HAL
open science

# Space-only hyperbolic approximation of the Vlasov equation

Philippe Helluy, Nhung Pham, Anaïs Crestetto

## To cite this version:

Philippe Helluy, Nhung Pham, Anaïs Crestetto. Space-only hyperbolic approximation of the Vlasov equation. ESAIM: Proceedings, 2013, 43, pp.17-36. 10.1051/proc/201343002 . hal-00797974

**HAL Id: hal-00797974**

**https://hal.science/hal-00797974v1**

Submitted on 7 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Space-only hyperbolic approximation of the Vlasov equation

Philippe HELLUY, Nhung PHAM, A. CRESTETTO

March 7, 2013

philippe.helluy@unistra.fr

**Abstract**

We construct an hyperbolic approximation of the Vlasov equation in which the dependency on the velocity variable is removed. The resulting model enjoys interesting conservation and entropy properties. It can be numerically solved by standard schemes for hyperbolic systems. We present numerical results for one-dimensional classical test cases in plasma physics: Landau damping, two-stream instability.

## 1 Introduction

The most precise models for plasma physics are based on a kinetic description of the charged particles. It is thus important to provide efficient numerical methods for solving such models. Generally, the particles are described by a distribution function $f$. The quantity $f(x, v, t)dx\,dv$ counts the particles having a position $x$ and a velocity $v$ at time $t$ in a small volume $dx\,dv$ in the phase space. The distribution function is solution of the Vlasov equation. The Vlasov equation is coupled with the Maxwell equations governing the electromagnetic field, or with the Poisson equation when only the electric field is relevant.

The main difficulty of a kinetic model is that it is a time-dependent system of PDE, set in a six-dimensional space, which leads to very heavy numerical simulations. In some physical configurations, with a strong effect of particles collisions, it is possible to reduce the complexity and returning to a three-dimensional model, such as the MHD model. But in many cases, when the collisions between particles are negligible, it is necessary to rely on the full kinetic model. This is a key point for understanding turbulence inside tokamak plasmas, for instance.

The Particle-In-Cell (PIC) method (see for instance [3, 14]) is a popular method for computing collisionless plasma, because it allows performing simulations in complex configurations with a relatively low amount of memory and CPU ressource. However the PIC method is based on an initial random choice of the particles and thus presents numerical noise. Also, it is difficult to ensure the energy conservation. Therefore, Eulerian methods for solving kinetic equations are becoming more and more popular. They allow a better control of the conservation and numerical errors (for a review of eulerian methods, see [7]). We can mention for instance the semi-lagrangian approach, which relies on a grid of the phase space. With this approach it is possible to perform realistic computations on tokamak core plasmas [10].

In this paper, we propose a general Eulerian approach for solving the Vlasov equation. It is based on the semi-discretization of the Vlasov equation with respect to the velocity variable. In this way, we construct a family of reduced models, depending on the velocity discretization parameter $P$. For low values of $P$, we recover fluid behavior, while high values of $P$ allow precise approximation of the kinetic model. For a fixed velocity parameter $P$, the approximate model is a linear hyperbolic system, with non-linear source terms. It is possible to establish in a very simple way conservation and entropy properties. The unknown depends on space and time instead of the full phase-space variables. These two features leads to a simpler reuse of existing solvers for hyperbolic equations. It is possible to incorporate in this way plasma kinetic models into a general, highly optimized solver. The coupling to other fluid plasma models is also simplified.

We apply our approach to the simple case of the one-dimensional Vlasov-Poisson system. We present numerical results for classical plasma physics test cases.

## 2 Plasma mathematical model

In the following, we present our approach on the one-dimensional classical Vlasov-Poisson system. But it can obviously be generalized to higher dimensions, relativistic cases and Vlasov-Maxwell systems.

### 2.1 Vlasov-Poisson 1D model

We consider the Vlasov equation

$$\partial_t f + v\partial_x f + E\partial_v f = 0. \tag{1}$$

The first unknown is the distribution function $f(x, v, t)$ that depends on the space variable $x$ the velocity variable $v$ and the time $t$. It measures, for instance, the quantity of electrons in a plasma having a velocity $v$ at point $x$ and time $t$. Therefore, it should be a non negative quantity. The second unknown is the electric field $E(x, t)$. We suppose that $x \in ]0, L[$ and we consider periodic boundary conditions at $x = 0$ and $x = L$

$$E(0, t) = E(L, t), \quad f(0, v, t) = f(L, v, t).$$

In principle, we should assume that $v \in ]-\infty, +\infty[$ and that the distribution function vanishes at $v = \pm\infty$. But for practical reasons, we have to bound the velocity space. We suppose that $v \in ]-V, V[$. The Vlasov equation (1) is a transport equation, we have thus also to apply boundary conditions at $v = -V$ or $v = V$ depending on the sign of $E$

$$E(x, t) > 0 \Rightarrow f(x, -V, t) = 0. \tag{2}$$

$$E(x, t) < 0 \Rightarrow f(x, V, t) = 0. \tag{3}$$

Assuming that the electric potential is also $L-$periodic, we can assume that the electric field has a zero mean value

$$\int_x E = 0. \tag{4}$$

It is solution of the one-dimensional Poisson equation

$$\partial_x E = -1 + \int_v f dv. \tag{5}$$

The quantity $\rho_0 = 1$ corresponds, for instance, to the charge density of the background ions.

We also call the systems (1), (5) the Vlasov-Poisson 1D model.

The equations (1), (5) are supplemented by an initial condition

$$E(x, 0) = E_0(x), \quad f(x, v, 0) = f_0(x, v).$$

In our applications (Landau damping and two-stream instability), this initial condition also satisfies

$$\frac{1}{L}\int_x\int_v f_0 = 1, \quad \int_x\int_v v f_0 = 0. \tag{6}$$

Then, we can deduce that these two conditions remain true for all time[1]

$$\frac{1}{L}\int_x\int_v f = 1, \quad \int_x\int_v v f = 0. \tag{7}$$

Let us recall that property (7) is very specific to the test cases that we compute below. For other classical test cases, such as the bump-on-tail instability test case [6], the property has to be adapted.

[1] In fact this result is true if $f$ has compact support in $v$. In our applications, it is only an approximation. We suppose that we can neglect $f(x, \pm V, t)$.

### 2.2 Vlasov-Ampère 1D Model

The simulation of the Vlasov-Poisson system requires the resolution of an elliptic equation for the electric potential. In the one-dimensional framework, the equation reduces to (5), which is numerically solved by an FFT-based algorithm. It is interesting to propose an equivalent equation for the electric field, which implies the simple numerical resolution of a local differential equation.

We consider the Poisson equation (5) and take the derivative of $f$ with respect to $x$. Suppose that $f$ is a function with compact support in $v$ and that we can change the order of integration of $f$, then

$$\begin{array}{rrcl} & \partial_t\left(\partial_x E\right) & = & \partial_t\left(-1+\int_v f\right) \\ \Leftrightarrow & \partial_x\partial_t E & = & \int_v \partial_t f \\ & & = & \int_v\left(-v\partial_x f - E\partial_v f\right) \\ & & = & -\partial_x\left(\int_v vf\right) - E\int_v \partial_v f, \end{array}$$

by hypothesis, $f$ has a compact support in $v$, thus

$$\begin{array}{rcl} E\int_v \partial_v f & = & E.\left[f\right]_{v=-\infty}^{v=+\infty} \\ & = & 0, \end{array}$$

thus, finally, we have

$$\begin{array}{rrcl} & \partial_x\partial_t E & = & -\partial_x\left(\int_v vf\right) \\ \Rightarrow & \partial_t E & = & -\int_v vf + C(t). \end{array} \tag{8}$$

We integrate now with respect to $x$ the two sides of this equation for obtaining

$$\int_x \left(\partial_t E\right) dx = \int_x \left(-\int_v vf + C(t)\right) dx.$$

We have

$$\int_x \left(\partial_t E\right) dx = \partial_t\left(\int_x E dx\right) = 0.$$

Finally, we deduce

$$C(t) = \frac{1}{L}\int_x\int_v vf dv dx = 0.$$

The equation (8) becomes

$$\partial_t E = -\int_v f v dv \tag{9}$$

which we call the Ampère equation.

