evaluation of the Jacobian matrix of a^n. A fixed-point algorithm can then be implemented. But, if we assume linear reconstruction of the advection term at points $(x_i - \alpha_i)$ (as it is supposed in [26,14]), an explicit algorithm can be used. The main steps of this new algorithm are detailed in the sequel.

Starting from (2.12) and denoting by $[x_j, x_{j+1}]$ the cell in which $(x_i - \alpha_i)$ falls, the linear reconstruction of a^n writes

$$\alpha_i = \Delta t\, [(1-\beta)a^n(x_j) + \beta a^n(x_{j+1})], \tag{2.13}$$

where β is such that

$$x_i - \alpha_i = x_j + \beta, \quad x_j = x_0 + j\Delta x, \quad x_i = x_0 + i\Delta x. \tag{2.14}$$

Injecting the expression of α_i into (2.13) leads to

$$\beta[\Delta x + \Delta t\,(a^n(x_{j+1}) - a^n(x_j))] = (i-j)\Delta x - \Delta t\, a^n(x_j), \tag{2.15}$$

from which an expression of β can be deduced

$$\beta = [(i-j)\Delta x - \Delta t\, a^n(x_j)]/[\Delta x + \Delta t(a^n(x_{j+1}) - a^n(x_j))]. \tag{2.16}$$

Now, it remains to determine the j index. To do that, it must be remarked that β given by (2.16) lives in the interval $[0,1]$. Hence, from (2.15), we can deduce an expression for $x_i = i\Delta x$

$$i\Delta x = j\Delta x + \Delta t\ a^n(x_j) + \beta[\Delta x + \Delta t(a^n(x_{j+1}) - a^n(x_j))].$$

Using the fact that $\beta \in [0,1]$, and by remarking that $[\Delta x + \Delta t(a^n(x_{j+1}) - a^n(x_j))] > 0$ provided that Δt is small enough, we deduce

$$i\Delta x \in [M_j, M_{j+1}], \text{ with } M_j = x_j + \Delta t\, a^n(x_j).$$

Under the assumption that Δt is small enough, the non-decreasing sequence $(M_j)_{j=0,\ldots,N-1}$ forms a non-uniform mesh from which the location of x_i can be found easily. The algorithm is then the following for each $i = 0, \ldots, N-1$:

- determination of j such that $x_i \in [M_j, M_{j+1}]$
- determination of β with (2.16)
- determination of α_i with (2.14)

This algorithm has been proved to be faster than classical iterative based methods. Obviously, it leads to the same displacement α_i.

2.1.1.2. Runge–Kutta methods. We can also employ classical techniques like Runge–Kutta (RK) schemes for the integration of (2.10). Note that even if we use high order RK schemes, we cannot achieve more than second order accuracy in time by the fact that we solve (2.10) instead of (2.7). However, we observe a better behaviour for a fourth order instead of a second order RK method. As an example, a second order RK method can be defined as follows:

$$k_1 = a^n(x_i), \quad k_2 = a^n(x_i - 2\Delta t\, k_1),$$

which leads to the following approximation

$$X_i^{n-1} = x_i - \Delta t(k_1 + k_2).$$

In our experiments, cubic spline interpolation have been used to evaluate the advection field a^n which is known on the grid points $x_i, i = 0, \dots, N-1$.

2.1.2. Reconstruction step

Once the computation of the characteristics is done, we have to explain how to interpolate the primitive function like in (2.11). We suppose known pointwise values of the primitive on the grid $G_{i-1/2} \approx G(s, x_{i-1/2}), i = 0, \dots, N$ and we want to reconstruct $G(s,x)$ (which we will denote thereafter $G(x)$ to simplify the notations) in order to be able to evaluate the primitive at the feet of the characteristic.

We consider here a periodic framework, which imposes that

$$G_{i-1/2} = G_{r-1/2} + qG_{N-1/2}, \text{ with } i = r + qN, r \in \{0, \dots, N-1\}, \quad q \in \mathbb{Z}.$$

Let us define for $\alpha \in \mathbb{R}$ an interpolation operator:

$$\Lambda_\alpha : \mathbb{R}^{\mathbb{Z}} \to \mathbb{R},$$

which satisfies

$$\Lambda_k(f_i) = f_k, k \in \mathbb{Z}.$$

We can write in a general way

$$G(x) = \Lambda_{\alpha+1/2}(G_{i-1/2}), \quad x = \alpha\Delta x.$$

In this subsection, we present some reconstructions.

2.1.2.1. Lagrange reconstruction. Let $d \in \mathbb{N}$. The centered Lagrange reconstruction of degree $2d+1$ is

$$\Lambda_\alpha(f_j) = \sum_{j=i-d}^{i+d+1} f_j \ell_j(\alpha), \quad i \leqslant \alpha < i+1, (f_j) \in \mathbb{R}^{\mathbb{Z}}, \quad \ell_j(\alpha) = \prod_{k=j-d,k\neq j}^{j+d+1} (\alpha - k)/(j-k),$$

which leads to

$$G(x) = \sum_{j=i-d}^{i+d+1} G_{j-1/2} L_j(x), \quad \forall x \in [x_{i-1/2}, x_{i+1/2}], \tag{2.17}$$

where

$$L_j(x) = \prod_{k=j-d,k\neq j}^{j+d+1} (x - x_{k-1/2})/(x_{j-1/2} - x_{k-1/2}).$$

For $d = 1$, this reconstruction corresponds to the PFC method introduced in [11] in which the slope limiters step is not performed. This approach and similar ones have been also introduced in [18]. See [19] for a more complete bibliography.

2.1.2.2. Spline reconstruction. The B-spline function is classically recursively defined by

$$B_d(x) = \int_{\mathbb{R}} B_{d-1}(t) B_0(x-t)dt, \quad B_0(x) = 1_{[-1/2,1/2]}(x).$$

The interpolation operator then writes

$$\Lambda_\alpha(f_j) = \sum_{i\in\mathbb{Z}} \eta_i(f_j) B_d(\alpha - i),$$

which leads to

$$G(x) = \sum_{i\in\mathbb{Z}} \eta_i B_d\Big(\frac{x - x_{i-1/2}}{\Delta x}\Big), \tag{2.18}$$

where the coefficients η_i are determined by the interpolating constraints

$$\sum_{i\in\mathbb{Z}} \eta_i B_d\Big(\frac{x_{j-1/2} - x_{i-1/2}}{\Delta x}\Big) = G_{j-1/2}, \quad j \in \mathbb{Z}.$$

In the case where $d = 3$, we obtain a cubic spline reconstruction

$$6B_3(x) = \begin{cases} (2-|x|)^3 & \text{if } 1 \leqslant |x| \leqslant 2, \\ 4 - 6x^2 + 3|x|^3 & \text{if } 0 \leqslant |x| \leqslant 1, \\ 0 & \text{otherwise}, \end{cases} \tag{2.19}$$

and we obtain the following linear system

$$A\eta = \begin{pmatrix} 4 & 1 & 0 & 0 & \cdots & 1 \\ 1 & 4 & 1 & 0 & & \vdots \\ 0 & 1 & 4 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & 0 & 1 & 4 & 1 \\ 1 & 0 & 0 & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \vdots \\ \eta_{N-2} \\ \eta_{N-1} \end{pmatrix} = 6 \begin{pmatrix} G_{1/2} + \frac{1}{6}G_{N-1/2} \\ G_{3/2} \\ \vdots \\ \vdots \\ G_{N-3/2} \\ \frac{5}{6}G_{N-1/2} \end{pmatrix}.$$

The coefficients $\eta_i, i \notin [0, N-1]$ are deduced from the solutions of the previous linear system by

$$\eta_{-i} = \eta_{-i+N} - G_{N-1/2}, \quad \forall i \in [0, N-1], \quad \eta_{i+N} = \eta_i + G_{N-1/2}, \quad \forall i \in [0, N-1].$$

This approach (for $d = 3$) has been introduced in [33] as the Parabolic Spline Method. Their formulation refers to the reconstruction of the function g which is a $\mathcal{C}^1$ piecewise parabolic function. The two formulations (by using primitive G or the function g) are completely equivalent, as already explained in [33].

2.1.2.3. Hermite reconstruction. We can consider a $\mathcal{C}^1$ reconstruction of $G(x)$, using a Hermite interpolation operator:

$$\Lambda_\alpha(f_j) = f_i + f_i'\alpha + (f_{i+1} - f_i - f_i')\alpha^2 + (f_{i+1}' + f_i' - 2(f_{i+1} - f_i))\alpha^2(\alpha - 1), \quad i \leqslant \alpha < i + 1.$$

The derivative value f_j' needs to be estimated. As in [12], we can use a fourth order accurate formula:

$$f_j' = \frac{1}{12\Delta x}\left(f_{j-2} - f_{j+2} + 8(f_{j+1} - f_{j-1})\right). \tag{2.20}$$

Note we can use a higher order formula, e.g. a sixth-order one:

$$f_j' = \frac{1}{60\Delta x}(f_{j+3} - f_{j-3} + 9(f_{j-2} - f_{j+2}) + 45(f_{j+1} - f_{j-1})). \tag{2.21}$$

This reconstruction with (2.20) (resp. (2.21)) corresponds to the PPM method [6] (resp. [7]), in which the slope limiters step is not performed. We will denote it by **PPM1**, resp. **PPM2**. We can also remark that the 4 points uncentered approximations

$$f_{j+}' = \frac{1}{6\Delta x}(-f_{j+2} + 6f_{j+1} - 3f_j - 2f_{j-1}), \quad f_{j-}' = \frac{1}{6\Delta x}(f_{j-2} - 6f_{j-1} + 3f_j + 2f_{j+1})$$

destroy the $\mathcal{C}^1$ property of the reconstruction, since the right (f_{j+}') and left derivative (f_{j-}') do not coincide and we recover the Lagrange reconstruction of degree 3, which will be denoted by **LAG**. By considering the average quantity $f_j' = \left(f_{j+}' + f_{j-}'\right)/2$, we regain the $\mathcal{C}^1$ property and recover formula (2.20).

We can even obtain the previous cubic spline reconstruction, with the following choice of f_j':

$$\frac{\Delta x}{3}\left(f_{j+1}' + 4f_j' + f_{j-1}'\right) = f_{j+1} - f_{j-1},$$

which corresponds to a Simpson approximation of $\int_{x_{i-1}}^{x_{i+1}} f'(x)dx$.

2.1.3. Implementation issues

In order to summarize, at time s, we have values $g_0^{\text{old}}, \ldots, g_{N-1}^{\text{old}}$, and we have the feet of the characteristics $x_{i+1/2}(s), i = 0, \ldots, N$ computed by an algorithm of Section 2.1.1. We can then define the displacements $\alpha_{i+1/2} = \frac{x_{i+1/2} - x_{i+1/2}(s)}{\Delta x}$. The new values $g_0^{\text{new}}, \ldots, g_{N-1}^{\text{new}}$ at time t are then computed by an algorithm of Section 2.1.2.

In the case of a Hermite type reconstruction, which contains all the reconstructions that we will consider in the numerical results, we describe here further some details of the numerical implementation.

2.1.3.1. Constant advection case. We set $\alpha = \alpha_{i+1/2}$, in the case of constant advection. We then compute for $i = 0, \ldots, N-1$,

$$G_i = x(1-x)^2 g_{j_i^+}' + x^2(x-1)g_{(j_i+1)^-}' + x^2(3-2x)g_{j_i}^{\text{old}}, \text{ with } i + \alpha = j_i + x, \quad 0 \leqslant x < 1.$$

The new values are then given by

$$g_i^{\text{new}} = g_{j_i}^{\text{old}} + (G_{i+1} - G_i).$$

For the Lagrange reconstruction of degree 3 (**LAG**), the derivatives are given by

$$g'_{j^+} = \frac{5}{6}g_j^{\text{old}} - \frac{1}{6}g_{j+1}^{\text{old}} + \frac{1}{3}g_{j-1}^{\text{old}}, \quad g'_{j+1^-} = \frac{5}{6}g_j^{\text{old}} + \frac{1}{3}g_{j+1}^{\text{old}} - \frac{1}{6}g_{j-1}^{\text{old}}. \tag{2.22}$$

For **PPM1**, we have

$$g'_{j^+} = g'_{j^-} = \frac{7}{12}\left(g_j^{\text{old}} + g_{j-1}^{\text{old}}\right) - \frac{1}{12}\left(g_{j-1}^{\text{old}} + g_{j-2}^{\text{old}}\right). \tag{2.23}$$

For **PPM2**, we have

$$g'_{j^+} = g'_{j^-} = \frac{1}{60}\left(g_{j+3}^{\text{old}} - g_{j-3}^{\text{old}} + 9\left(g_{j-2}^{\text{old}} - g_{j+2}^{\text{old}}\right) + 45\left(g_{j+1}^{\text{old}} - g_{j-1}^{\text{old}}\right)\right). \tag{2.24}$$

Finally, for **PSM**, we also have $g'_{i^+} = g'_{i^-} = g'_i$, and the derivative g'_i is obtained by first computing the solution of the almost tridiagonal system

$$g'_{i-1} + 4g'_i + g'_{i+1} = 3\left(g_i^{\text{old}} + g_{i+1}^{\text{old}}\right). \tag{2.25}$$

2.1.3.2. General case. In the general case, we compute for $i = 0, \ldots, N-1$,

$$G_i = x(1-x)^2 g'_{j_i^+} + x^2(x-1)g'_{(j_i+1)^-} + x^2(3-2x)g_{j_i}^{\text{old}}, \text{ with } i + \alpha_{i-1/2} = j_i + x, \quad 0 \leqslant x < 1,$$

with the same definition of the derivatives, and the new values are given by

$$g_i^{\text{new}} = \sum_{k=j_i}^{j_{i+1}-1} g_k^{\text{old}} + (G_{i+1} - G_i).$$

2.1.4. Slope limiters

In this subsection, we focus on the description of different filters which can be adapted to the previous reconstruction. It is well known that high order schemes can generate new extrema, violate the monotonicity and develop numerical oscillations. In order to avoid or reduce these problems, filters have been introduced. This point has been studied by many authors and remains the subject of recent developments (see e.g. [11,15,33,7,30] and references therein).

