Vlasov simulations of beams with a moving grid

E. Sonnendrücker[1], F. Filbet[2], A. Friedman[3], E. Oudet[1], J.-L. Vay[4]
[1]*IRMA, Université Louis Pasteur, Strasbourg, France*
[2]*MAPMO, CNRS, Orléans, France*
[3]*Lawrence Livermore National Laboratory, Livermore, California, USA*
[4]*Lawrence Berkeley National Laboratory, Berkeley, California, USA*

1 Introduction

Thanks to the rapid increase of computing power in recent years, simulations of plasmas and particle beams based on direct solution of the Vlasov equation on a multi-dimensional phase-space grid are becoming attractive as an alternative to Particle-In-Cell (PIC) simulations. Their strength lies essentially in the fact that they are noiseless and that all parts of phase space, including the tail of the distribution, are equally well resolved. Their major drawback is that, for inhomogeneous systems, many of the grid points (where no particles are present) are wasted. This is especially the case for beam simulations where the beam moves rapidly through the phase space (due to varying alternating-gradient focusing forces, for example). This inefficiency has made such Vlasov simulations unsuitable for those cases.

One of the methods which has proven very efficient for the direct resolution of the Vlasov equation is the semi-Lagrangian method [1, 3]. It consists in updating the values of the distribution function at the grid nodes by following the characteristics ending at these nodes backwards and interpolating the value at the bottom of the characteristics from the known values at the previous time step. In general the interpolation grid is fixed, but this is not mandatory.

This paper introduces the concept of a moving grid which is mapped at each time step from a logical uniform grid to the beam, so that it contains the whole beam without needing too many points with vanishing values of the distribution function. In order to implement this new method, we introduce a new time stepping algorithm which does not rely on the time splitting procedure traditionally used in Vlasov solvers.

The model we consider throughout this paper is the nonrelativistic Vlasov equation coupled self-consistently with Poisson's equation. It reads

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0, \tag{1}$$

the self electric field $\mathbf{E}$ is computed from Poisson's equations

$$-\epsilon_0 \nabla^2 \phi = \rho(\mathbf{x}, t) = q \int f(\mathbf{x}, \mathbf{v}, t)\, dv, \qquad \mathbf{E} = -\nabla \phi.$$

The magnetic field is external and considered to be known.

The paper is organized as follows: We first recall the traditional semi-Lagrangian method. After that, we introduce a new time stepping algorithm that does not require splitting and which is

required when the coordinate axes are not aligned with the x and v directions. We then describe the moving grid algorithm in a general setting first and finally present its application to the simulation of beams in transverse phase-space along with some first numerical results.

2 The semi-Lagrangian method for the Vlasov equation

The semi-Lagrangian method consists in computing a numerical approximation of the solution of the Vlasov equation (1) on a phase space grid by using the property of the equation that the distribution function f is conserved along characteristics. More precisely, for any times s and t we have

$$f(\mathbf{x}, \mathbf{v}, t) = f(\mathbf{X}(s; \mathbf{x}, \mathbf{v}, t), \mathbf{V}(s; \mathbf{x}, \mathbf{v}, t), s),$$

where $(\mathbf{X}(s; \mathbf{x}, \mathbf{v}, t), \mathbf{V}(s; \mathbf{x}, \mathbf{v}, t)$ are the characteristics of the Vlasov equation which are solution of the system of ordinary differential equations

$$\frac{d\mathbf{X}}{ds} = \mathbf{V}, \tag{2}$$

$$\frac{d\mathbf{V}}{ds} = \mathbf{E}(\mathbf{X}(s), s) + \mathbf{V}(s) \times \mathbf{B}(\mathbf{X}(s), s), \tag{3}$$

with initial conditions $\mathbf{X}(t) = \mathbf{x}$, $\mathbf{V}(t) = \mathbf{v}$.

From this property, f^n being known one can induce a numerical method for computing the distribution function f^{n+1} at the grid points $(\mathbf{x}_i, \mathbf{v}_j)$ consisting of the following two steps:

1. For all i, j, compute the origin of the characteristic ending at $\mathbf{x}_i, \mathbf{v}_j$, i.e. an approximation of $\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$, $\mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$.

2. As $f^{n+1}(\mathbf{x}_i, \mathbf{v}_j) = f^n(\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1}), \mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1}))$, f^{n+1} can be computed by interpolating f^n which is known at the grid points at the points $\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1}), \mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$.