We also call the system (1), (9) the Vlasov-Ampère 1D model.

## 3 Velocity basis expansion

For each time $t$ the Vlasov equation is set in a six-dimensional $(x, v)$ space. We will perform a semi-discretization of (1) with respect to the velocity $v$ in order to obtain a system of PDE set only in the $x$ space. We shall call this new system of equations the reduced Vlasov model. We could expand the distribution function $f$ on a basis of arbitrary functions depending on $v$. See for instance [4] and included references. For practical numerical reasons, we decide to choose a classical Lagrange finite element interpolation basis, because it leads to a sparse matrix representation of the hyperbolic system.

### 3.1 Continuous interpolation by the finite element method

First, we introduce some notations and recall how is constructed the finite element basis. We consider an arbitrary polynomial degree $d$. The reference element is defined by

$$\hat{Q} = ]-1,1[.$$

We define the $d+1$ reference nodes by

$$\hat{N}_i = -1 + 2\frac{i-1}{d}, \quad i = 1 \cdots d+1.$$

We mesh the interval $]-V,V[$ with $N$ finite elements $(Q_i)_{i=1\cdots N}$ and nodes $(N_j)_{j=1\cdots P}$. The total number of nodes in this interval is $P = d \cdot N + 1$. In practice, we suppose that the nodes are equally spaced in $]-V,V[$

$$N_j = -V + \frac{2V}{dN}(j-1).$$

As usual, in our program, we have a connectivity array for detecting that node $N_j$ is the $k^{th}$ local node of a given element $Q_i$

$$j = \text{connec}(k,i) = k + (i-1)d, \quad 1 \leq k \leq d+1, \quad 1 \leq i \leq N.$$

We also use the notation

$$N_j = N_{k,i}$$

and then, element $Q_i$ has its support in the interval $[N_{1,i}, N_{d+1,i}]$.

We construct a transformation $\tau_i$ that maps element $\hat{Q}$ onto $Q_i$. For this purpose we consider the Lagrange polynomials on $\hat{E}$, defined by

$$\hat{L}_k(\hat{v}) = \prod_{l \neq k} \frac{\hat{v} - \hat{N}_l}{\hat{N}_k - \hat{N}_l}. \tag{10}$$

The transformation is then given by

$$\tau_i(\hat{v}) = \sum_{k=1}^{d+1} \hat{L}_k(\hat{v}) N_{k,i}. \tag{11}$$

Because the nodes of the mesh are equally spaced in our application, in practice the transformation $\tau_i$ is linear. We construct the interpolation basis in such a way that each basis function $\varphi_j$ is associated to a node $N_j$ of the mesh and satisfies

$$\varphi_i(N_j) = \delta_{ij},$$

where $\delta_{ij}$ denotes the Kronecker symbol. We recall how to compute the basis function $\varphi_j$. Let $v \in ]-V,V[$. Necessarily, $v$ belongs at least to one finite element $Q_i$. Two cases are possible

1. Node $N_j$ belongs to finite element $Q_i$, i.e.

   $$\exists k, N_j = N_{k,i}.$$

   Then

   $$\varphi_j(v) = \hat{L}_k(\hat{v}), \quad v = \tau_i(\hat{v}). \tag{12}$$

2. Node $N_j$ does not belong to $Q_i$, then

   $$\varphi_j(v) = 0.$$

### 3.2 Application to Vlasov velocity discretization

We suppose that the distribution function is approximated by expansion on the basis $\{\varphi_j\}_{j=1\cdots P}$

$$f(x,v,t) = \sum_{j=1}^{P} w_j(x,t)\varphi_j(v), \tag{13}$$

we shall also use the convention of sum on repeated indices

$$f(x,v,t) = w_j(x,t)\varphi_j(v). \tag{14}$$

Because of the interpolation property of the basis$\{\varphi_j\}_{j=1\cdots P}$

$$\varphi_i(N_j) = \delta_{ij},$$

we have

$$f(x,N_i,t) = \sum_{j=1}^{P} w_j(x,t)\varphi_j(N_i) = w_i(x,t).$$

Therefore, we can approximate the initial condition in the following way

$$w_j(x,0) = f(x,N_j,0) = f_0(x,N_j).$$

We now describe the weak formulation from which we construct the reduced model. Let $f$ be a solution of the Vlasov equation (1) with the boundary condition defined by (2), (3). We recall that

$$E^+ = \max(0,E), \quad E^- = \min(0,E).$$

Then, for all basis function $\varphi_i$, we have

$$\partial_t \int_v f\varphi_i + \partial_x \int_v vf\varphi_i + E\int_v \partial_v f\varphi_i + \frac{E^+}{2} f(\cdot,-V,\cdot)\varphi(-V) - \frac{E^-}{2} f(\cdot,V,\cdot)\varphi_i(V) = 0. \tag{15}$$

This weak form of the transport equation introduces an upwinding only at the boundaries $v = \pm V$. The advantages and drawbacks of this approach are discussed in [9].

If we introduce the expression (14) in the weak formulation (15), we obtain

$$\begin{aligned} &\partial_t w_j \int_v \varphi_i\varphi_j + \partial_x w_j \int_v v\varphi_i\varphi_j + Ew_j \int_v \varphi_i\varphi_j' \\ &+\frac{E^+}{2} w_j\varphi_j(-V)\varphi_i(-V) - \frac{E^-}{2} w_j\varphi_j(V)\varphi_i(V) \quad = \quad 0. \end{aligned}$$

We can thus introduce the following matrices of dimension $P \times P$

$$M = (\int_v \varphi_i\varphi_j), \quad A = (\int_v v\varphi_i\varphi_j).$$

$$B(E) = \frac{E^+}{2E}\varphi_j(-V)\varphi_i(-V) - \frac{E^-}{2E}\varphi_j(V)\varphi_i(V) + \int_v \varphi_i\varphi_j'.$$

We obtain the following equation

$$M\partial_t w + A\partial_x w + EB(E)w = 0, \tag{16}$$

in which $w$ is the vector of $P$ components

$$w = (w_1, w_2, ..., w_P)^T.$$

Obviously $A$ and $M$ are symmetric matrices and $M$ is positive definite. It is then classical to prove that system (16) is hyperbolic, i.e. that $M^{-1}A$ is diagonalizable with real eigenvalues [8].

On the other hand, thanks to an integration by part, we find that

$$B_{ij} = -B_{ij} \text{ if } (i,j) \neq (1,1) \text{ and } (i,j) \neq (P,P).$$

In other words the matrix $B$ is "almost" skew-symmetric.

In addition

$$\text{if } E > 0, \quad B_{11}(E) = 0, \quad B_{PP}(E) = \frac{1}{2} \tag{17}$$

and

$$\text{if } E < 0, \quad B_{11}(E) = -\frac{1}{2}, \quad B_{PP}(E) = 0. \tag{18}$$

Multiplying equation (16)by $w$ and integrating in $x$, we can establish the following energy estimate

$$\frac{d}{dt}\int_x w^T M w = \frac{1}{2}E^- w_1^2 - \frac{1}{2}E^+ w_P^2 \leq 0. \tag{19}$$

This estimate shows that it is essential to modify the source term matrix $\left(\int_v \varphi_i\varphi_j'\right)$ with the correction (17), (18) in order to obtain a stable approximation.

## 3.3 Practical computation of $M, A, B$

### 3.3.1 Gauss-Legendre integration

The (normalized) Legendre polynomials are defined, for $n \geq 0$, by

$$l_n(x) = \frac{\sqrt{n+\frac{1}{2}}}{n!2^n} \frac{d^n}{dx^n}((x^2-1)^n).$$

The $n$ zeros of $l_n$ are distinct and in $]-1,1[$. We denote by $(\xi_i)_{i=1\cdots n}$ the zeros of $l_n$ and by

$$\omega_i = \frac{-\sqrt{2n+1}\sqrt{2n+3}}{(n+1)l_n'(\xi_i)l_{n+1}(\xi_i)}$$

the integration weights. Then, we have the quadrature formula

$$\int_{-1}^{1} g(v)dv \simeq \sum_{i=1}^{n} \omega_i g(\xi_i).$$

The formula is exact if $g$ is a polynomial of degree $2n-1$.