One first physical requirement, which is our main objective is the preservation of the positivity. Note that this property is global and well defined. A more general property is the preservation of the maximum principle; we should here distinguish global and local extrema. The maximum and minimum are well defined for the initial function (which is generally given by a formula) and are good candidates for global extrema during all the simulation, as done in [11]. The use of local bounds is more ambiguous. Indeed even for global bounds computed via the current solution, the maximum value at current time can decrease during the simulation due to the numerical diffusion. Thus, filters based on such values may enforce artificially the decrease of the maximum and therefore accelerate the diffusion even more. In fact, we should try to keep already existing extrema, not generate new ones numerically and take care that we do not damage the order of convergence of the scheme in regions where the solution is smooth. In [30], a linear reconstruction in the nearest cells enables to determine a local extremum. Other strategies consist in limiting the derivatives of the reconstructed function (which can be high where the solution has not a smooth behaviour) as in [7,16].

We have tested several filters. Instead of presenting all of them, we will here only deal with a few of them which seemed relevant to us. A filter will here have three ingredients; at first the extrema definition, then the extrema limitation procedure which enforces that the reconstruction does not violate the extrema definition, and finally an oscillation limitation procedure. We consider here a reconstruction like in Section 2.1.3, and we will modify the values $g'_{j+}, g'_{(j+1)-}$.

2.1.4.1. Extrema definition. We have at first to define the bounds $g_{\min}, g_{\max}$ in which we want to keep our solution. For this, we will consider:

- *positive extrema:* $g_{\min} = 0, g_{\max} = \infty$,
- *global extrema:* $g_{\min} = \min g^0(x), g_{\max} = \max g^0(x)$, where g^0 is the initial function which will be advected,
- *Umeda extrema:* local extrema are defined as in [30]:

$$g_{\max} = \min(\max g^0(x), \max(g_{\max 1}, g_{\max 2})), g_{\min} = \max(\min g^0(x), \min(g_{\min 1}, g_{\min 2})),$$

with

$$g_{\max1} = \max\left(\max\left(g_{i-1}^{\text{old}}, g_i^{\text{old}}\right),\ \min\left(2g_{i-1}^{\text{old}} - g_{i-2}^{\text{old}}, 2g_i^{\text{old}} - g_{i+1}^{\text{old}}\right)\right)$$
$$g_{\max2} = \max\left(\max\left(g_{i+1}^{\text{old}}, g_i^{\text{old}}\right),\ \min\left(2g_{i+1}^{\text{old}} - g_{i+2}^{\text{old}}, 2g_i^{\text{old}} - g_{i-1}^{\text{old}}\right)\right)$$
$$g_{\min1} = \min\left(\min\left(g_{i-1}^{\text{old}}, g_i^{\text{old}}\right),\ \max\left(2g_{i-1}^{\text{old}} - g_{i-2}^{\text{old}}, 2g_i^{\text{old}} - g_{i+1}^{\text{old}}\right)\right)$$
$$g_{\min2} = \min\left(\min\left(g_{i+1}^{\text{old}}, g_i^{\text{old}}\right),\ \max\left(2g_{i+1}^{\text{old}} - g_{i+2}^{\text{old}}, 2g_i^{\text{old}} - g_{i-1}^{\text{old}}\right)\right)$$

In the non-constant case, the positive extrema can be defined. However, in the general case, the possible contraction of the volume can lead to value g_i outside the extrema bounds (see e.g. the case of a constant initial condition). Hence, we propose to relax the definition of the extrema as follows: we replace $g_{\min}$ by $\min\,(g_i^{\text{old}}, g_{\min})$ and $g_{\max}$ by $\max\,(g_i^{\text{old}}, g_{\max})$. Note that this procedure has no effect in the constant advection case.

2.1.4.2. Extrema limitation. The *Hyman filter* is given by the following algorithm

$$g' = \max\left(g', \max\left(g_{\min}, -2g_{\max} + 3g_{j_i}^{\text{old}}\right)\right);$$
$$g' = \min\left(g', \min\left(g_{\max}, 3g_{j_i}^{\text{old}} - 2g_{\min}\right)\right);$$

where g' takes successively the value g'_{j+} and $g'_{(j+1)+}$. This filter ensures that the functions $x \to G_i(x) - xg_{\min}$ and $x \to xg_{\max} - G_i(x)$ are non-decreasing on $[0,1]$, if $g_{\min} \leqslant g_{j_i}^{\text{old}} \leqslant g_{\max}$. Thus, the positive extrema are preserved and in the constant advection case, the global extrema are also preserved. We could also use the PFC limiter (see [11]), which was intended for the LAG reconstruction. Another possibility is to modify $g'_{j+}, g'_{(j+1)-}$ at least possible so that the constraint $g_{\min} \leqslant G'_i(x) \leqslant g_{\max}$, for all $0 \leqslant x \leqslant 1$ is satisfied. As an example, we can solve the minimization problem for $\left|\left(g'_{j+}\right)^{\text{new}} - g'_{j+}\right| + \left|\left(g'_{(j+1)-}\right)^{\text{new}} - g'_{(j+1)-}\right|$. However, it may be not always a good idea or not useful to remain the nearest possible to the first reconstructed derivative, which may be sometimes a bad approximation.

2.1.4.3. Oscillation limitation. We have seen that the derivative can be computed by the PSM or LAG (supposed more diffusive) reconstruction. We can even use the lower order formula $g'_{j,m} = \left(g_j^{\text{old}} + g_{j-1}^{\text{old}}\right)/2$, which leads to an even more diffusive reconstruction. We have added the following filter which tends to privilege the derivative of the most diffusive reconstruction if the error between the two reconstructions is too large. The aim is to damp the spurious oscillations, which are detected when the error is large. Such an approach has been performed in [7]: one first step consists in looking for the closest reconstruction (among the left, right Lagrange derivatives (2.22) and the PPM one given by (2.23) or (2.24)) to $g'_{j,m}$. This strategy enables to add some accuracy to the low order formula $g'_{j,m} = \left(g_j^{\text{old}} + g_{j-1}^{\text{old}}\right)/2$. Our approach is similar, we compare the Lagrange derivative g'_{j+} to the PSM reconstructed one (using (2.25) in order to correct in the best sense the lower order formula $g'_{j,m}$. In other words, if $\left(g'_{j+,LAG} - g'_{j,m}\right)\left(g'_{j,PSM} - g'_{j,m}\right) < 0,\ g'_{j+} = g'_{j,m}$, else

$$g'_{j+} = g'_{j,m} + s\ \min\left(C|g'_{j+,LAG} - g_{j,m}|, |g'_{j,PSM} - g'_{j,m}|\right),\ \text{ with } s = sign\left(g'_{j,PSM} - g'_{j,m}\right), \tag{2.26}$$

where $C > 1$. Similarly, we modify $g'_{(j+1)-}$ as follows. Let $s = sign\left(g'_{j+1,PSM} - g'_{j+1,m}\right)$. If $\left(g'_{(j+1)-,LAG} - g'_{j+1,m}\right)\left(g'_{j+1,PSM} - g'_{j+1,m}\right) < 0,\ g'_{(j+1)-} = g'_{j+1,m}$, else

$$g'_{(j+1)-} = g'_{j+1,m} + s\ \min\left(C|g'_{(j+1)-,LAG} - g'_{j+1,m}|, |g'_{j+1,PSM} - g'_{j+1,m}|\right).$$

Obviously, the scheme is dependent from the choice of C, but we find that the best compromise is $C = 2.5$ in our numerical tests. By increasing C, the minimum in (2.26) will be $g'_{j,PSM}$ so that the filter will have no effect. By decreasing C, the method will have a more diffusive behaviour.

2.2. Conservative forward semi-Lagrangian methods for one-dimensional problems

Another strategy to update in a conservative way the unknowns consists in advancing in time the mesh points. We take the notations of Section 2.1, we suppose here to know the solution at time t (that is $\bar{g}_i(t), i = 0, \dots, N-1$) and want to compute it at time s (that is $\bar{g}_i(s), i = 0, \dots, N-1$). We reconstruct here the primitive function $G(s,x) = \frac{1}{\Delta x}\int_{x_{-1/2}(s)}^{x} g(s,y)dy$. Note that we change the integration constant of the primitive in comparison to the backward case, in order to simplify the computations. Using (2.6), the primitive function G has to satisfy

$$G(s, x_{i-1/2}(s)) = \sum_{k=0}^{i-1} \frac{1}{\Delta x}\int_{x_{i-1/2}(s)}^{x_{i+1/2}(s)} g(s,x)dx = \sum_{k=0}^{i-1} \bar{g}_k(t), \quad i = 1, \dots, N, G(s, x_{-1/2}(s)) = 0. \tag{2.27}$$

We then have to interpolate the primitive function on the grid points to update the unknowns:

$$\bar{g}_i(s) = G(s, x_{i+1/2}) - G(s, x_{i-1/2}). \tag{2.28}$$

To summarize, for going from time t to s, we need to

- compute at least numerically the values $x_{i-1/2}(s), i = 0, \dots, N$,
- reconstruct numerically a primitive function (satisfying the interpolation constraints (2.27)) on $x_{i-1/2}, i = 0, \dots, N$.

In the rest of the section, the two steps of the method are detailed.

2.2.1. Computation of the characteristics curves

The same strategy used in the previous method are employed here. In the case where a is constant, the integration is straightforward and, as mentioned in Section 2.1.1, when a depends of the unknown, a two time step algorithm is used. We introduce $\bar{g}_i^n \approx g_i(t^n)$, assume that $\bar{g}_i^{n-1}$ and $\bar{g}_i^n$ are known and that the advection term $a(t^n, X(t^n))$ can be computed with $\bar{g}_i^n, i = 0, \dots, N-1$. The value $x_{i+1/2}(t^{n+1})$ is then approximated by $X_{i+1/2}^{n+1} \approx X(t^{n+1})$ thanks to

$$\frac{dX(t)}{dt} = a^n(X(t)), \quad X(t^{n-1}) = x_{i+1/2}, \quad t \in [t^{n-1}, t^{n+1}], \tag{2.29}$$

and $\bar{g}_i^{n+1}$ can be computed by formula (2.28) with $s = t^{n+1}$:

$$\bar{g}_i^{n+1} = G^{n+1}(x_{i+1/2}) - G^{n+1}(x_{i-1/2}),$$

where $G^{n+1}(\cdot) \approx G(t^{n+1}, \cdot)$ is the primitive which is reconstructed from the values $\bar{g}_i^{n-1}, i = 0, \dots, N-1$, using the interpolation conditions (2.27), with $t = t^{n-1}$ and $s = t^{n+1}$:

$$G^{n+1}\left(X_{i-1/2}^{n+1}\right) = \sum_{k=0}^{i-1} \bar{g}_k^{n-1}, \quad i = 1, \dots, N, \quad G^{n+1}\left(X_{-1/2}^{n+1}\right) = 0.$$

To compute $X_{i+1/2}^{n+1}$, we can use the same methods as for the backward case, in which we make the changes

$$X_i^{n-1} \to X_i^{n+1}, \quad \Delta t \to -\Delta t.$$

2.2.2. Reconstruction step on a non-uniform mesh

We suppose known pointwise values of the primitive on the uniform mesh $G_{i-1/2} \approx G\left(s, X_{i-1/2}^{n+1}\right), i = 0, \dots, N$ and we want to reconstruct $G(s,x)$ (which we will denote thereafter $G(x)$ to simplify the notations) in order to be able to evaluate the primitive at the uniform grid.

We consider here a periodic framework, which imposes that

$$X_{i-1/2}^{n+1} = X_{r-1/2}^{n+1} + q\left(X_{N-1/2}^{n+1} - X_{-1/2}^{n+1}\right), \quad \text{with } i = r + qN, \quad r \in \{0, \dots, N-1\}, \quad q \in \mathbb{Z},$$

and we also have

$$G_{i-1/2} = G_{r-1/2} + qG_{N-1/2}.$$

Let us define for a mesh $Y = (y_j)_{j\in\mathbb{Z}}$ (y_j is an increasing sequence) and $x \in \mathbb{R}$ an interpolation operator:

$$\Lambda_x^Y : \mathbb{R}^{\mathbb{Z}} \to \mathbb{R},$$

which satisfies

$$\Lambda_{y_k}^Y(g_i) = g_k, \quad k \in \mathbb{Z}.$$

We can write in a general way

$$G(x) = \Lambda_x^X(G_{i-1/2}), \quad X = \left(X_{j-1/2}^{n+1}\right)_{j\in\mathbb{Z}}.$$

In this subsection, we present some reconstructions Λ_x^Y.

2.2.2.1. Lagrange reconstruction. Let $d \in \mathbb{N}$. The centered Lagrange reconstruction of degree $2d+1$ is

$$\Lambda_x^Y(f_j) = \sum_{j=i-d}^{i+d+1} f_j \ell_j^Y(x), \quad y_i \leqslant x < y_{i+1}, \quad (f_j) \in \mathbb{R}^{\mathbb{Z}}, \quad \ell_j^Y(x) = \prod_{k=j-d, k\neq j}^{j+d+1} (x - y_k)/(y_j - y_k).$$

2.2.2.2. Spline reconstruction. The *B*-spline function is classically recursively defined by

$$B_{j,d}^Y(x) = \frac{x - y_j}{y_{j+d} - y_j} B_{j,d-1}^Y(x) + \frac{y_{j+d+1} - x}{y_{j+d+1} - y_{j+1}} B_{j+1,d-1}^Y(x), \quad B_{j,0}^Y(x) = 1_{[y_j, y_{j+1}]}(x).$$

The interpolation operator then writes $\Lambda_x^Y(f_j) = \sum_{i\in\mathbb{Z}} \eta_i(f_j) B_{i,d}^Y(x)$, where the coefficients η_i are determined by the interpolating constraints

$$\sum_{i\in\mathbb{Z}} \eta_i B_{i,d}^Y(y_j) = f_j, \quad j \in \mathbb{Z}.$$

In the case where $d=3$, the primitive writes $G(x)=\sum_{i\in\mathbb{Z}}\eta_i B^X_{i,3}(x)$, with

$$\sum_{i\in\mathbb{Z}}\eta_i B^X_{i,3}\left(X^{n+1}_{j-1/2}\right)=G_{j-1/2},\quad j\in\mathbb{Z}.$$

Using (2.19), the coefficients η_i are solution to the following linear system

$$A\eta=\begin{pmatrix} D_0 & C_0 & 0 & 0 & \cdots & A_0\\ A_1 & D_1 & C_1 & 0 & & \vdots\\ 0 & A_2 & D_2 & C_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & & 0 & A_{N-2} & D_{N-2} & C_{N-2}\\ C_{N-1} & 0 & 0 & 0 & A_{N-1} & D_{N-1}\end{pmatrix}\begin{pmatrix}\eta_0\\ \eta_1\\ \vdots\\ \vdots\\ \eta_{N-2}\\ \eta_{N-1}\end{pmatrix}=\begin{pmatrix}G_{1/2}+A_0G_{N-1/2}\\ G_{3/2}\\ \vdots\\ \vdots\\ G_{N-3/2}\\ (1-C_{N-1})G_{N-1/2}\end{pmatrix}$$

where the components of the matrix are defined for $i=0,\dots,N-1$, by

$$A_i=B^X_{i-1,3}\left(X^{n+1}_{i-1/2}\right),\quad C_i=B^X_{i+1,3}\left(X^{n+1}_{i-1/2}\right),\quad D_i=B^X_{i,3}\left(X^{n+1}_{i-1/2}\right),$$

which leads to the explicit formulae

$$A_i=\frac{(y_{i+1}-y_i)^2}{(y_{i+1}-y_{i-1})(y_{i+1}-y_{i-2})},\quad C_i=\frac{(y_i-y_{i-1})^2}{(y_{i+1}-y_{i-1})(y_{i+2}-y_{i-1})},\quad D_i=1-A_i-C_i,$$

with $y_i=X^{n+1}_{i-1/2}$. Let us mention that a good behaviour of the present method requires a good approximation of the characteristics curves $X^{n+1}_{i+1/2}$ at time t^{n+1}. Indeed, this strong dependence can be explained by the fact that they refer to the conditions of interpolation.