### 3.3.2 Computation

Firstly, we observe that $M, A$ and $B$ are sparse matrices. Indeed, with $j_1, j_2$ such that $N_{j_1}, N_{j_2}$ are not in the same element, we have

$$\varphi_{j_1}(v).\varphi_{j_2}(v) = 0, v \in ]-V, V[$$

thus

$$M_{j_1 j_2} = A_{j_1 j_2} = B_{j_1 j_2} = 0.$$

More precisely, we can state that $M, A$ and $B$ are band matrices. The bandwidth is equal to the degree $d$ of the local polynomial interpolation. We only have to take care of the non-zero terms of these matrices.

In order to achieve exact numerical integration, we consider $d+1$ Gauss-Legendre integration points $\hat{\xi}_i$ on the reference element $\hat{Q}$ with weights $\hat{\omega}_i$. It ensures the correct integration of a polynomial of degree $2d+1$. Such polynomials arise in the terms $\int_v v\varphi_i(v)\varphi_j(v)dv$. The algorithm for assembling the matrices is then the following:

1. We loop on the elements of the mesh, we loop on the local nodes of each element. Using the connectivity array, we construct the shape of the sparse matrices. More precisely, if

   $$j_1 = \text{connec}(k_1, i) \text{ and } j_2 = \text{connec}(k_2, i)$$

   then the corresponding $(j_1, j_2)$ elements in the matrices are $\neq 0$.

2. We loop again on the elements of the mesh. For each element $Q_i$ we perform the following algorithm

```
do k1=1,d+1
do k2=1,d+1
    j1=connec(k1,i)
        j2=connec(k2,i)
        M(j1,j2)=M(j1,j2)+∫_{v∈Q_i} φ_{j1}φ_{j2}
        A(j1,j2)=A(j1,j2)+∫_{v∈Q_i} vφ_{j1}φ_{j2}
        B(j1,j2)=B(j1,j2)+∫_{v∈Q_i} φ_{j1}φ'_{j2}
    enddo
enddo
```

Of course, in this algorithm, the matrices $M, A$ and $B$ are stored in a sparse way, for saving computer memory. The first and last terms of the diagonal of $B$ have also to be corrected according to (17) and (18).

For computing the integrals over $Q_i$, we use the geometric transformation $\tau_i$ and Gauss numerical integration.

For $M$, we obtain

$$\begin{aligned}\int_{v\in Q_i} \varphi_{j_1}\varphi_{j_2} &= \int_{\hat{v}=-1}^{1} \hat{L}_{k_1}\hat{L}_{k_2}\frac{dv}{d\hat{v}} \\ &= \int_{\hat{v}=-1}^{1} \hat{L}_{k_1}\hat{L}_{k_2}\tau_i'(\hat{v}) \\ &= \sum_{l=1}^{d+1} \hat{\omega}_l \hat{L}_{k_1}(\hat{\xi}_l)\hat{L}_{k_2}(\hat{\xi}_l)\tau_i'(\hat{\xi}_l).\end{aligned}$$

For $A$, we obtain

$$\begin{aligned}\int_{v\in Q_i} v\varphi_{j_1}\varphi_{j_2} &= \int_{\hat{v}=-1}^{1} v\hat{L}_{k_1}\hat{L}_{k_2}\frac{dv}{d\hat{v}} \\ &= \int_{\hat{v}=-1}^{1} \tau_i(\hat{v})\hat{L}_{k_1}\hat{L}_{k_2}\tau_i'(\hat{v}) \\ &= \sum_{l=1}^{d+1} \hat{\omega}_l \hat{L}_{k_1}(\hat{\xi}_l)\hat{L}_{k_2}(\hat{\xi}_l)\tau_i(\hat{\xi}_l)\tau_i'(\hat{\xi}_l).\end{aligned}$$

For $B$, we obtain

$$\begin{aligned}\int_{v\in Q_i} \varphi_{j_1}\varphi_{j_2}' &= \int_{\hat{v}=-1}^{1} \hat{L}_{k_1}\frac{d}{dv}\hat{L}_{k_2}\frac{dv}{d\hat{v}} \\ &= \int_{\hat{v}=-1}^{1} \hat{L}_{k_1}\frac{d}{d\hat{v}}\hat{L}_{k_2} \\ &= \sum_{l=1}^{d+1} \hat{\omega}_l \hat{L}_{k_1}(\hat{\xi}_l)\frac{d}{d\hat{v}}\hat{L}_{k_2}(\hat{\xi}_l).\end{aligned}$$

We observe that we can compute and store the values and derivatives of the reference basis functions at the reference Gauss points at the beginning of the computation. In the same way, we can compute the upwind and downwind convection matrices

$$A^{\pm} = (\int_v v^{\pm}\varphi_i\varphi_j),$$

where

$$v^+ = \max(v, 0),\ v^- = \min(v, 0).$$

These matrices are used in the upwind scheme below.

In Appendix 1 we give additional details on the storage and assembly of the sparse matrices $M, A$ and $B$.

# 4 Finite volume schemes

We have replaced the original Vlasov equation, written in the $(x, v)$-space by an hyperbolic system written in the $x$-space only. The new representation (16) is called the reduced Vlasov system. We can solve this system by any efficient numerical method for hyperbolic systems. In the sequel, we compare the first order upwind Godunov scheme and a second order centered scheme. In the future we intend to switch to a more general, high order, Discontinuous Galerkin (DG) approach.

## 4.1 Vlasov equation

We consider a finite volume approximation. We assume that the spatial domain $]0, L[$ is split into $N_x$ cells. The cell $C_i$ is the interval $\left]x_{i-1/2}, x_{i+1/2}\right[$, $i = 1..N_x$. For practical reasons, we also consider two virtual cells $C_0$ and $C_{N_x+1}$ for applying the periodic boundary condition. At the beginning of a time step, we copy the values of the cell $C_{N_x}$ to the cell $C_0$, and the values of the cell $C_1$ to the cell $C_{N_x+1}$. The center of the cell $C_i$ is $x_i = i\triangle x - \frac{\triangle x}{2}$. The space step is $\triangle x = \frac{L}{N_x}$. We also consider a sequence of times $t_n$, $n \in \mathbb{N}$, such that $t_0 = 0$ and $t_n = n\triangle t$, where $\triangle t$ satisfies the following CFL condition

$$\triangle t = \alpha \frac{\triangle x}{V},\ 0 < \alpha \leq 1.$$

We are looking for an approximation of the vector $w$ in the cell $C_i$

$$w_i^n \simeq w(x,t), \quad x \in C_i,\ t = t_n$$

We consider the initial conditions $E_0 = E(\cdot, t = 0)$, and $f_0 = f(\cdot, \cdot, t = 0)$. We recall that the $j^{th}$ component of the initial condition is given by

$$(w(x,0))_j = f_0(x, N_j)$$

for each $j = 1 \cdots P$.

We consider the reduced Vlasov model (16), we have

$$\begin{aligned} M\partial_t w + A\partial_x w + EB(E)w &= 0 \\ \Rightarrow M\partial_t w &= -A\partial_x w - EB(E)w \end{aligned}$$

If we denote $S = -EB(E)w$, we obtain the following evolution equation

$$M\partial_t w = -A\partial_x w + S. \tag{20}$$

We consider a finite volume approximation of (20). We denote by $w_i(t)$ a piecewise constant approximation of $w$ in each cell

$$w_i(t) \simeq w(x,t), \quad x \in C_i.$$

The numerical flux is denoted by $(w_a, w_b) \mapsto F(w_a, w_b)$. We obtain the following semi-discrete (in space) approximation

$$M\partial_t w_i = -\frac{F(w_i, w_{i+1}) - F(w_{i-1}, w_i)}{\triangle x} + S(w_i^n).$$