3. The constant advection case

In this section, we are concerned with a constant advection field, so (2.4) can be rewritten equivalently in a conservative form

$$\frac{\partial g}{\partial t}+\frac{\partial(ag)}{\partial x}=0,\quad x\in I\subset\mathbb{R},$$

or in an advective form

$$\frac{\partial g}{\partial t}+a\frac{\partial g}{\partial x}=0,\quad x\in I\subset\mathbb{R}.$$

3.1. Equivalence between conservative and advective approach

We will make here the link between the conservative and advective approaches. We first recall the advective approach and give a condition on the interpolation for which we will prove that both approaches are algebraically equivalent. As a particular case, we will notice that Lagrange or splines constructions previously introduced realize this condition. We underline that this equivalence only holds for uniform meshes in a constant advection case and with periodic boundary conditions.

3.1.1. Advective approach

We consider here pointwise values $g_i(s)=g(s,x_i)$, which are updated this time through

$$g_i(t)=g(s,x_i-a(t-s)),s,\quad t\in\mathbb{R}.$$

For getting the function at time t, we thus need a reconstruction $g(x)$ of the function at time s in order to be able to evaluate the function at the feet of the characteristics $x_i-a(t-s)$. We then write with the same interpolation operator defined for the conservative approach

$$g(x)=\Lambda_\alpha(g_i),\quad x=\alpha\Delta x.$$

3.1.2. Equivalence conditions

The conservative and advective approach are then equivalent iff

$$G(x_{i+1/2}+\alpha\Delta x)-G(x_{i-1/2}+\alpha\Delta x)=g(x_i+\alpha\Delta x),$$

that is

$$\Lambda_{\alpha+1}(G_{i-1/2})-\Lambda_\alpha(G_{i-1/2})=\Lambda_\alpha(g_i).\tag{3.1}$$

We identify here the pointwize value g_i with the average one $\bar{g}_i$, so that we have $G_{i+1/2} - G_{i-1/2} = g_i$. This corresponds to a midpoint approximation of $\bar{g}_i$ at initial time.

The property (3.1) is true under the following conditions:

- Λ_α is a linear operator
- $\Lambda_{\alpha+1}(f_i) = \Lambda_\alpha(f_{i+1})$ (shift invariance property).

Indeed, we have from (3.1)

$$\Lambda_{\alpha+1}(G_{i-1/2}) - \Lambda_\alpha(G_{i-1/2}) = \Lambda_\alpha(G_{i+1/2}) - \Lambda_\alpha(G_{i-1/2}) = \Lambda_\alpha(G_{i+1/2} - G_{i-1/2}) = \Lambda_\alpha(g_i).$$

3.1.3. Examples

In the case of Lagrange reconstruction, we have

$$\ell_j(\alpha+1) = \prod_{k=j-d,k\neq j}^{j+d+1} (\alpha+1-k)/(j-1-k+1) = \prod_{k=j-1-d,k\neq j-1}^{j+d} (\alpha-k)/(j-1-k) = \ell_{j-1}(\alpha),$$

and thus for $i \leqslant \alpha < i+1$,

$$\Lambda_{\alpha+1}(f_j) = \sum_{j=i+1-d}^{i+d+2} f_j \ell_j(\alpha+1) = \sum_{j=i+1-d}^{i+d+2} f_j \ell_{j-1}(\alpha) = \sum_{j=i-d}^{i+d+1} f_{j+1}\ell_j(\alpha) = \Lambda_\alpha(f_{j+1}),$$

and the conditions are then fulfilled.

In the case of the spline reconstruction, we have

$$\Lambda_{\alpha+1}(f_j) = \sum_{i\in\mathbb{Z}} \eta_i(f_j) B_d(\alpha-(i-1)) = \sum_{i\in\mathbb{Z}} \eta_{i+1}(f_j) B_d(\alpha-i).$$

We have the interpolation conditions

$$\Lambda_{k+1}(f_j) = f_{k+1} = \sum_{i\in\mathbb{Z}} \eta_{i+1}(f_j) B_d(k-i), \quad k\in\mathbb{Z}. \tag{3.2}$$

We have also

$$\Lambda_\alpha(f_{j+1}) = \sum_{i\in\mathbb{Z}} \eta_i(f_{j+1}) B_d(\alpha-i), \quad \text{with } \sum_{i\in\mathbb{Z}} \eta_i(f_{j+1}) B_d(k-i) = f_{k+1}, \quad k\in\mathbb{Z}. \tag{3.3}$$

Thus, from (3.2) and (3.3), we deduce that $\eta_{i+1}(f_j) = \eta_i(f_{j+1})$, by unicity of the solution (see [10]) which leads to

$$\Lambda_{\alpha+1}(f_j) = \sum_{i\in\mathbb{Z}} \eta_i(f_{j+1}) B_d(\alpha-i) = \Lambda_\alpha(f_{j+1}).$$

In the case of the Hermite reconstruction, we also check that the conditions are fulfilled.

3.1.4. Counterexample

We consider the following quadratic Lagrange reconstruction

$$\Lambda_{2i+\alpha}(f_j) = f_{2i} + (f_{2i+1} - f_{2i})\alpha + \frac{f_{2i+2} - 2f_{2i+1} + f_{2i}}{2}\alpha(\alpha-1), \quad 0 \leqslant \alpha < 2.$$

We then have for $0 \leqslant \alpha < 1$,

$$\Lambda_{2i+\alpha}(G_{j-1/2}) = G_{2i-1/2} + (G_{2i+1/2} - G_{2i-1/2})\alpha + \frac{G_{2i+3/2} - 2G_{2i+1/2} + G_{2i-1/2}}{2}\alpha(\alpha-1),$$

and thus

$$\Lambda_{2i+\alpha+1}(G_{j-1/2}) - \Lambda_{2i+\alpha}(G_{j-1/2}) = g_{2i} + (g_{2i+1} - g_{2i})\alpha \neq \Lambda_{2i+\alpha}(g_j).$$

Note that advective method is here not conservative, since we have for $0 \leqslant \alpha < 1$,

$$\sum_{i=0}^{N-1} \Lambda_{\alpha+i}(g_j) = \sum_{i=0}^{N/2-1} \Lambda_{\alpha+2i}(g_j) + \sum_{i=0}^{N/2-1} \Lambda_{\alpha+2i+1}(g_j) = (1-2\alpha+2\alpha^2)\sum_{i=0}^{N/2-1} g_{2i} + (1+2\alpha-2\alpha^2)\sum_{i=0}^{N/2-1} g_{2i+1},$$

so that this quantity can depend on α which is not the case for a conservative approach.

3.2. Numerical results

We present the numerical results for the different numerical schemes we proposed. We first study 1D linear advection before the Vlasov–Poisson case.

Let us define for $\alpha \in \mathbb{R}$ and $N \in \mathbb{N}^*$ a transport operator $\mathcal{T}_{\alpha,N} : \mathbb{R}^N \to \mathbb{R}^N$. If $(f_0,\dots,f_{N-1})$ is a discretization of a function f, then $\mathcal{T}_{\alpha,N}(f_0,\dots,f_{N-1})$ should be a discretization of the shifted function $x \to f(x+\alpha)$. We also denote by $\mathcal{T}^x_\alpha : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ (resp. $\mathcal{T}^v_\alpha : \mathbb{R}^{N_v} \to \mathbb{R}^{N_v}$) a transport operator which shifts along the x (resp. v) direction. The transport operator $\mathcal{T}_{\alpha,N}$ denotes one of the different reconstructions of the Section 2.1.2. Indeed, we compare the three different reconstructions: Lagrange (LAG), PPM1 and splines (PSM). For these three reconstructions, we apply filters presented in 2.1.4: the extrema limitation uses the Hyman approach and we compare the influence of the global extrema to the Umeda ones [30]. The extensions UM and GL are added to the three reconstructions LAG, PPM and PSM. Let us note that the PPM2 reconstruction gives slightly better results than PPM1 one. However, we chose to show only results associated to PPM1 in the sequel. We also implemented classical methods of the literature: the PFC method of [11], the UMEDA method of [30] (they are both based on a Lagrange reconstruction but the first one uses global filter whereas the latter one uses the Umeda filter), and the SPL method (standard advective cubic splines reconstruction without filter). Note that in the present constant advection case, SPL and PSM are the same methods. We have added a last method PSM2, which consists in applying the procedure of the paragraph "oscillation limitation" of Section 2.1.4 (i.e. PSM reconstruction with oscillation limitation by (2.26)) and then the Hyman approach with the global extrema definition. Note that the Forward Update method introduced in Section 2.2 is not shown in the present case; indeed the displacements are constant so that only uniform mesh are generated by the characteristics and the method is completely equivalent to the SPL or the PSM methods.

3.2.1. Application to the linear advection

We proposed to first apply the new methods by solving the 1D linear advection equation

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = 0, \quad x \in [0, L]$$

with a constant velocity $v = 1$. We intend to test the different methods proposed above by using rectangular and sinusoidal initial profiles, as in [30]. We impose periodic boundary conditions and the numerical parameters are chosen as follows: $\Delta t = 0.1, \Delta x = 1, L = 80$ so that $N_x = 80$ (the courant number is equal to 0.1). The numerical solution is compared to the analytical one after 8000 iterations. We also implement a third case from [33], the numerical parameters of which are: $\Delta t = 1/71, \Delta x = 1/50, L = 1$ so that $N_x = 50$ (the courant number is equal to $50/71 \approx 0.7$). The numerical solution is transported during 71 time steps. This test enables to test local filters.

The numerical results are shown in Fig. 1 in which the analytical solution is plotted for comparison.

The first remarks note that Lagrange based methods are the most diffusive since the amplitude of the sinusoidal waves are strongly damped, whereas it is well preserved for the other methods (PPM and PSM based methods). For rectangular wave (left column), the action of the limiters is more clear. The SPL method creates artificial extrema whereas other methods are bounded between 0 and 1. We can also notice the fact that Lagrange based methods are more diffusive around the discontinuities. For PPM and PSM reconstructions, even if the local filter avoids oscillations, the numerical results are not very good. The new filter which leads to PSM2 leads to more accurate results, since it does not present oscillation and discontinuities are not too diffused. For the third test (which does not contain discontinuity, only strong gradients), the influence of local filters can be emphasized. As noticed in [30], the Umeda filter leads to non-oscillatoring results; the application of this local filter (together with the Hyman limitation) to LAG, PSM or PPM then ensures that local maxima are preserved so that no artificial ones are created. Let us remark the difference between LAG-UM and UMEDA which shows the influence of the Hyman limitation. The new filter applied to PSM2 is not non-oscillatoring since small oscillations remain around local extrema, but as noticed before PSM2 remains positive and respect the maximum principles.

3.2.2. Application to the Vlasov–Poisson model

As another application of the constant advection case, we are concerned with the Vlasov–Poisson model, the unknown of which $f = f(t,x,v)$ is the electron distribution function. It depends on the spatial variable $x \in [0,L]$ where $L > 0$ is the size of the domain, the velocity $v \in \mathbb{R}$ and the time $t \geqslant 0$. The time evolution of this distribution function is given by the following phase space transport equation, the Vlasov equation

$$\frac{\partial f}{\partial t} + v\partial_x f + E(t,x)\partial_v f = 0, \tag{3.4}$$

with the initial condition

$$f(0,x,v) = f_0(x,v).$$

The electric field $E(t,x)$ is given by the coupling with the distribution function f through the Poisson equation

$$\partial_x E(t,x) = \rho(t,x) - \rho^i, \quad \int_0^L E(t,x)\mathrm{d}x = 0, \tag{3.5}$$