Of course we have also to introduce a time discretization in order to compute

$$w_i^n \simeq w_i(t_n), \quad x \in C_i.$$

### 4.1.1 First-order Euler method

For the time integration we can consider the classical explicit Euler method, which is of order 1. It reads

$$\begin{aligned} M\frac{w_i^{n+1} - w_i^n}{\triangle t} &= -\frac{F(w_i^n, w_{i+1}^n) - F(w_{i-1}^n, w_i^n)}{\triangle x} + S(w_i^n) \\ \Rightarrow \quad w_i^{n+1} &= w_i^n - \frac{\triangle t}{\triangle x}\left(M^{-1}F(w_i^n, w_{i+1}^n) - M^{-1}F(w_{i-1}^n, w_i^n)\right) + \triangle t M^{-1} S(w_i^n) \end{aligned} \tag{21}$$

### 4.1.2 Second order improved Euler method

A time second order scheme is given by the following algorithm

$$\begin{aligned} M\frac{w_i^{n+1/2} - w_i^n}{\triangle t/2}, &= -\frac{F(w_i^n, w_{i+1}^n) - F(w_{i-1}^n, w_i^n)}{\triangle x} + S(w_i^n), \\ M\frac{w_i^{n+1} - w_i^n}{\triangle t}, &= -\frac{F(w_i^{n+1/2}, w_{i+1}^{n+1/2}) - F(w_{i-1}^{n+1/2}, w_i^{n+1/2})}{\triangle x} + S(w_i^{n+1/2}). \end{aligned} \tag{22}$$

### 4.1.3 Numerical flux

We consider the centered or upwind numerical flux. The centered flux is given by

$$F(w_L, w_R) = A\frac{w_L + w_R}{2} - \frac{\varepsilon V}{2}(W_R - W_L).$$

The upwind flux is given by

$$F(w_L, w_R) = A^+ w_L + A^- w_R.$$

Of course, the first order explicit Euler integration cannot be used with the centered flux, because the resulting scheme is always unstable.

### 4.1.4 Implementation aspects

In practice, for saving CPU time and memory, we use two subroutines for computations with the sparse matrices in skyline format. The first one computes the product of a sparse matrix and a vector. The other one is used to solve the linear system (by the $LU$ method)

$$Mw = s,$$

where $M$ is also a sparse matrix.

## 4.2 Ampère equation

We have explained how we evolve the reduced Vlasov equation. We have also to evolve the electric field. We have implemented two methods. The simplest method consists in solving the Ampère equation (9). The other method is based on a resolution of the Poisson equation (5).

We first consider the Ampère equation

$$\partial_t E = -\int_v f v dv.$$

From the representation (14) for $f$, we have

$$\begin{array}{rcl} \int_v f v dv & = & \sum_{j=1}^P w_j(x,t) \int_v v\varphi_j(v) dv \\ & = & \sum_{j=1}^P w_j(x,t)\zeta_j, \end{array}$$

where

$$\zeta_j = \int_v v\varphi_j(v) dv.$$

The vector $(\zeta_j)_{j=1\cdots P}$ is computed with an assembly algorithm and Gauss numerical integration as described in Section 3.3.

For the time integration we use, as for $w$, the first order scheme or the second order improved Euler scheme. Actually, as it is explained in Section 5.2, we have to modify a little bit the time integration in order to obtain precise results.

## 4.3 Poisson equation

We can also compute the electric field by solving the Poisson equation (5). As in many other works, we use the FFT (Fast Fourier Transform) algorithm.

We consider the cell-centered electric field approximation $(E_0, E_1, ..., E_{N_x-1})$. We denote by $\widehat{E}_k$ its DFT (Discrete Fourier Transform, we denote by $I = \sqrt{-1}$)

$$\widehat{E}_k = \sum_{j=0}^{N-1} E_j e^{\frac{-2I\pi jk}{N}},\ k = 0 \cdots N-1,$$

and the inverse DFT is given by

$$E_j = \frac{1}{N}\sum_{k=0}^{N-1} \widehat{E}_k e^{\frac{2I\pi jk}{N}},\ \forall k = 0..N-1.$$

We consider the Poisson equation (5)

$$\begin{array}{rcl} \partial_x E & = & -1 + \int_v f dv \\ & = & -1 + \sum_{j=1}^{P} w_j(x,t) \int_v \varphi_j(v) dv \\ & = & -1 + \sum_{j=1}^{P} w_j(x,t) \varrho_j \end{array}$$

in which $\varrho_j = \int_v \varphi_j(v)dv$. We can compute $\varrho_j$ over each element $Q_i$ by the assembly algorithm of Section 3.3. We use the notation

$$\sigma(x) = -1 + \int_v f(x,v,\cdot)dv.$$

A centered finite-difference approximation of the Poisson equation reads

$$\partial_x E(x_i) \simeq \frac{E_{i+1} - E_{i-1}}{2\triangle x} = \sigma(x_i). \tag{23}$$

Taking the DFT of the two sides of (23) we obtain

$$\widehat{E}_k = \frac{\triangle x}{I sin \frac{2\pi k}{N_x}} \widehat{\sigma}_k,\ k \neq 0.$$

And, when $k = 0$, we have

$$\widehat{E}_0 = \sum_{j=0}^{N-1} E_j = 0,\ \text{(because of condition (4))}.$$

Of course, we use the efficient FFT algorithm (see [12]) for computing the DFT.

### 4.4 Electric energy

The electric energy is defined by

$$\Xi(t) = \sqrt{\int_0^L E(x,t)^2 dx}.$$

We approximate it by

$$\Xi(t_n) \simeq \sqrt{\triangle x \sum_{i=1}^{N_x} (E_i^n)^2}$$

## 5 Test cases

For testing our solver, we consider several test cases. The first two tests are designed in order to validate the pure transport in the $x$ or $v$ direction. In particular, we will show that the correction (17), (18) is essential in order to obtain correct results. In our numerical experiments, the discretization parameters are $N = 20$, $d = 5$ and $N_x = 128$.

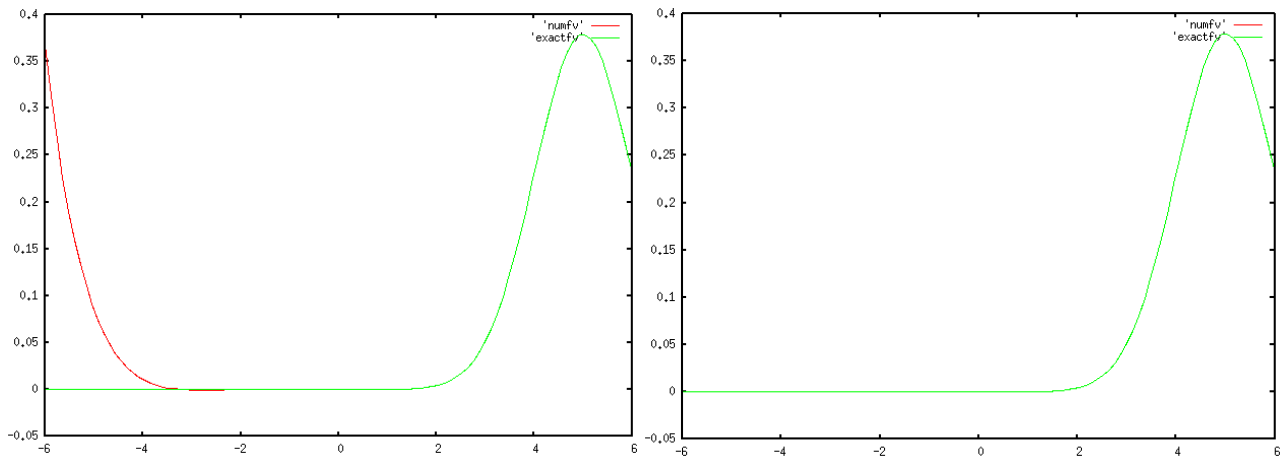


Figure 1: The $v$-transport for the Vlasov equation after $T = 5$, we compare the exact solution (green curve) and the numerical solution (red curve). Left: without the correction on $B(E)$. Right: with correction.

## 5.1 The transport equation

Consider the Vlasov equation

$$\partial_t f + v\partial_x f + E\partial_v f = 0.$$

If we suppose that $\partial_x f = 0$ and the electric field is constant with respect to $t$, namely $E(x,t) = E(x,t=0) = E_0(x)$, thus the Vlasov equation becomes

$$\partial_t f + E(x)\partial_v f = 0, \tag{24}$$

which is a $v$-transport equation.