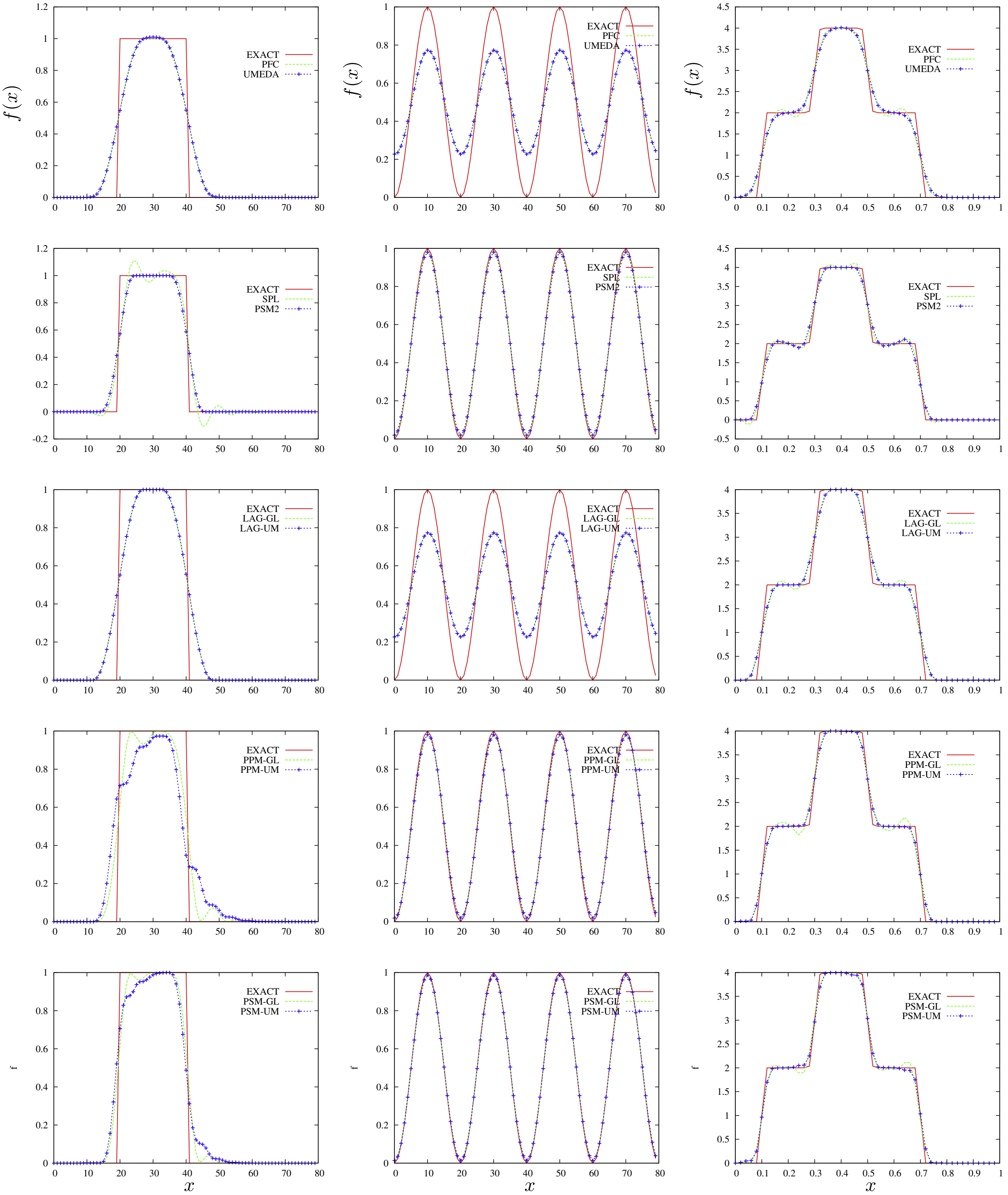


Fig. 1. Numerical results for the linear advection. Left: rectangular wave; middle: sinusoidal wave; right: double rectangular wave.

where the electron charge density ρ is given by $\rho(t,x) = \int_{\mathbb{R}} f(t,x,v)dv$ and ρ^i denotes the ion density. In this work, we restrict ourselves to a uniform background of ions which leads to $\rho^i = 1$ after a suitable choice of dimensionless parameters.

In view of finite volumes formulation, it will be convenient to re-write the Vlasov equation into a conservative form

$$\frac{\partial f}{\partial t} + \partial_x(vf) + \partial_v(E(t,x)f) = 0. \tag{3.6}$$

The Vlasov–Poisson model preserves some physical quantities with time which will be analysed and compared for the different numerical methods. First of all, the Vlasov–Poisson equation preserves the L^p norms for $p \geqslant 0$

$$\frac{d}{dt}\|f(t)\|_{L^p} = 0. \tag{3.7}$$

The total energy is also constant in time

$$\frac{d}{dt}\mathcal{E}(t) = \frac{d}{dt}\mathcal{E}_k(t) + \frac{d}{dt}\mathcal{E}_e(t) = \frac{d}{dt}\int_0^L \int_{\mathbb{R}} f(t,x,v)\frac{|v|^2}{2}dxdv + \frac{1}{2}\frac{d}{dt}\int_0^L \int_{\mathbb{R}} |E(t,x)|^2 dxdy, \tag{3.8}$$

where $\mathcal{E}_e$ and $\mathcal{E}_k$ denote the electric and kinetic energy, respectively. From a numerical point of view, the good conservation of these different quantities is an important feature for Vlasov simulations.

3.2.2.1. The general algorithm. We first review the main steps of a semi-Lagrangian method in the case of directional splitting with constant advection, which is applied for the discretization of the Vlasov–Poisson model.

The unknown quantities are then $f^n_{k,\ell}$ which are approximations of $f(t^n, x_k, v_\ell)$. We suppose periodic boundary conditions so that we only have to compute at each time t^n

$$f^n_{k,\ell}, \text{ for } k = 0,\dots,N_x - 1, \quad \ell = 0,\dots,N_v - 1.$$

The time splitting algorithm then reads (see [5])

Step 0. Initialization: $f_{k,\ell} = f_0(x_k, v_\ell),\ k = 0,\dots,N_x - 1,\ \ell = 0,\dots,N_v - 1.$

Step 1. Half time step shift along the x-axis: For each $\ell = 0,\dots,N_v - 1, (f_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}^x_\alpha\left((f_{k,\ell})_{k=0}^{N_x-1}\right)$ with $\alpha = -v_\ell\Delta t/2$.

Step 2. Computation of the charge density and the electric field by integrating (3.5).

Step 3. Shift along the v-axis: For each $k = 0,\dots,N_x - 1, (f_{k,\ell})_{\ell=0}^{N_v-1} \to \mathcal{T}^v_\alpha\left((f_{k,\ell})_{\ell=0}^{N_v-1}\right)$ with $\alpha = -E_k\Delta t$.

Step 4.a Half time step shift along the x-axis: For each $\ell = 0,\dots,N_v - 1, (f_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}^x_\alpha\left((f_{k,\ell})_{k=0}^{N_x-1}\right)$ with $\alpha = -v_\ell\Delta t/2$.

Step 4.b We have $f^n_{k,\ell} = f_{k,\ell}$, for $k = 0,\dots,N_x - 1, \ell = 0,\dots,N_v - 1$.

Step 4.c Half time step shift along the x-axis: For each $\ell = 0,\dots,N_v - 1, (f_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}^x_\alpha\left((f_{k,\ell})_{k=0}^{N_x-1}\right)$ with $\alpha = -v_\ell\Delta t/2$.

Step 5. $n \to n+1$ and loop to **Step 2**.

Note that if we make no diagnostic of the distribution function, we can simplify **Step 4a–c** into

Step 4. Shift along the x-axis:
For each $\ell = 0,\dots,N_v - 1, (f_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}^x_\alpha\left((f_{k,\ell})_{k=0}^{N_x-1}\right)$ with $\alpha = -v_\ell\Delta t$.

In the sequel, we present numerical results for the Vlasov–Poisson equation for which several choices of transport operators $\mathcal{T}_{\alpha,N}$ are performed.

3.2.2.2. Numerical results. We are interested in testing our numerical schemes to the nonlinear Vlasov–Poisson model. Three tests are used to that purpose: the strong Landau damping, the bump-on-tail and the two stream instability test cases.

3.2.2.2.1. Strong Landau damping. The initial condition associated to the Vlasov–Poisson model is

$$f(x,v,t=0) = \frac{1}{\sqrt{2\pi}}\exp(-v^2/2)(1 + 0.5\cos(kx)), \quad x \in [0,L], v \in [-v_{\max}, v_{\max}],$$

where $k = 0.5, v_{\max} = 6, L = 2\pi/k$ corresponds to the length of the domain in the x-direction. The numerical parameters are $N_x = N_v = 128, \Delta t = 0.1$ (so that the Courant number is 6.1) and the number of iterations is 1000.

This test case presents very fine structures which move in the phase space due to the free transport term. Hence, after a first (linear) phase during which the amplitude of the electric energy $\mathcal{E}_e(t)$ decreases, nonlinear effects then starts to play a role and the amplitude of $\mathcal{E}_e(t)$ increases. An oscillatoring behaviour is then observed (see [11,20]) for the amplitude of the electric energy.

We are then interested in the time history of the electric energy $\mathcal{E}_e(t)$, but also in the L^2 norm of f and the total energy $\mathcal{E}(t)$, the definition of which are given by (3.7) and (3.8). Let us note that all the methods preserve the L^1 norm except SPL (which does not include appropriated filter).

On Fig. 2, the different numerical schemes are then compared with respect to these latter quantities. First, we can observe that all the methods present a good behaviour regarding the time evolution of the electric energy, compared to the numerical results available in the literature (see [11,20]). Some differences appear on the behaviour of the L^2 norm and the total energy. Indeed, the use of local filter (Umeda filter) introduces some additional diffusion which kills the spurious oscillations and consequently makes the L^2 norm decreasing. We can observe on the middle column of Fig. 2 that the Umeda filter makes the different methods (LAG, PPM or PSM) decrease the L^2 norm more rapidly than the global filter. The other influence is clear when one looks at the time history of the total energy: the behaviour is nearly the same for most of the methods in the linear phase (slight increase of $\mathcal{E}(t)$), but when nonlinear effects become significant (since $t \approx 30\omega_p^{-1}$), the local filter acts more often than the global one, which leads to an increase of the total energy. Similar remarks have been performed in [8].

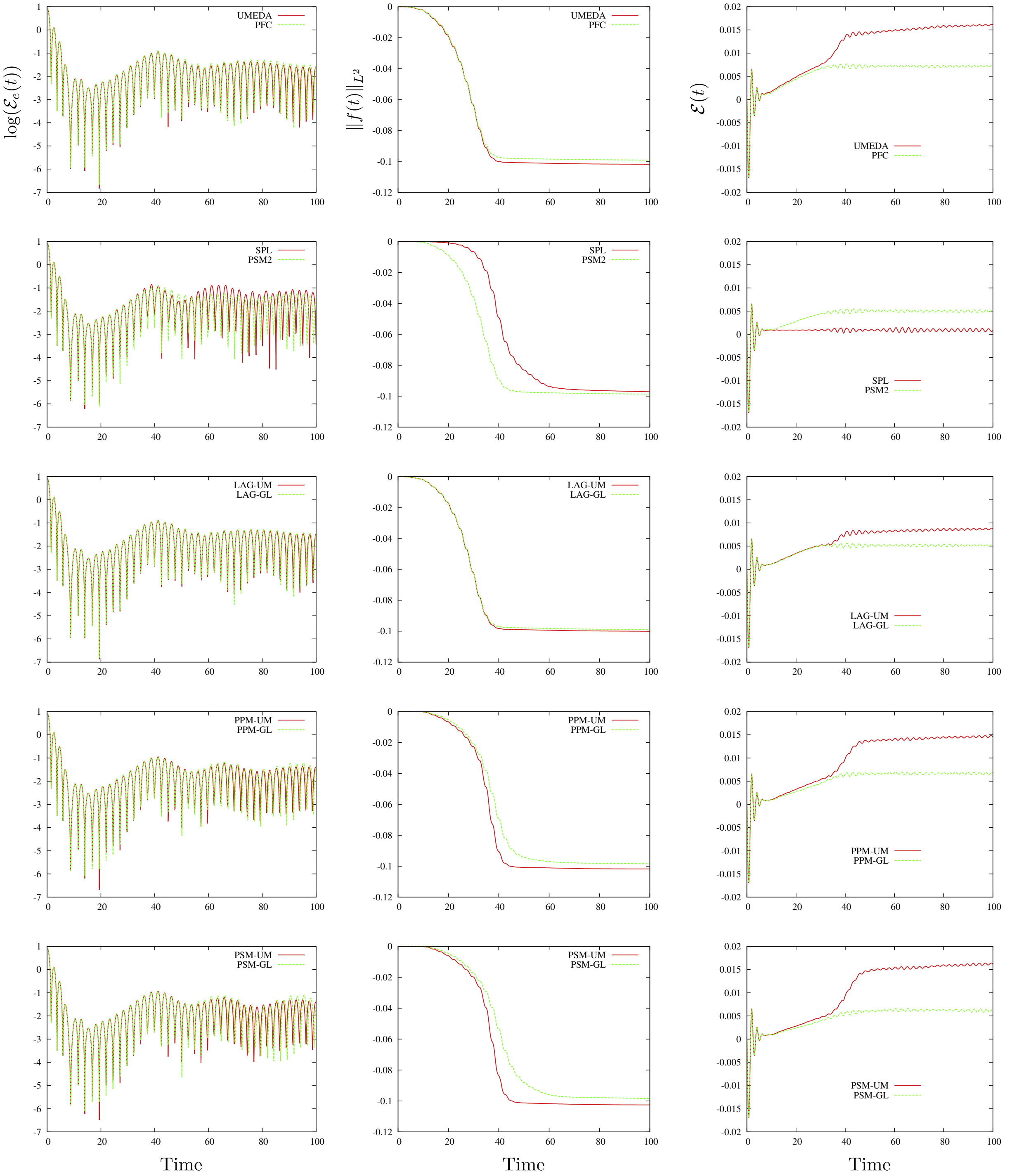


Fig. 2. Numerical results for the strong Landau damping. Left: time history of the electric energy (semi-log scale); middle: time history of the L^2 norm; right: time history of the total energy.

From this point of view, PSM2 has a correct behaviour since it preserves the total energy up to 0.5%. As expected, the behaviour of its L^2 norm is intermediate between PSM (or PPM) and Lagrange based methods.