If we suppose that the electric field vanishes at any time and any position, we obtain the $x$-transport equation

$$\partial_t f + v\partial_x f = 0 \tag{25}$$

### 5.1.1 The $v$-transport equation

Consider the $v$-transport equation (24) with a given initial condition $f_0$. Thanks to the method of characteristics, the exact solution is given by

$$f(x,v,t) = f_0(x, v - E(x)t).$$

Suppose that the initial conditions is given by

- The distribution function

$$f_0(x,v) = (1 + \epsilon cos(kx))\frac{1}{\sqrt{2\pi}}e^{-\frac{v^2}{2}},$$

- The electric field

$$E_0(x) = 1,$$

- The domain $L = \frac{2\pi}{k}$.

Let us take the values of parameter $k = 0.2$ and $\epsilon = 5 \times 10^{-2}$. The results are given on Figure (1). We compare the numerical solution of the $v-$transport equation and the exact solution. We can see that without the correction (17), (18) the numerical solution is wrong.

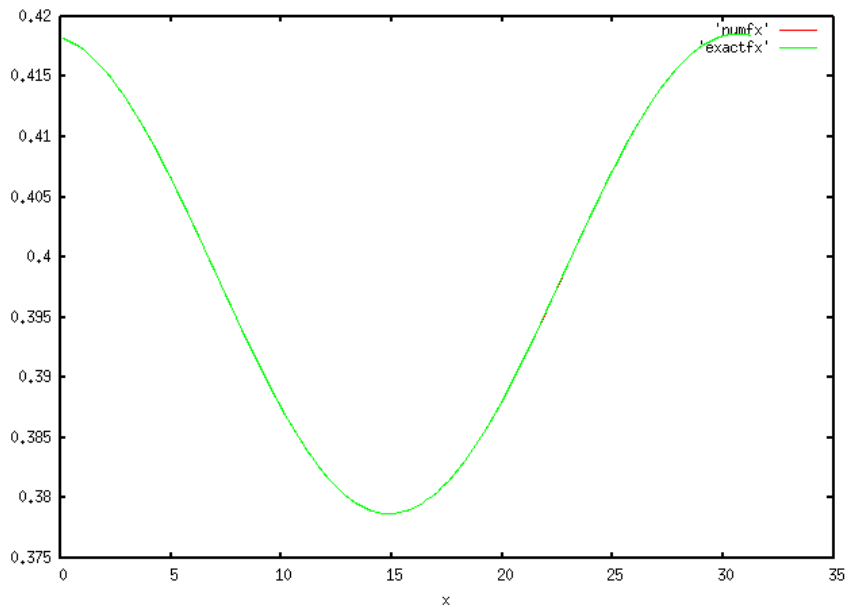


Figure 2: The $x$-transport test for the Vlasov equation after $T = 20$. The exact and numerical curves are superimposed.

#### 5.1.2 The $x$-transport equation

We consider also the $x$-transport equation (25) with a given initial condition $f_0$. The exact solution of this transport equation is given by

$$f(x, v, t) = f_0(x - vt, v).$$

With the same initial condition $f_0$ as in the $v$-transport test, we obtain the result of Figure 2.

### 5.2 The Landau damping

In this test case, we consider the following initial data

- The distribution function

$$f_0(x, v) = (1 + \epsilon cos(kx)) \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}},$$

- the electric field

$$E_0(x) = \frac{\epsilon}{k} sin(kx),$$

- the domain size $L = \frac{2\pi}{k}$.

For small $\epsilon$, thanks to a linear approximation of the non-linear Vlasov-Poisson system, it is possible to compute an approximate analytic solution of the electric field. The details of the computation are given in [13]. The electric field is given by

$$E(x, t) = 4\epsilon r e^{\omega_i t} \sin(kx) \cos(\omega_r t - \varphi), \tag{26}$$

where $\omega_i, \omega_r$ are the real part and the imaginary part of $\omega$, respectively. The numerical values of $\omega$, $r$ and $\varphi$ are given in the following table

| $k$ | $\omega$ | $re^{I\varphi}$ |
|---|---|---|
| 0.5 | $\pm 1.4156 - 0.1533I$ | $0.3677e^{\pm I 0.536245}$ |
| 0.4 | $\pm 1.2850 - 0.0661I$ | $0.424666e^{\pm I 0.3357725}$ |
| 0.3 | $\pm 1.1598 - 0.0126I$ | $0.63678e^{\pm I 0.114267}$ |
| 0.2 | $\pm 1.0640 - 5.510 \times 10^{-5} I$ | $1.129664e^{\pm I 0.00127377}$ |

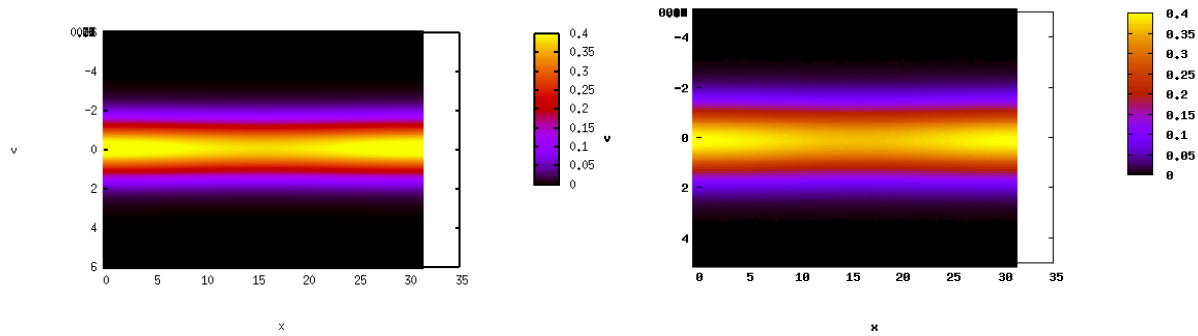


Figure 3: The distribution function of the Landau damping test case at time $t = 0$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

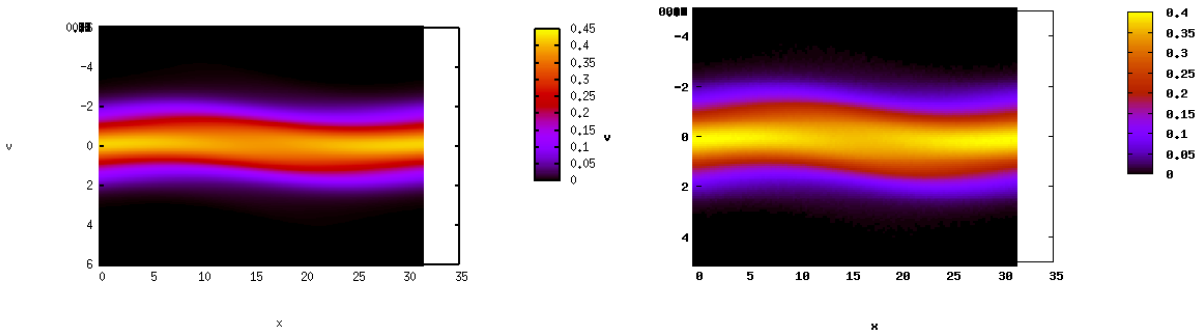


Figure 4: The distribution function of the Landau damping test case at time $t = 10$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

In addition, the distribution function can be computed by a well validated method, such as the PIC method. We compare our numerical results with the PIC results and also with the analytic solution.

Firstly, we compare the distribution function of the reduced Vlasov-Poisson method and of the PIC method (taken from [5]). The value of the parameters are $k = 0.2$ and $\epsilon = 5 \times 10^{-2}$. We plot the distribution function computed by the two methods at different times $t = 0$, $t = 10$, $t = 20$, $t = 30$ and at $t = 100$. The results are on Figure 3, 4, 5, 6 and 7.

The reduced Vlasov approximation satisfies only an $L^2$ stability estimate (19). Such estimate does not ensure the positivity of $f$. Indeed, in our simulations we observe at some points slightly negative values of $f$. But the numerical results are anyway very satisfactory. We explain in Appendix 2 how we could avoid negative values of $f$.