3.2.2.3. Bump-on-tail test case. The numerical schemes are now validated on a test case introduced in [23], and numerical results are available in [11,12,22]. In the present work, numerical results obtained by the methods of Section 2 are applied on the bump-on-tail instability test case for which the initial condition writes

$$f_0(x,v) = \tilde{f}(v)[1 + 0.04\cos(kx)], \quad x \in [0, L], v \in [-v_{\max}, v_{\max}],$$

with $k = 0.5$, $v_{\max} = 9, L = 20\pi$; moreover, we have

$$\tilde{f}(v) = n_p \exp(-v^2/2) + n_b \exp\left(-\frac{|v-u|^2}{2v_t^2}\right)$$

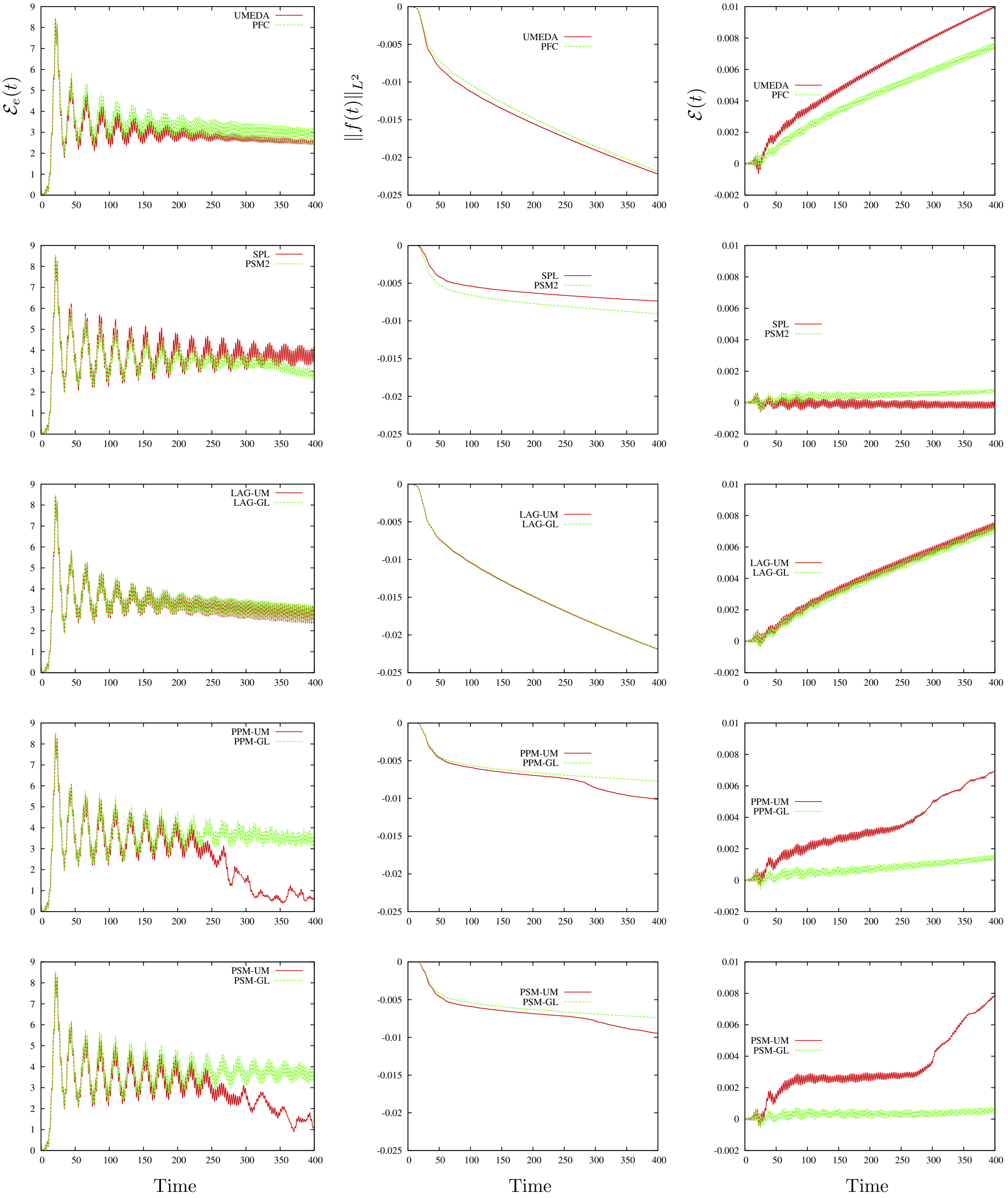


Fig. 3. Numerical results for the bump-on-tail test case. Left: Time history of the electric energy; middle: Time history of the L^2 norm; right: Time history of the total energy.

whose parameters are

$$n_p = \frac{9}{10(2\pi)^{1/2}}, \quad n_b = \frac{2}{10(2\pi)^{1/2}}, \quad u = 4.5, \quad v_t = 0.5.$$

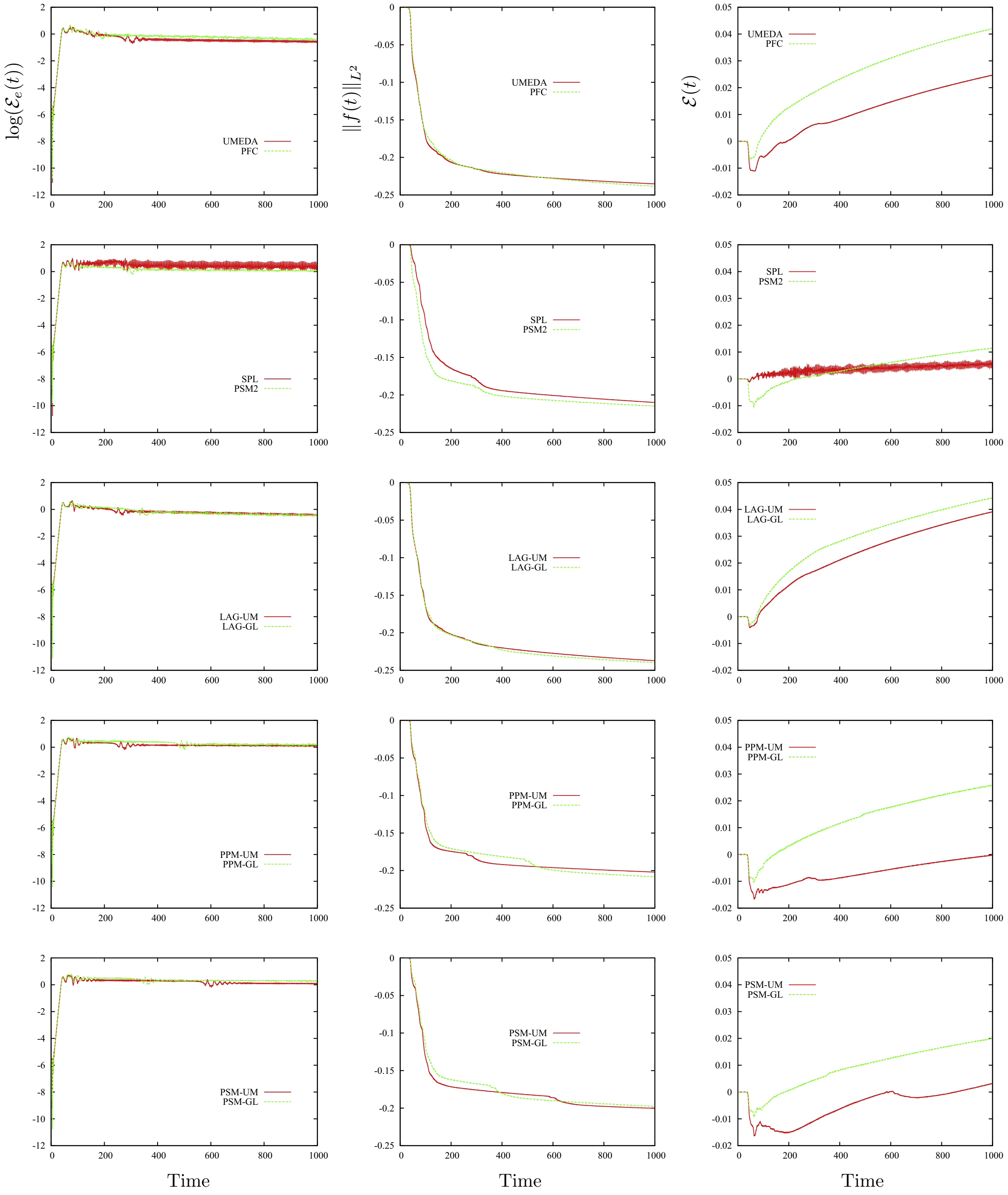


Fig. 4. Numerical results for the two stream instability test case. Left: Time history of the electric energy (semi-log scale); middle: Time history of the L^2 norm; right: Time history of the total energy.

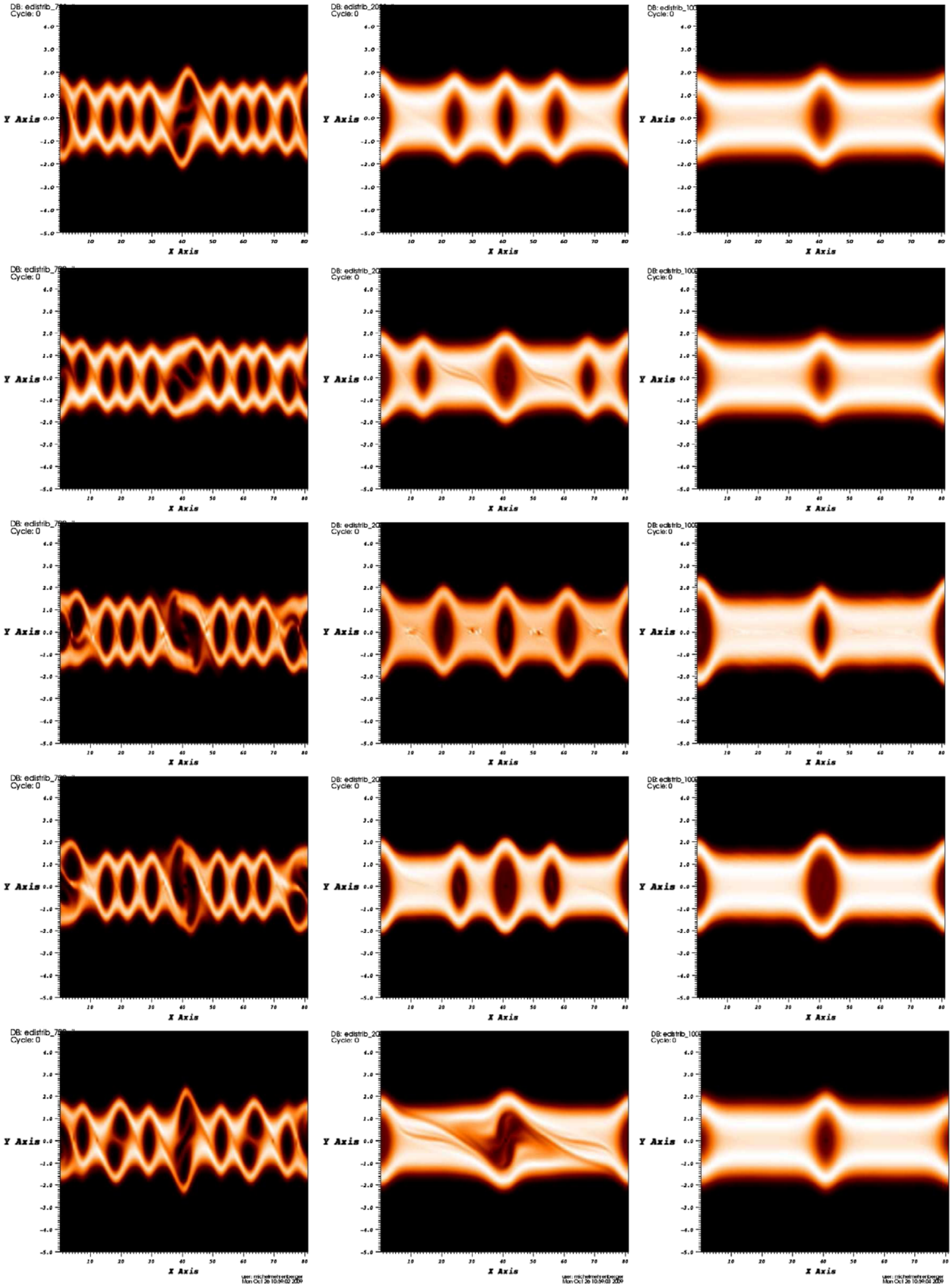


Fig. 5. Distribution function for different times for the two stream instability test case. Left: $t = 70\ \omega_p^{-1}$; middle: $t = 200\ \omega_p^{-1}$; right: $t = 1000\ \omega_p^{-1}$. From top to bottom: LAG-GL, LAG-UM, PPM-GL, PPM-UM, PFC.

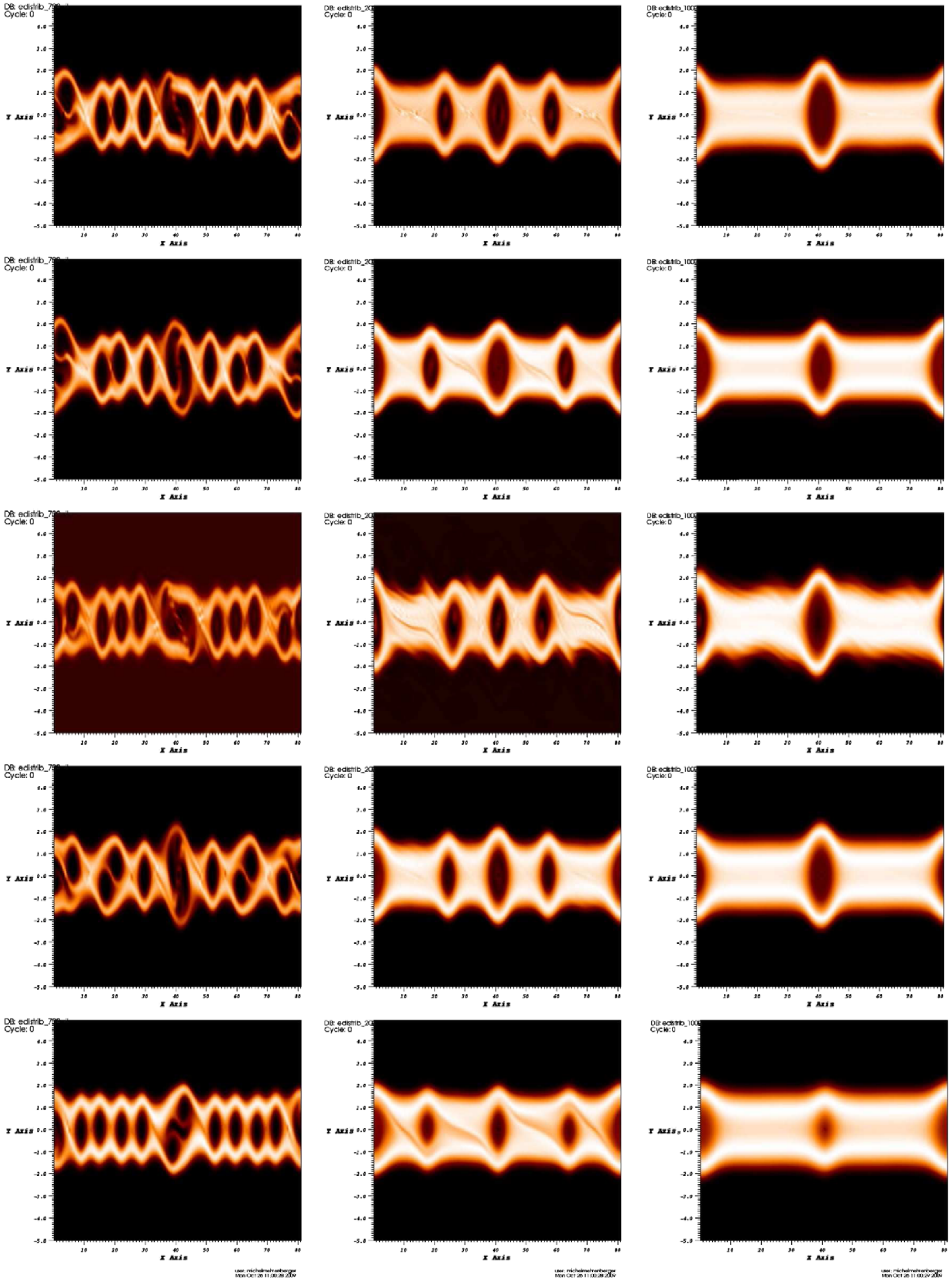


Fig. 6. Distribution function for different times for the two stream instability test case. Left: $t = 70\ \omega_p^{-1}$; middle: $t = 200\ \omega_p^{-1}$; right: $t = 1000\ \omega_p^{-1}$. From top to bottom: PSM-GL, PSM-UM, SPL, PSM2, UMEDA.