We also plot the logarithm of the electric energy in order to compare the reduced Vlasov-Poisson method with the PIC method on Figure 8.

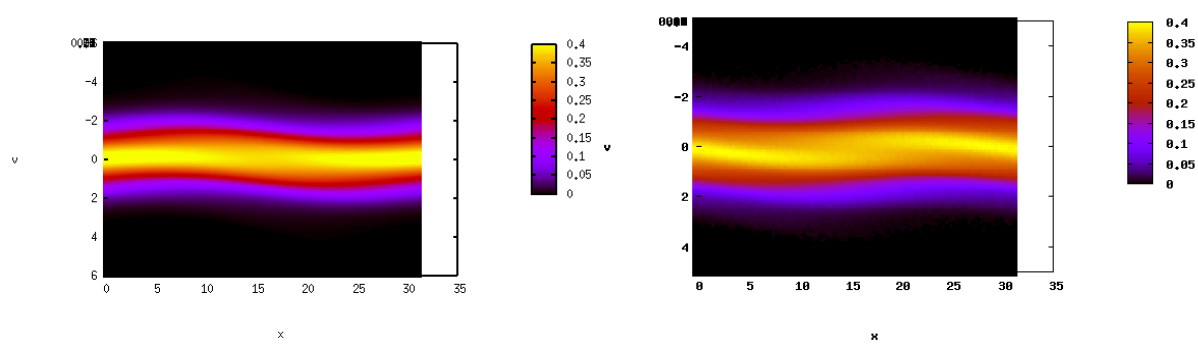


Figure 5: The distribution function of the Landau damping test case at time $t = 20$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

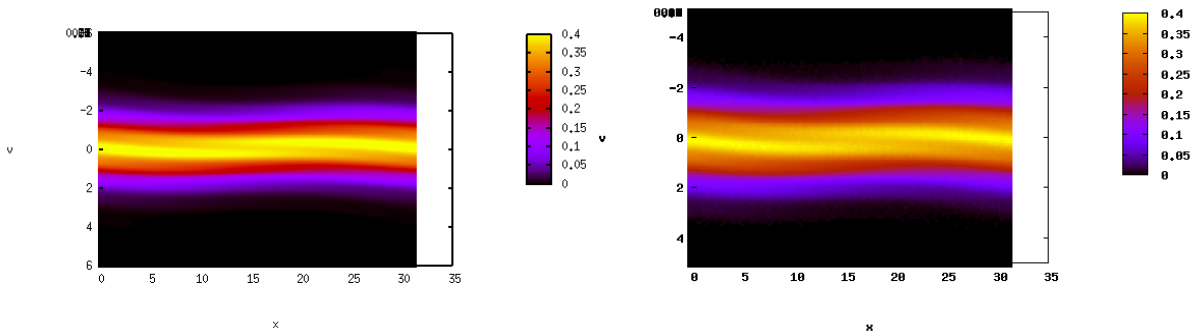


Figure 6: The distribution function of the Landau damping test case at time $t = 30$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

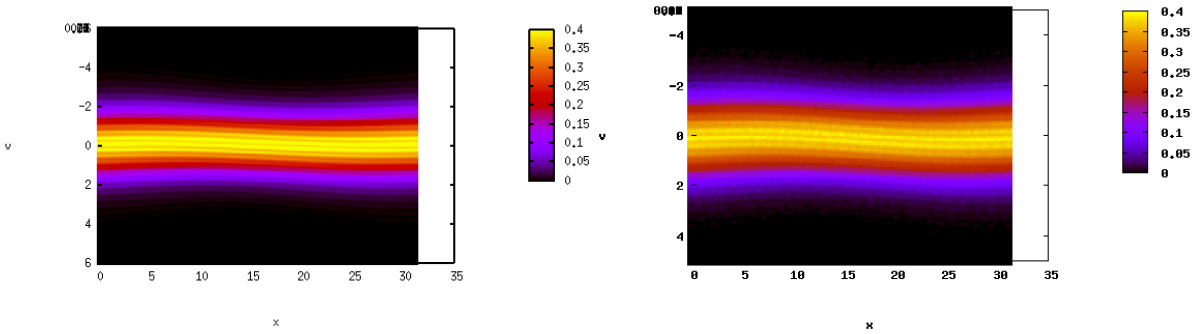


Figure 7: The distribution function of the Landau damping test case at time $t = 100$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

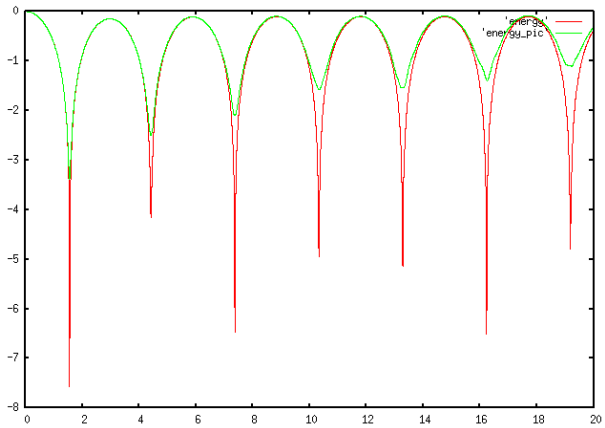


Figure 8: The electric energy of the Landau damping test case up to time $t = 20$, the green curve is computed by the PIC method and the red curve is computed with the reduced Vlasov-Poisson method (second order scheme).

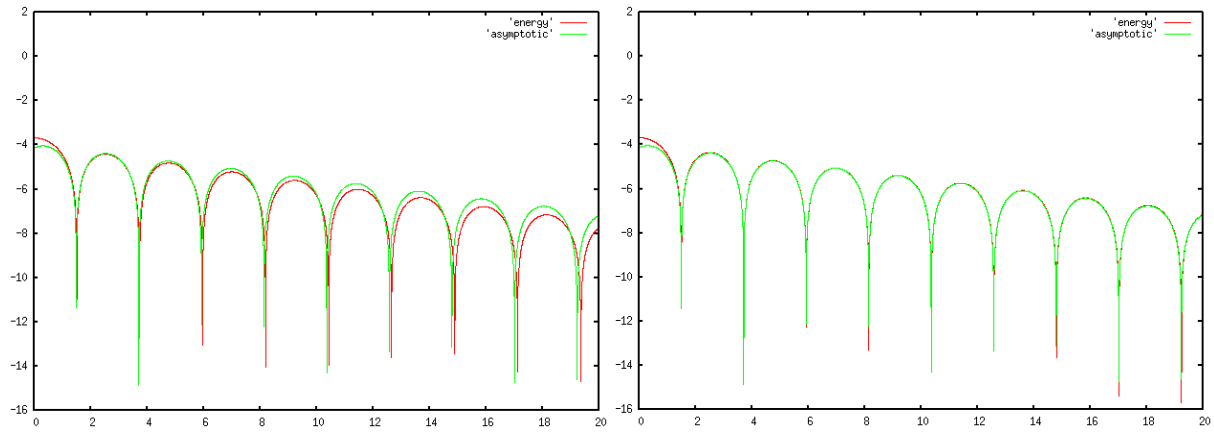


Figure 9: The electric energy of the Landau damping test case up to time $t = 20$, the green curve "asymptotic" is computed with the formula (26) and the red curve "energy" is computed with the reduced Vlasov-Poisson method. Left: Euler scheme with the upwind flux. Right: Rung-Kutta scheme with the centered flux.