The numerical parameters are $N_x = 128, N_v = 128, \Delta t = 0.1$ (so that the Courant number is 1.8) and the number of iterations is 4000.

For this test case, we are interested in the time evolution of the total energy $\mathcal{E}$ together with the electric energy $\mathcal{E}_e$ given by (3.8). We also look after the L^2 norm of f (see (3.7)) which are conserved with time. The same is true for the total energy whereas the electric energy is expected to present an oscillatoring behaviour for large times (see [22]). In this test, three vortices are created in the phase space which are moving along the velocity $v = v_t$ (BGK equilibrium, see [23]) and which can merge. As a consequence, a loss of the oscillatoring behaviour of the electric energy is observed. Our goal is to compare the different methods, as in the previous test.

Fig. 3, left column shows the time evolution of the electric energy for the different methods. The main features (see [23,22]) of the expected behaviour are respected by all the methods: the electric energy presents a maximum at $t \approx 20\omega_p^{-1}$ and then an slowing oscillatoring behaviour on which is superimposed the oscillation of the system at ω_p. Nevertheless, different classes can be distinguished for larger times: Lagrange based methods strongly damp the slow oscillations whereas PPM-UM and PSM-UM present a break due to the merging of two vortices in the phase space. For the methods based on Lagrange interpolation (LAG, PFC, UMEDA), the oscillations of the electric energy due to the particles trapping are damped and the amplitude is decreasing for large time. It is less the case for the splines based methods (SPL, PSM-GL, PSM2) which keep the slow oscillatoring behaviour around a constant amplitude up to the end of the simulation. This can be explained by the fact that fine structures are developed in the vortices; they are quickly eliminated by the methods based on Lagrange interpolation whereas splines methods follow these thin details of the phase space solution for longer times.

These observations are emphasized by Fig. 3, middle and right columns. The L^2 norm of Lagrange based methods has a strong decrease compared to other ones. We can observe a break in the time evolution of the L^2 norm for PPM-UM and PSM-UM which corresponds to the vortices merging and to an associated dissipation of fine structures. This phenomenum appears later for methods on which the global filter is applied or for the PSM2 approach. PSM2 approach is not so diffusive and presents a good behaviour since the vortices merging occurs a larger times compared to other local filter.

Finally, on Fig. 3 (right column), the time evolution of the total energy is plotted for the different methods. This quantity is quite difficult to preserve at the discrete level (see [8,29,4]). In particular, it is very difficult to ensure both positivity of the solution and conservation of the total energy for nonlinear tests. Let us recall that all the methods preserve the positivity except SPL. We can observe the influence of the slope limiters on the results: indeed, the Lagrange based methods (PFC, UMEDA and LAG) present roughly the same behaviour. This figure also emphasizes the good behaviour of PSM2 since the conservation of the total energy is about 0.1%, which is similar to the PSM-GL or PPM-GL methods on which global filter is applied.

3.2.2.4. Two stream instability test case. The initial condition associated to the Vlasov–Poisson model is taken from [30]

$$f(x,v,t=0) = \frac{1}{2v_{th}\sqrt{2\pi}}\left[\exp(-(v-u)^2/(2v_{th}^2) + \exp(-(v+u)^2/(2v_{th}^2)\right](1+0.05\cos(kx)),$$

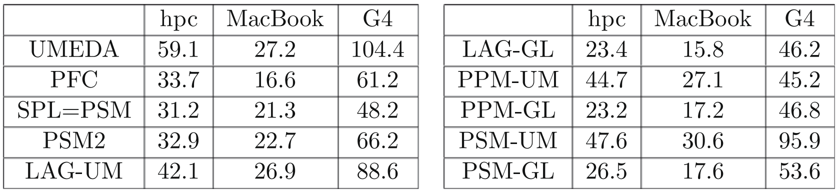
where $x \in [0, 26\pi]$, $v \in [-5, 5]$, $v_{th} = 0.3, u = 0.99, k = 0.5$. The numerical parameters are $N_x = N_v = 128$, and $\Delta t = 0.1$ (so that the Courant number is 0.78). The simulations are stopped at $t = 1000\omega_p^{-1}$.

As in the previous cases, we look at the conservation of the preserved quantities: the L^2 norm of f and the total energy. We are also interested in the time history of the electric energy (in semi-log scale) to distinguish the linear phase from the nonlinear one.

Since the initial datum is a perturbed unstable equilibrium, we expect an instability to start at the beginning. Then, after saturation, trapped particles oscillate in the electric field until the end of the simulation.

Numerical results are shown on Fig. 4. The first remarks concern the good behaviour of the local filter compared to the global one. Indeed, for all the reconstruction, it improves the total energy conservation compared to the global one, without degenerating the L^2 norm conservation. We can notice for this latter quantity that it presents some small jumps which correspond to the merging of vortices (the initial distribution function presents 13 modes initially so that up to 13 vortices are created). Hence, filters kill spurious oscillations and in some sense provide from the merging of vortices which can also be accelerated by the oscillations. In this test, PSM2 also presents a good behaviour, in particular regarding the total energy conservation even if the Umeda filter also gives rise to good results when it is applied to PPM or PSM. We plot on Figs. 5 and 6 the distribution function for the different methods at different times of the simulations. We first remark that all the methods lead to 4 vortices at $t = 200\omega_p^{-1}$ (except PFC) and to 2 vortices at the end of the simulation ($t = 1000\omega_p^{-1}$). Some differences can however be detected. For example, SPL (Fig. 6, third line) presents spurious oscillations at $t = 200\omega_p^{-1}$ and $t = 1000\omega_p^{-1}$. Note that SPL also creates negative values for the distribution function. We can also observe the numerical diffusion on the final centered vortex; this corresponds to the decay of the L^2 norm for Lagrange based methods whereas a saturation is observed for the other methods.

We finally show on Fig. 7 the CPU time of the different methods for the present numerical test. This CPU time is measured in seconds for 3 different machines: hpc (SUN X4600 Processor Opteron Core Duo 2.8 GHz and memory 64Go), MacBook



	hpc	MacBook	G4
UMEDA	59.1	27.2	104.4
PFC	33.7	16.6	61.2
SPL=PSM	31.2	21.3	48.2
PSM2	32.9	22.7	66.2
LAG-UM	42.1	26.9	88.6

	hpc	MacBook	G4
LAG-GL	23.4	15.8	46.2
PPM-UM	44.7	27.1	45.2
PPM-GL	23.2	17.2	46.8
PSM-UM	47.6	30.6	95.9
PSM-GL	26.5	17.6	53.6

Fig. 7. Comparison of computational cost for the two stream instability test case. $N_x = N_v = 128$, 1000 iterations.

(Processor 2.4 GHz Intel Core 2 Duo and memory 2 GB 1067 MHz DDR3) and G4 (processor 1.67 GHz PowerPC G4 and memory 1 GB DDR2). In the simulations the 2D diagnostics are not included.

4. Non-constant case: The guiding-center model

In this work, we also deal with another type of Vlasov equation for which the advection term is not constant. The so-called guiding-center model enters in this category (see [26]). This model, which has been derived to describe highly magnetized plasma in the transverse plane of a tokamak, considers the evolution of the particles density $\rho(t,x,y)$

$$\partial_t \rho + E^{\perp} \cdot \nabla \rho = 0, \tag{4.1}$$

where the electric field

$$E = E(x,y) = (E_x(x,y), \quad E_y(x,y)),$$

satisfies a Poisson equation

$$-\Delta \Phi = \rho, \quad E = -\nabla \Phi. \tag{4.2}$$

We denote by $E^{\perp} = (E_y, -E_x)$. The specificity of (4.1) lies on the fact that one-dimensional splitting cannot (in principle) be applied (see [26,17]) since the advection term $E^{\perp}$ depends on (x,y). Consequently, this model contains additional difficulties compared to the Vlasov–Poisson model and seems to be a good candidate to test numerical methods.

To that purpose, we briefly recall the conservation properties of (4.1) which should be preserved in the best manner by the numerical schemes. The guiding-center model (4.1) preserves the total mass, the L^2 norm of the density (enstrophy) and the L^2 norm of the electric field (energy)

$$\frac{d}{dt} \int \int \rho(t,x,y) dx dy = \frac{d}{dt} \|\rho(t)\|_{L^2} = \frac{d}{dt} \|E(t)\|_{L^2} = 0. \tag{4.3}$$

4.1. The general algorithm

In this subsection, we review the main steps of a semi-Lagrangian method in the case of directional splitting which is applied for the discretization of the guiding-center-Poisson model.

4.1.1. Grid notations

Let $N_x, N_y \in \mathbb{N}^*, y_{\max} > 0$, a time step $\Delta t > 0$.

We define then classically as notations

$$\Delta x = L_x/N_x, \quad \Delta y = L_y/N_y \quad x_k = kL_x/N_x, \quad y_\ell = \ell L_y/N_y$$

for $k = 0, \ldots, N_x,\ \ell = 0, \ldots, N_y$ and $t^n = n\Delta t,\ n \in \mathbb{N}$.

4.1.2. Discretization of the distribution function

The unknown quantities are then $\rho^n_{k,\ell}$ which are approximations of $\rho(t^n, x_k, y_\ell)$. We suppose periodic boundary conditions so that we only have to compute at each time t^n

$$\rho^n_{k,\ell}, \text{ for } k = 0, \ldots, N_x - 1, \quad \ell = 0, \ldots, N_y - 1.$$

4.1.2.1. Transport operator. Let us define for $(\alpha_k) \in \mathbb{R}^{N_x+1}$ a transport operator $\mathcal{T}_\alpha : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$. For the conservative approaches we detailed in Section 2.1, this operator writes

$$\mathcal{T}_\alpha(\bar{\rho}_0, \bar{\rho}_1, \ldots, \bar{\rho}_{N_x-1}) = \left(\frac{1}{\Delta x} \int_{x_{k-1/2}-\alpha_{k-1/2}}^{x_{k+1/2}-\alpha_{k+1/2}} \rho(x) dx \right)_{k=0,\ldots,N-1}.$$

The sequence α is determined following one of the algorithms detailed in Sections 2.1.1 and 2.2.1.

Algorithm 1.

```
Step 0. Initialization:$\rho_{k,\ell} = \rho_0(x_k, y_\ell), k = 0,\ldots,N_x - 1, \ell = 0,\ldots,N_y - 1$.
Step 1. Compute of the electric field $\left(E_x^0, E_y^0\right)$ by integrating (4.2).
Step 2. Compute $\rho_{k,\ell}^1$ using $\rho^0$:
Step 2a. Half time step shift along the x-axis:
Compute the x-displacement for each $\ell \alpha_k = \Delta t/4E_y^0(x_k - \alpha_k, y_\ell)$ For each $\ell = 0,\ldots,N_y - 1, (\rho_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}_\alpha^x\left((\rho_{k,\ell})_{k=0}^{N_x-1}\right)$.
Step 2b. Shift along the y-axis:
Compute the y-displacement for each $k\alpha_\ell = -\Delta t/2E_x^0(x_k, y_\ell - \alpha_\ell)$ For each $k = 0,\ldots,N_x - 1, (\rho_{k,\ell})_{\ell=0}^{N_y-1} \to \mathcal{T}_\alpha^y\left((\rho_{k,\ell})_{\ell=0}^{N_y-1}\right)$.
Step 2c. Half time step shift along the x-axis:
Compute the x-displacement for each $k\alpha_k = \Delta t/4E_y^0(x_k - \alpha_k, y_\ell)$ For each $\ell = 0,\ldots,N_y - 1, (\rho_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}_\alpha^x\left((\rho_{k,\ell})_{k=0}^{N_x-1}\right)$.
Step 3. Compute the electric field $\left(E_x^1, E_y^1\right)$ by integrating (4.2).
Step 4. Compute $\rho_{k,\ell}^{n+1}$ using $\rho^{n-1}, \rho^n$:
Step 4a. Half time step shift along the x-axis:
Compute the x-displacement for each $\ell \alpha_k = \Delta t/2E_y^n(x_k - \alpha_k, y_\ell)$
For each $\ell = 0,\ldots,N_y - 1, (\rho_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}_\alpha^x\left((\rho_{k,\ell})_{k=0}^{N_x-1}\right)$.
Step 4b. Shift along the y-axis: Compute the y-displacement for each $k\alpha_\ell = -\Delta t E_x^n(x_k, y_\ell - \alpha_\ell)$ For each $k = 0,\ldots,N_x - 1, (\rho_{k,\ell})_{\ell=0}^{N_y-1} \to \mathcal{T}_\alpha^y\left((\rho_{k,\ell})_{\ell=0}^{N_y-1}\right)$.
Step 4c. Half time step shift along the x-axis:
Compute the x-displacement for each $\ell \alpha_k = \Delta t/2E_y^n(x_k - \alpha_k, y_\ell)$
For each $\ell = 0,\ldots,N_y - 1, (\rho_{k,\ell})_{k=0}^{N_x-1} \to \mathcal{T}_\alpha^x\left((\rho_{k,\ell})_{k=0}^{N_x-1}\right)$.
Step 5. Compute the electric field $\left(E_x^{n+1}, E_y^{n+1}\right)$ by integrating (4.2).
Step 6. $n \to n + 1$ and loop to Step 4.
```

Different methods will be compared. We consider the new methods PSM (i.e. without filter), PSM2, and the forward update method (FUM) (detailed in Section 2.2 with cubic spline for the reconstruction step). We consider also the traditional advective semi-Lagrangian method without splitting SPL2D with full two-dimensional cubic splines interpolation developed in [26] and the splitting method using the one-dimensional cubic splines advective approach SPL1D. The characteristics are solved using the midpoint formula for backward methods and the RK4 algorithm is used for FUM.

4.2. Numerical results: Kelvin–Helmholtz instability test case

We consider the Kelvin–Helmholtz instability in the periodic–periodic case (i.e. periodic boundary conditions are considered in the x and y direction) for which the growth rate of the instability can be computed *a priori*. This is of great importance to check quantitatively the accuracy of the code.

Following the computations of [24], the linearization (4.1) and (4.2) leads to the so-called stability Rayleigh equation. Considering as initial condition a periodic perturbation of the equilibrium solution to (4.1) and (4.2), it is possible to start a Kelvin–Helmholtz instability. The difference between the Dirichlet-periodic case, i.e. periodic boundary conditions in the x direction and Dirichlet ones in the y direction, (which has been solved in [26]) occurs in the neutrally stable solution which is equal to 1 in our case (instead of $\sin(y/2)$ in the Dirichlet-periodic case). Then, we can deduce the initial condition for (4.1) and (4.2),

$$\rho(x, y, t = 0) = \sin(y) + \varepsilon \cos(kx),$$

where $k = 2\pi/L_x$ is the wave number associated to the length L_x of the domain in the x-direction. The size of the domain in the y-direction is $L_y = 2\pi$. Shoucri's analysis predicts an instability when k is chosen lower than 1. Otherwise, the initial perturbation remains unchanged, neither damped (since (4.1) and (4.2) is only a fluid model, not a kinetic model), neither increased.

Various approaches can be employed to determine the instability growth rate of the chosen mode k. A finite difference numerical scheme has been applied to approximate the stability Rayleigh equation which leads to a eigenvalue problem. The results obtained by this way are very closed to those obtained by numerically solving the linearized problem as performed in [24].

The numerical parameters are chosen as follows:

$$k = 0.5, N_x = N_y = 128, \quad \text{and } \Delta t = 0.1.$$

We have checked that the maximal displacement in the x direction is about 1 cell, whereas the maximal displacement in the y direction increases during the linear phase up to 5 cells and remains between 4 and 5 in the non linear phase. In particular, we remark that the time step is not restricted by the classical CFL condition, which imposes that the maximal displacement is lower than one cell. Let us recall that periodic conditions are considered here; even if the present test bears similarities with the Dirichlet-periodic test presented in [26], the dynamics of the unknown is quite different in the present periodic–periodic context.

For this test case, we are interested in the time evolution of the conserved quantities (4.3). We also look carefully at the conservation of the total mass in order to verify the difference between conservative and non-conservative methods. As a diagnostic, it is also interesting to look after the 2D unknown to realize the fine structures developed along the simulation.

In Fig. 8, the time histories of the total mass and the L^2 norm of the density is plotted for the different methods. First, as discussed in [17,21,28], SPL1D does not preserve exactly the total mass whereas other methods do. This is expected since this approach solves the non-conservative form of the equation which is not appropriate with the splitting procedure. Then, we observe that the conservative methods present very similar behaviour compared to the method of reference SPL2D: they are conservative and the decay of the L^2 norm occurs at $t \approx 30\,\omega_p^{-1}$. This decay corresponds to the saturation of the instability. Very fine structures are created which cannot be captured by the numerical schemes since their size becomes smaller than the grid size.

In Fig. 9, the logarithm of the first Fourier mode of the electric field E_x is plotted as a function of time. The linear theory predicts an exponential growing, the rate of which can be computed *a priori* by solving an eigenvalue problem. This can be performed and the results can be compared to the numerical results. The numerical growth rate corresponds to the slope of the straight line which approximates the logarithm of the first Fourier mode of E_x in the linear phase (between $t \approx 5\,\omega_p^{-1}$ and $t \approx 10\,\omega_p^{-1}$). Considering different values of the wave number k, it is possible to plot the quantity ω/k (where ω is the growth rate of the first Fourier mode of E_x) as a function of $(k_s - k)$ where $k_s = 1$ in our case ($k_s = \sqrt{3}/2$ in the Dirichlet-periodic case). This is performed in Fig. 10 (right); we can observe the very good agreement between the analytical and the numerical values. This kind of validation is of great importance since a quantitative comparison can be performed, at least in the linear phase.

On Fig. 10 (left), the L^2 norm of the electric field is plotted as a function of time. This quantity is preserved with time by the continuous model. The conservative and splitting procedure based methods present a very good conservation of the energy whereas it is not the case of the non-conservative method SPL1D. The method SPL2D does not preserve very well the energy compared to FUM or PSM for example. We can observe that the influence of the filter is very weak since the numerical results of PSM and PSM2 are very close (see Fig. 11). However, we can observe on the distribution function that some

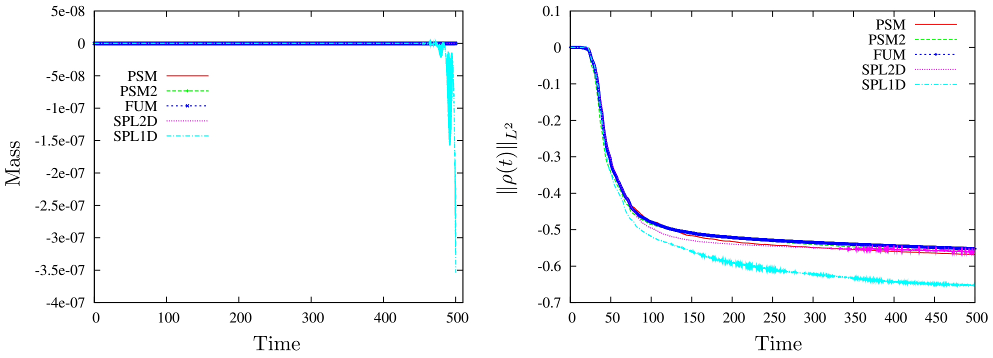


Fig. 8. Time evolution of the total mass and the enstrophy for SPL (with splitting (SPL1D) and without splitting (SPL2D)), FUM, PSM and PSM2. $N_x = N_y = 128, \Delta t = 0.1$ for the Kelvin–Helmholtz instability test.

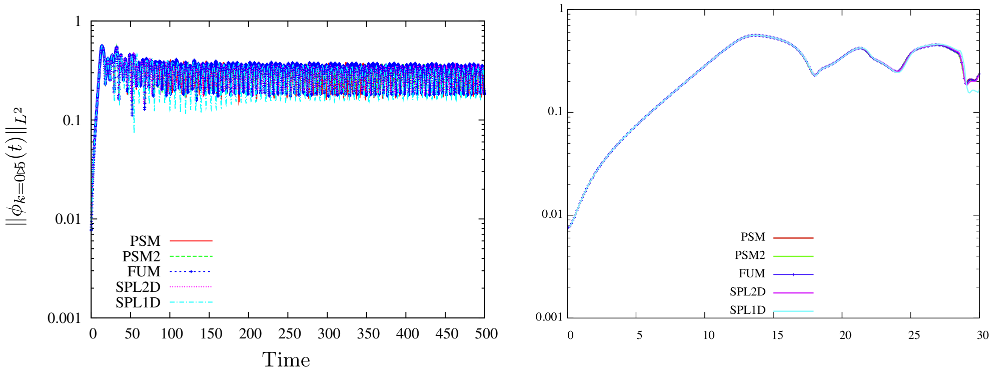


Fig. 9. Time evolution of the logarithm of the first Fourier mode for SPL (with splitting (SPL1D) and without splitting (SPL2D)), FUM, PSM and PSM2. A zoom has been applied on the right figure. $N_x = N_y = 128, \Delta t = 0.1$ for the Kelvin–Helmholtz instability test.

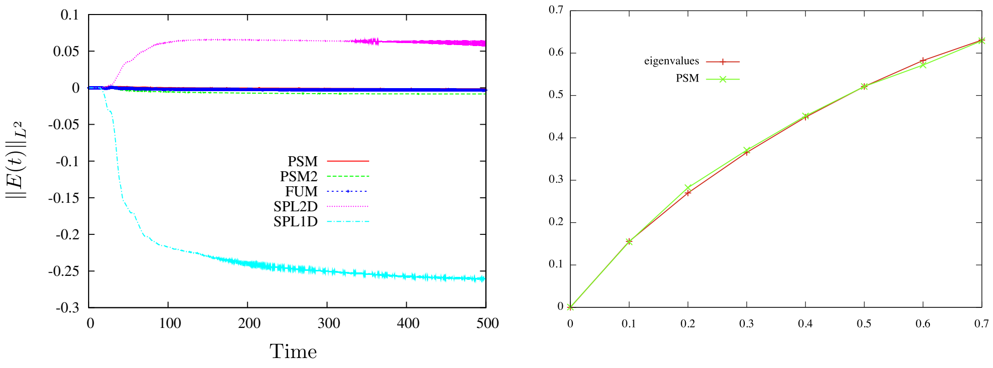


Fig. 10. Left figure: time evolution of the energy for SPL (with splitting (SPL1D) and without splitting (SPL2D)), FUM PSM and PSM2. Right figure: normalized growth rate ω/k as a function of $1-k$. $N_x = N_y = 128, \Delta t = 0.1$ for the Kelvin–Helmholtz instability test.

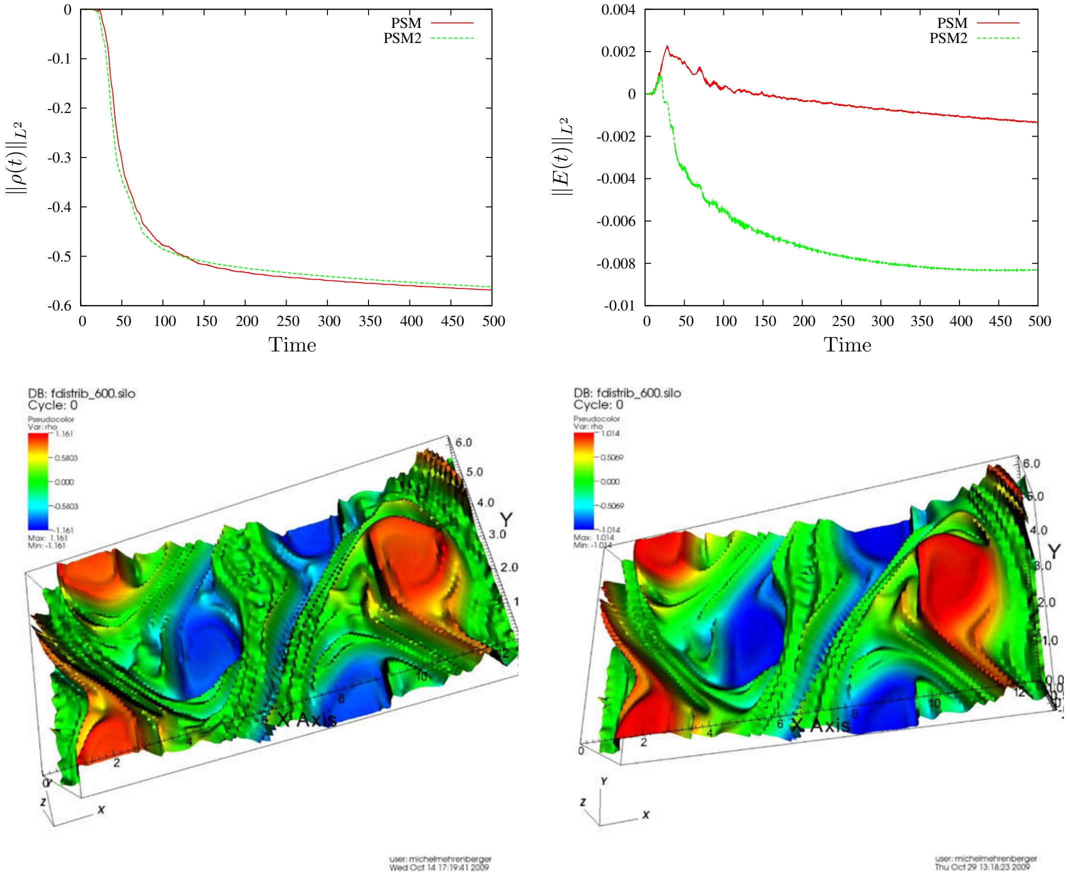


Fig. 11. Kelvin–Helmholtz instability test: comparison of the PSM and PSM2 method; time history of the L^2 norm of the distribution function and the electric field and distribution functions at time $t = 60\omega_p^{-1}$ (left: PSM, right: PSM2). $N_x = N_y = 128, \Delta t = 0.1$.

oscillations are suppressed by the filter, without affecting too much the L^2 norm. We have observed that the extrema are better conserved for PSM2 compared to PSM, even if it is not strictly respected.

Finally, on Figs. 12and 13 we plot the distribution function for the different methods at time $t = 30$ and $t = 60\omega_p^{-1}$. We add here the numerical result associated to the conservative splitting with LAG reconstruction (that is the PFC method without filter). These results confirm the previous observations: first, the LAG method is more diffusive (the thin structures are smoothed) and the SPL1D scheme leads to a bad behaviour since the main structures are not respected. In contrast, the FUM and PSM present a good behaviour, very similar to SPL2D. Note that we have some overshoots for the refined SPL2D solution (lower right subplot), which the difference of color scales.