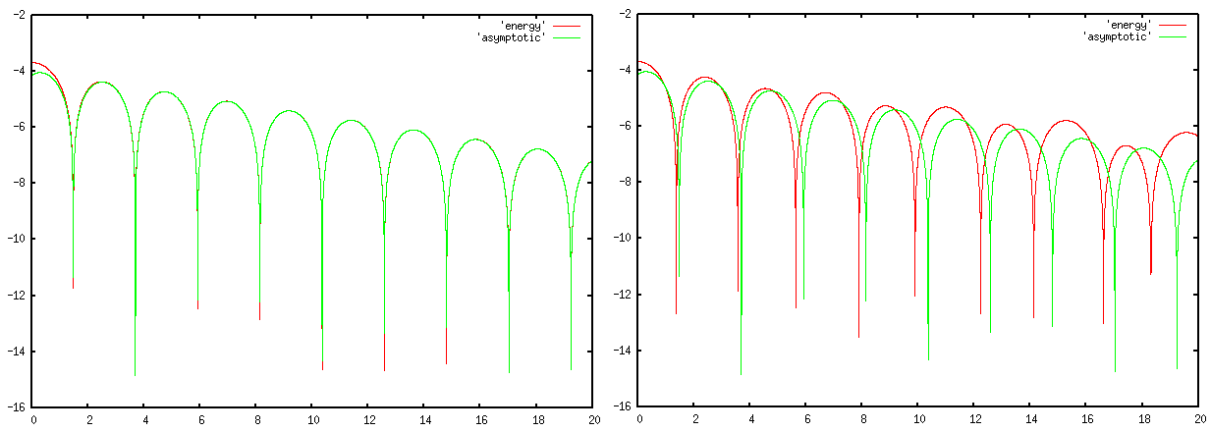


Figure 10: The electric energy of the Landau damping test case up to time $t = 20$, the green curve "asymptotic" is computed with the formula (26) and the red curve "energy" is computed with the reduced Vlasov-Ampère method. Left: semi-implicit scheme. Right: explicit scheme.

Now, we take the parameters $k = 0.5$ and $\epsilon = 5 \times 10^{-3}$. In Figure 9, we plot the logarithm of the electric energy computed by reduced Vlasov-Poisson method and by the formula (26). We compare the first and second order schemes. We can see that the second order scheme is more precise than the first order scheme.

We have also tested the reduced Vlasov-Ampère resolution. We have observed that it is not possible to use a fully explicit time integration for the Ampère equation (9). If we compute $E_i^{n+1}$ from $w_i^n$ by the first order explicit time integration, for instance, the results are not very precise and we observe an error increase as time goes by. We obtain better results if we use $w_i^{n+1}$ for estimating $\partial_t E(x_i, t_n)$. This is a kind of semi-implicit scheme. The results are compared on Figure 10.

### 5.3 Two-stream instability

In this test case, the initial distribution function is given by

$$f_0(x, v) = (1 + \epsilon cos(kx)) \frac{1}{2\sqrt{2\pi}} \left( e^{-\frac{(v-v_0)^2}{2}} + e^{-\frac{(v+v_0)^2}{2}} \right),$$

in which the velocity $v_0$ is given. The value of parameters for this test case are $k = 0.2$, $\epsilon = 5 \times 10^{-3}$ and $v_0 = 3$. The distribution function is plotted at times $t = 0$, $t = 20$, $t = 25$ and $t = 50$ in Figures 11, 12, 13 and 14. We compare the PIC method and the reduced Vlasov-Poisson method.

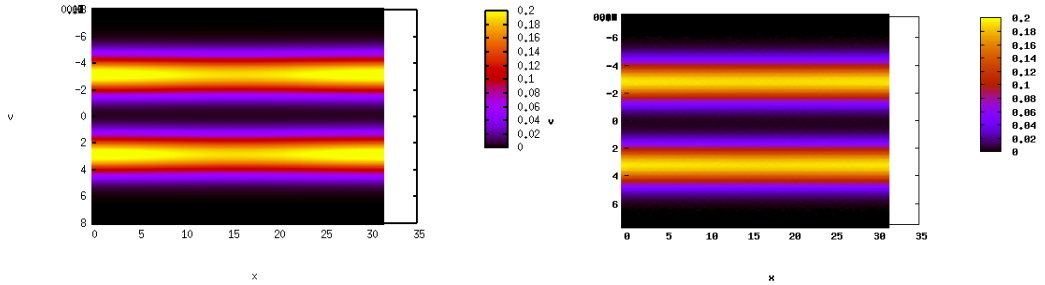

Figure 11: The distribution function of the two-stream test case at time $t = 0$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

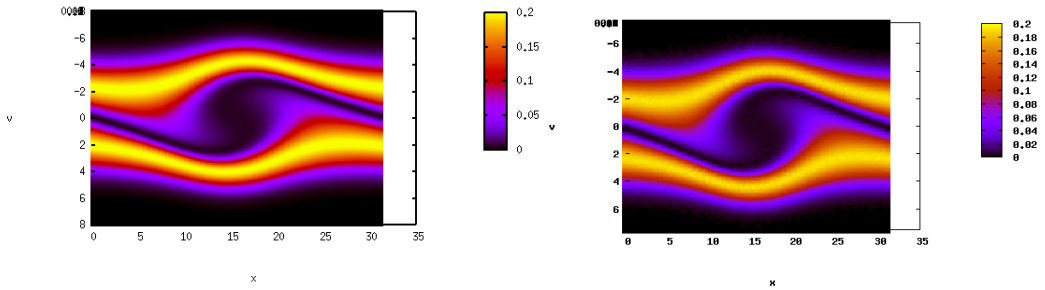

Figure 12: The distribution function of the two-stream test case at time $t = 20$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

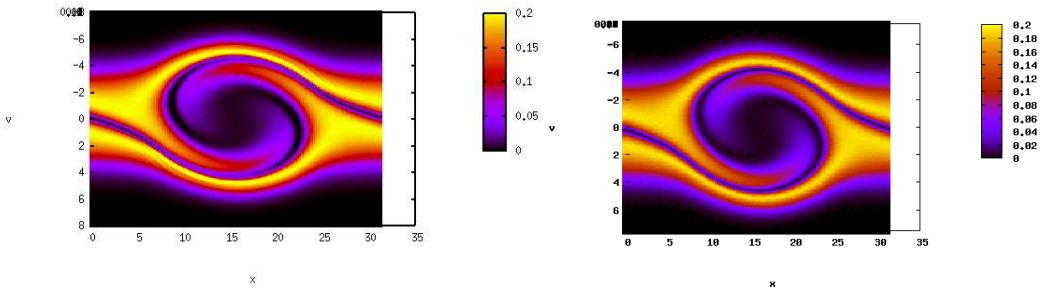

Figure 13: The distribution function of the two-stream test case at time $t = 25$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

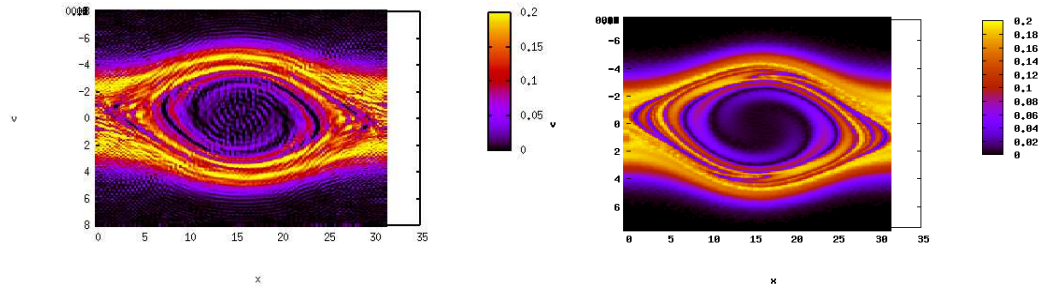

Figure 14: The distribution function of the two-stream test case at time $t = 50$. Left: reduced Vlasov-Poisson method. Right: the PIC method.

At time $t = 50$, we observe small numerical oscillations in the reduced Vlasov-Poisson method. These oscillations are maybe due to the fact that we have almost no upwind mechanism in the resolution of the transport equation. Further investigations are necessary in order to control this phenomenon. For instance, we intend to implement an upwind Discontinuous Galerkin (DG) method in order to introduce more dissipation in the numerical method, while keeping high order.

In order to provide a better understanding of the dissipation, we also introduce in Appendix 2 a non-linear version of the reduced Vlasov approach. The non-linear approach allows to replace the energy estimate by an entropy estimate. It also allows to construct natural dissipative source term, which can be used for stabilizing the numerical method or for introducing a physical dissipative mechanism, such as collisions effect.