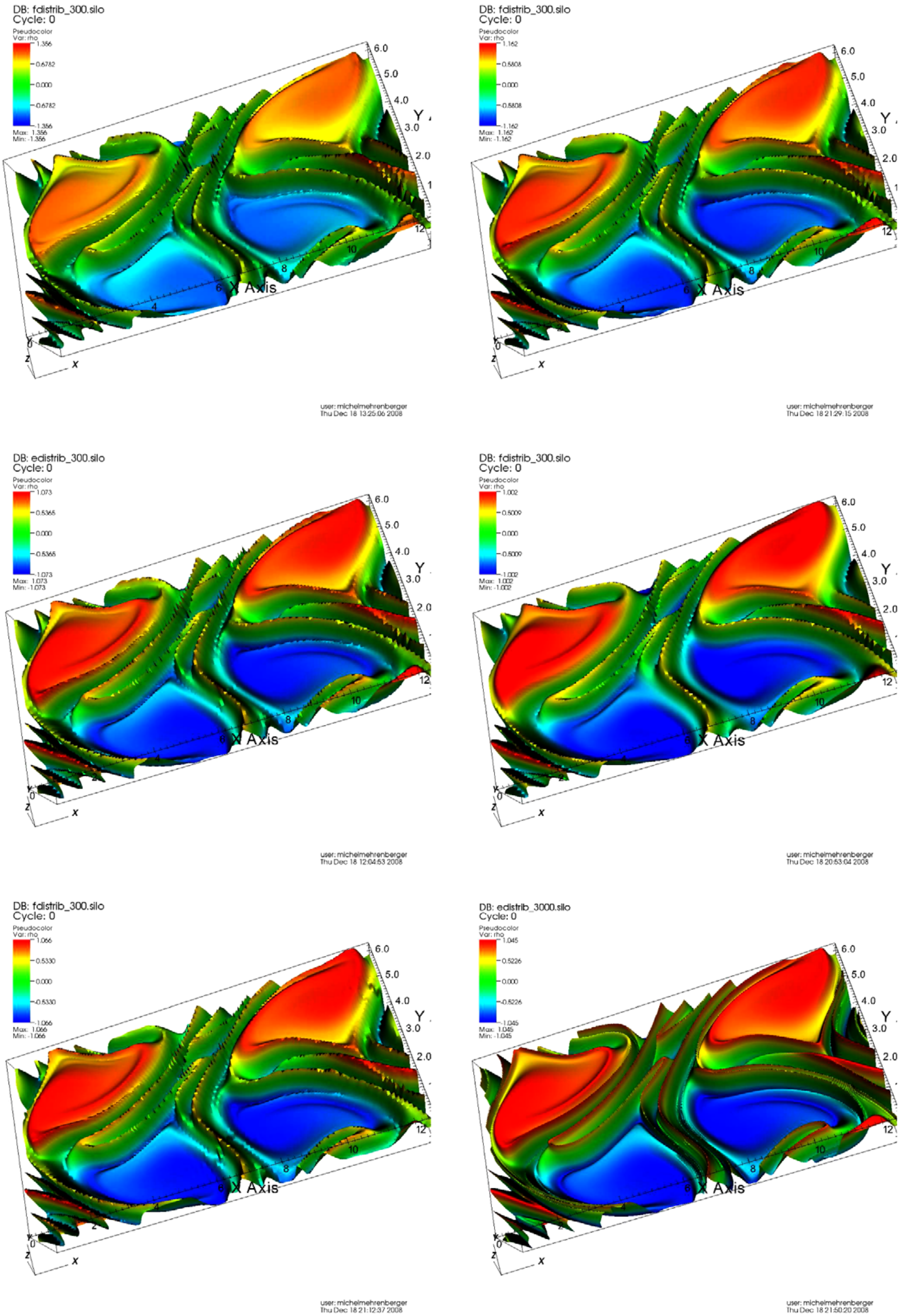


Fig. 12. Distribution function for the Kelvin–Helmholtz instability test at time $t = 30\omega_p^{-1}$. Respectively for top-left to bottom-right: PSM, FUM, SPL2D, LAG, SPL1D with $N_x = N_y = 128, \Delta t = 0.1$ and SPL2D with $N_x = N_y = 512, \Delta t = 0.01$.

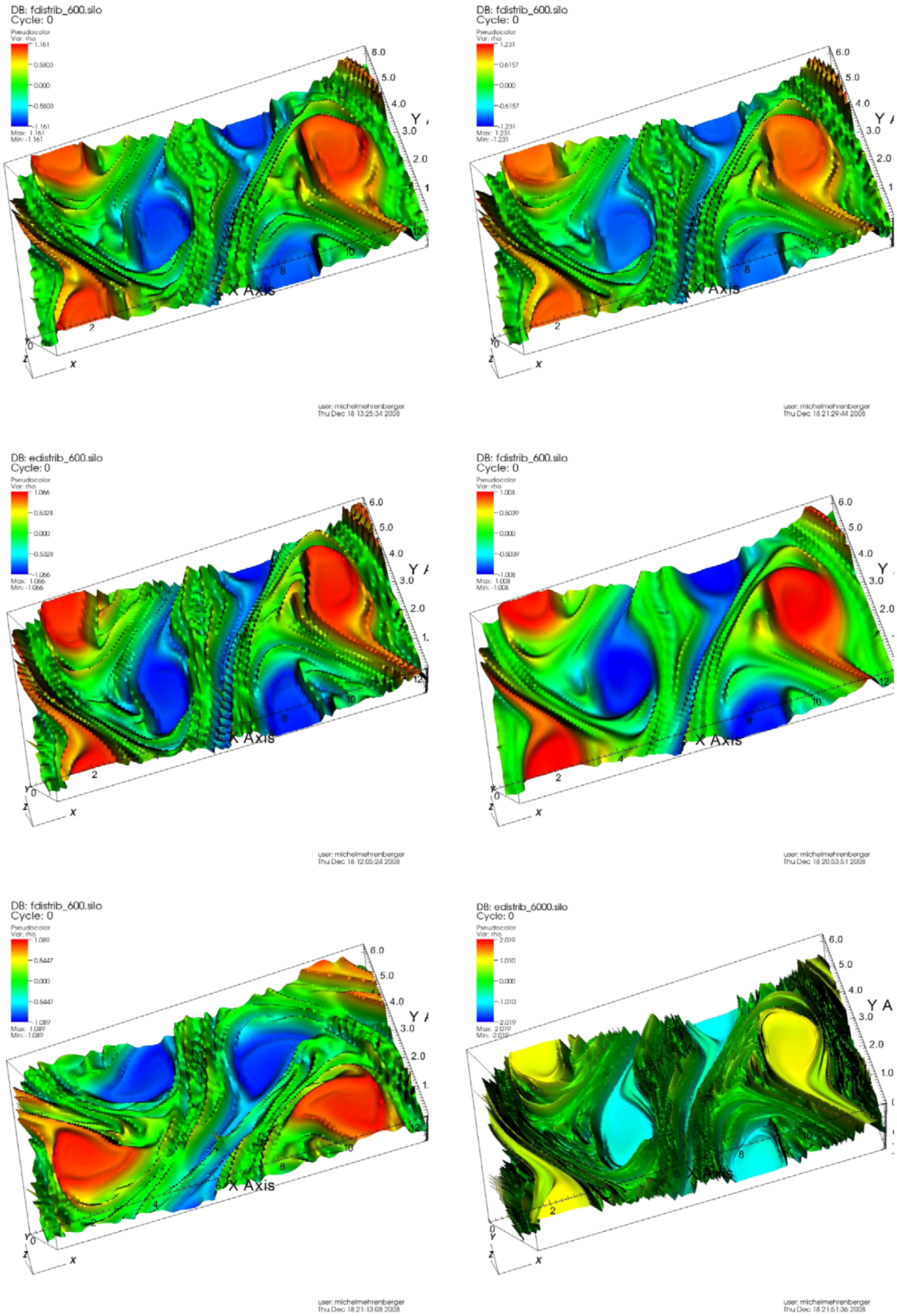


Fig. 13. Distribution function for the Kelvin–Helmholtz instability test at time $t = 60\omega_p^{-1}$. Respectively for top-left to bottom-right: PSM, FUM, SPL2D, PFC, SPL1D with $N_x = N_y = 128, \Delta t = 0.1$ and SPL2D with $N_x = N_y = 512, \Delta t = 0.01$.

As a conclusion, the conservative methods present a very good behaviour on this strongly nonlinear and large time case. The splitting procedure also enables to save memory since one-dimensional structures are often used (instead of two-dimensional structures for the computation of the cubic spline coefficients in SPL2D for example). On the other side, we remarked that SPL2D seems to be able to support larger time steps, but we think that the implementation of high order numerical scheme in time could stabilize PSM or FUM when large time steps are used. This extension will be studied in a future work.

5. Conclusion

In this work, new conservative methods have been introduced and then compared to existing methods for equations occuring in plasma physics. Several properties make them very competitive. On the one side, their inherent conservation property enables the use of splitting procedures, which make the implementation of multi-dimensional problems easier. On the other side, slope limiters can be introduced to ensure the positivity or to control spurious oscillations of the unknown.

When they are compared to existing semi-Lagrangian methods, we first observed that for the guiding-center problem, as expected, the advective approaches lead to inaccurate results when splitting procedure is applied. This is not the case for conservative methods. Moreover, they are at least as accurate as the reference methods (SPL2D). For simpler cases like the Vlasov–Poisson model in which the advection term is constant, we proved that the advective methods and their conservative counterparts are equivalent. Obviously, this does not remain true in the non-constant advection case (like for the guiding-center model) which often occurs in particular in gyrokinetic models. The extension of the PSM method to such multi-dimensional system is currently investigated.

Moreover, the splitting procedure makes easier the use of high order numerical scheme in time for backward and forward approaches. The use of high order time splittings (see [31]) is also under investigations.

References

[1] N. Besse, M. Mehrenberger, Convergence of classes of high order semi-Lagrangian schemes for the Vlasov–Poisson system, Math. Comput. 77 (2008) 93–123.
[2] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, Inst. of Phys. Publishing, Bristol/Philadelphia, 1991.
[3] M. Brunetti, X. Lapillonne, S. Brunner, T.M. Tran, Comparison of semi-Lagrangian and Eulerian algorithms for solving Vlasov-type equations, colloque numérique suisse, EPFL, avril 12, 2008.
[4] J.-A. Carillo, F. Vecil, Non-oscillatory interpolation methods applied to Vlasov-based models, SIAM J. Sci. Comput. 29 (2007) 1179–1206.
[5] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys. 22 (1976) 330–3351.
[6] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.
[7] P. Colella, M.D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema, J. Comput. Phys. 227 (2008) 7069–7076.
[8] N. Crouseilles, F. Filbet, Numerical approximation of collisional plasmas by high order methods, J. Comput. Phys. 201 (2004) 546–572.
[9] N. Crouseilles, T. Respaud, E. Sonnendrücker, A forward semi-Lagrangian scheme for the numerical solution of the Vlasov equation, Comput. Phys. Commun. 180 (2009) 1730–1745.
[10] C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978.
[11] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, J. Comput. Phys. 172 (2001) 166–187.
[12] F. Filbet, E. Sonnendrücker, Comparison of Eulerian Vlasov solvers, Comput. Phys. Commun. 151 (2003) 247–266.
[13] A. Ghizzo, P. Bertrand, M.L. Begue, T.W. Johnston, M. Shoucri, A Hilbert–Vlasov code for the study of high-frequency plasma beatwave accelerator, IEEE Trans. Plasma Sci. 24 (1996) 70.
[14] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarrazin, O. Sauter, E. Sonnendrücker, J. Vaclavik, L. Villard, A drift-kinetic semi-Lagrangian 4D code for ion turbulence simulation, J. Comput. Phys. 217 (2006) 395–423.
[15] F. Coquel, Ph. Helluy, J. Schneider, Second order entropy diminishing scheme for the Euler equations, Int. J. Numer. Meth. Fluids 50 (2006) 1029–1061.
[16] James M. Hyman, Accurate monotonicity preserving cubic interpolation, SIAM J. Sci. Stat. Comput. 4, 645–654.
[17] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrücker, O. Coulaud, Instability of the time splitting scheme for the one-dimensional and relativistic Vlasov–Maxwell system, J. Comput. Phys. 185 (2003) 512–531.
[18] J. Laprise, A. Plante, A class of semi-Lagrangian integrated-mass (SLIM) numerical transport algorithms, Mon. Wea. Rev. 123 (1995) 553–565.
[19] P.H. Lauritzen, An inherently mass-conservative semi-implicit semi-Lagrangian model, Ph.D. thesis, Department of Geophysics, University of Copenhagen, Denmark, September, 2005.
[20] G. Manfredi, Long-time behavior of nonlinear Landau damping, Phys. Rev. Lett. 79 (1997) 2815–2818.
[21] T. Nakamura, R. Tanaka, T. Yabe, K. Takizawa, Exactly conservative semi-Lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique, J. Comput. Phys. 174 (2001) 171–207.
[22] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space, Comput. Phys. Commun. 120 (1999) 122–154.
[23] M. Shoucri, Nonlinear evolution of the bump-on-tail instability, Phys. Fluids 22 (1979) 038.
[24] M. Shoucri, A two-level implicit scheme for the numerical solution of the linearized vorticity equation, Int. J. Numer. Meth. Eng. 17 (1981) 1525.
[25] P.K. Smolarkiewicz, J.A. Pudykiewicz, A class of semi-Lagrangian approximations for fluids, J. Atmos. Sci. 49 (1992) 2082–2096.
[26] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, J. Comput. Phys. 149 (1999) 201–220.
[27] W.Y. Sun, K.S. Yeh, R.Y. Sun, A simple semi-Lagrangian scheme for advection equations, Q.J.R. Meteorol. Soc. 122 (1996) 1211–1226.
[28] R. Tanaka, T. Nakamura, T. Yabe, Constructing exactly conservative scheme in a non-conservative form, Comput. Phys. Commun. 126 (2000) 232–243.
[29] T. Umeda, M. Ashour-Abdalla, D. Schriver, Comparison of numerical interpolation schemes for one-dimensional electrostatic Vlasov code, J. Plasma Phys. 72 (2006) 1057–1060.
[30] T. Umeda, A conservative and non-oscillatory scheme for Vlasov code simulations, Earth Planets Space 60 (2008) 773–779.
[31] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A 150 (1990) 62.
[32] M. Zerroukat, N. Wood, A. Staniforth, A monotonic and positive-definite filter for a semi-Lagrangian inherently conserving and efficient (SLICE) scheme, Q.J.R. Meteorol. Soc. 131 (2005) 2923–2936.
[33] M. Zerroukat, N. Wood, A. Staniforth, The parabolic spline method (PSM) for conservative transport problems, Int. J. Numer. Meth. Fluids 51 (2006) 1297–1318.