# 6 Conclusion

In this paper, we have constructed a new method for approximating the Vlasov equation. We first performed an approximation in the velocity direction. The resulting system is a first order system of hyperbolic equations. Its convective part is linear while the nonlinearity is concentrated into the source term, which couples the kinetic equation and the electromagnetic field. The solutions satisfy natural conservation and energy estimates. We are also able to provide a nonlinear version of the model, where the distribution function is positive and satisfies a natural entropy estimate.

It is then possible to apply to the reduced Vlasov model the whole range of numerical methods that have been developed for hyperbolic systems. In this paper we compare two classical methods: the upwind first order scheme and a centered second order scheme. We obtain good results on two classical test cases in plasma physics: Landau damping and two-stream instability.

In the future, we plan to extend our method to higher dimensional problems, relativistic particle beams, gyrokinetic plasma approximations and weak collisions models.

# References


[1] Altmann, Christoph; Belat, Thomas; Gutnic, Michael; Helluy, Philippe; Mathis, Hélène; Sonnendrücker, Éric; Angulo, Wilfredo; Hérard, Jean-Marc A local time-stepping discontinuous Galerkin algorithm for the MHD system. CEMRACS 2008—Modelling and numerical simulation of complex fluids, 33–54, ESAIM Proc., 28, EDP Sci., Les Ulis, 2009.

[2] Barth, Timothy On discontinuous Galerkin approximations of Boltzmann moment systems with Levermore closure. Comput. Methods Appl. Mech. Engrg. 195 (2006), no. 25-28, 3311–3330.

[3] C.K. Birdsall and A.B. Langdon. Plasma Physics via Computer Simulation. Institute of Physics (IOP), Series in Plasma Physics, 1991.

[4] Le Bourdiec, S.; de Vuyst, F.; Jacquet, L. Numerical solution of the Vlasov-Poisson system using generalized Hermite functions. Comput. Phys. Comm. 175 (2006), no. 8, 528–544.

[5] A. Crestetto, Optimisation de méthodes numériques pour la physique des plasmas. Application aux faisceaux de particules chargées, her thesis, October 2012.

[6] Crouseilles, Nicolas; Mehrenberger, Michel; Sonnendrücker, Eric Conservative semi-Lagrangian schemes for Vlasov equations. J. Comput. Phys. 229 (2010), no. 6, 1927–1953.

[7] F. Filbet, E. Sonnendrücker. Comparison of Eulerian Vlasov solvers Comput. Phys. Commun., 150 (2003), pp. 247–266.

[8] Godlewski, Edwige; Raviart, Pierre-Arnaud Numerical approximation of hyperbolic systems of conservation laws. Applied Mathematical Sciences, 118. Springer-Verlag, New York, 1996.

[9] Johnson, C.; Pitkäranta, J. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. Math. Comp. 46 (1986), no. 173, 1–26.

[10] G. Latu, N. Crouseilles, V. Grandgirard, E. Sonnendrücker, Gyrokinetic Semi-Lagrangian Parallel Simulation using a Hybrid OpenMP/MPI Programming, Recent Advances in PVM an MPI, Springer, LNCS, pp 356-364, Vol. 4757, (2007).

[11] Perthame, B. Boltzmann type schemes for gas dynamics and the entropy property. SIAM J. Numer. Anal. 27 (1990), no. 6, 1405–1421.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery: Numerical Recipes in C.

[13] E. Sonnendrücker Approximation numerique des equations de Vlasov-Maxwell, Notes du cours de M2, 18 mars 2010.

[14] D. Tskhakaya, R. Schneider, Optimization of PIC codes by improved memory management, Journal of Computational Physics, Volume 225, Issue 1, 1 July 2007, Pages 829-839

# Appendix 1: skyline storage of the matrices

For storing a sparse matrix $M$ (or $A$ or $B$) of size nnoe×nnoe in the skyline format, we use the following arrays: mdiag(1:nnoe) for storing the diagonal, msup(1:nsky) and mlow(1:nsky) for storing the upper and lower parts. The integer nsky is unknown at the beginning. We need also an additional array mkld(1:nnoe+1) for locating the beginning of each column (row) in msup (mlow). We use the convention mkld(1)=1 and mkld(nnoe+1)=nsky+1. In practice, for constructing mkld, we use an intermediate array prof(1:nnoe). prof(i) contains the number of stored elements of $M$ in column (row) i in msup (mlow). Thus prof(1)=0. For building prof, we use the following algorithm

```
prof=0
do k=1,nel
    do ii=1,d+1
        do jj=1,d+1
        i=connec(ii,k)
        j=connec(jj,k)
        prof(j)=max(prof(j),j-i)
        prof(i)=max(prof(i),i-j)
        enddo
    enddo
enddo
```

Once prof is known, mkld is built with the following algorithm

```
mkld(1)=1
do i=1,nnoe
    mkld(i+1)=mkld(i)+prof(i)
enddo
```

At the end of this algorithm, we know nsky=mkld(nnoe+1)-1, we can thus allocate the memory for msup and mlow. Finally, let $(i, j)$ corresponds to a non-zero element of matrix $M$ (it means that we can find an element index $k$ and two local node indices $ii$ and $jj$ such that $i = connec(ii, k)$ and $j = connec(jj, k)$. We can recover $M_{ij}$ from the arrays mdiag, mlow, msup, mkld by the following algorithm

```
if (i.eq.j) then
    M_ij = mdiag(i)
else if (j.gt.i) then
    M_ij = msup(l) with l=mkld(j+1)-j+i
```

```
else
    M_ij = mlow(l) with l=mkld(i+1)-i+j
endif
```

The structure of the matrices $A, B$ and $M$ are the same. We have to compute mkld only once.

# Appendix 2: Non-linear Vlasov reduction

For improving the dissipation, we consider now a source term in the Vlasov equation. It can represent collisions between particles. The Vlasov equation (1) becomes

$$\partial_t f + v\partial_x f + E\partial_v f = Q(f), \tag{27}$$

where $Q(f)$ is a source term, which we denote by the collision kernel. We also consider an entropy $S(f)$. The entropy is supposed to be a smooth strictly convex function of $f$. In practice, we will consider the following entropies

$$\text{(a) } S(f) = \frac{f^2}{2}, \quad \text{(b) } S(f) = f(\ln f - 1), \quad \text{(c) } S(f) = \frac{f^2}{2} - \varepsilon \ln f, \quad \varepsilon > 0.$$

The entropy variable is

$$g = S'(f).$$

Considering $S^*$, the Legendre transform of $S$

$$S^*(g) = \max_f (gf - S(f)),$$

we also have

$$f = S^{*'}(g).$$

In the previous sections, we expanded $f$ on an interpolation basis. Now, we expand the entropy variable $g$ in the Legendre basis. A first advantage is that if $S^{*'}$ is positive, the distribution function is automatically positive. It is the case with choice (b) and (c). With choice (a) we recover the reduced Vlasov model of the previous sections because then $S'(f) = g = f$.

Suppose that

$$g(x,t,v) = \sum_{k=1}^{P} g_k(x,t)\varphi_k(v). \tag{28}$$

We denote by $\Pi$ the orthogonal projection of $g$ on a subspace $\mathcal{V}_0$ of the interpolation space $\mathcal{V} = \text{span}\,\{\varphi_i,\ 1 \le i \le P\}$. For a given constant $\lambda > 0$, we can then consider the collision kernel

$$Q(f) = \lambda\left(\Pi g - g\right). \tag{29}$$

If $f$ has a compact support in $v$, we can prove the following results

- $\int_v Q(f)\varphi dv = 0, \quad \forall \varphi \in \mathcal{V}_0$. In particular, if $v \to v^m$ is in the space $\mathcal{V}_0$ then the $m-$moment of $f$ is conserved in the sense that
$$\int_x \int_v f v^m dv dx = Cst.$$
- The total entropy $\Sigma = \int_v S(f)dv$ is decreasing
$$\partial_t \Sigma + \partial_x G(\Sigma) \le 0, \quad \text{with } G(\Sigma) = \int_v vS(f).$$

The proof is an immediate adaptation of techniques presented in many works on non-linear hyperbolic systems and Boltzmann theory. We refer for instance to [2, 11, 8].

The collision kernel (29) can be used of course for introducing a physical phenomenon. But we can also use it as a numerical tool for damping numerical oscillations. We will investigate this kind of tools in forthcoming works.