This method can be simplified by performing a time-splitting separating the advection phases in physical space and velocity space, as in this case the characteristics can be solved explicitely.

The semi-Lagrangian method does not require any specific interpolation scheme. However, numerical experience dictates use of a high enough order so that diffusion, which is the most important numerical error in this method, is limited to an acceptable level. The only natural requirement is that the interpolation enables to get a good continuous reconstruction of f (at the lowest possible cost). Hence one could use a different interpolation grid for each time step. This can save a lot of time when the support of the distribution function evolves considerably over time, as is the case for a beam in a periodic focusing channel. In the remainder of this paper, we extend the semi-Lagrangian method to allow use of an interpolation grid which is moving in time.

3 A second order algorithm for the characteristics

When the grid transformation mixes space and velocity components the traditional splitting method ([2]) cannot be performed. Therefore we need to introduce an efficient method for solving the characteristics without splitting. A possible option would be to use the two time-steps method that

was introduced in [1]. However, this has the drawback of decoupling even and odd time steps. Let us instead introduce a second order predictor-corrector method to compute the origin of the characteristics based on an isochronous leap-frog algorithm.

Algorithm 1: Knowing the final position (X^{n+1}, V^{n+1}) at time step t_{n+1}, as well as f^n, ρ^{n-1}, $\mathbf{E}^n$ we can compute the initial position (X^n, V^n) using the following algorithm:

1. Predict $\bar{\mathbf{E}}^{n+1}$ using the continuity equation (or directly Ampere's law in 1D)

$$\rho^{n+1} = \rho^{n-1} - 2\Delta t \nabla \cdot \mathbf{J}^n,\ \mathbf{J}^n = q\int f^n(\mathbf{x},\mathbf{v})\mathbf{v}\, d\mathbf{v},\ -\nabla^2\phi^{n+1} = \frac{\rho^{n+1}}{\epsilon_0},\quad \bar{\mathbf{E}}^{n+1} = -\nabla\phi^{n+1}.$$

2. $\mathbf{V}^{n+\frac{1}{2}} = \mathbf{V}^{n+1} - \frac{\Delta t}{2}\bar{\mathbf{E}}^{n+1}(\mathbf{X}^{n+1})$; $\mathbf{X}^n = \mathbf{X}^{n+1} - \Delta t \mathbf{V}^{n+\frac{1}{2}}$; $\mathbf{V}^n = \mathbf{V}^{n+\frac{1}{2}} - \frac{\Delta t}{2}\mathbf{E}^n(\mathbf{X}^n)$.
3. $f^{n+1}(\mathbf{X}^{n+1}, \mathbf{V}^{n+1}) = interpolation(f^n)(\mathbf{X}^n, \mathbf{V}^n)$; $\rho^{n+1} = \int f^{n+1}\, dv$,
4. Correct $\bar{\mathbf{E}}^{n+1}$ using $-\nabla^2\phi^{n+1} = \frac{\rho^{n+1}}{\epsilon_0},\quad \bar{\mathbf{E}}^{n+1} = -\nabla\phi^{n+1}$.
5. If $\|\bar{\mathbf{E}}^{n+1} - \bar{\mathbf{E}}^{n+1}_{prev}\| > threshold$ go back to 2.

Our first 1D tests show that the error decreases very rapidly: the relative error is of the order of 10^{-2} after the predictor step, decreases to around 10^{-9} after one corrector step and reaches 10^{-15} after two corrector steps. Hence, given the other errors inherent in any discrete algorithm, a single corrector step is sufficient. Therefore the cost of the algorithm, which comes mostly from the interpolation step, is roughly the same as for the split algorithm, where one interpolation at each split step is necessary.

4 The semi-Lagrangian method on a moving grid

4.1 The algorithm for the Vlasov solve

The semi-Lagrangian method consist in two conceptually different steps:

1. An advection step which consists in solving a large number of decoupled ordinary differential equations. This step is completely independent of the grid and is most naturally performed in the physical space,

2. An interpolation step which is necessary to compute the value of the distribution function at the origin of the characteristics which are not on the grid. The interpolation grid is only needed to reconstruct the distribution function at every point in phase space at one given time step and needs not be the same at two different time steps.

In order to optimize step 2 one needs to position the interpolation points so as to be able to reconstruct f with a given acuracy at the lowest possible cost. In beam dynamics simulations the global movement of the beam is mostly determined by the external forces and even if the self forces are important it can be determined by the evolution of the envelope equation. Hence this information should be used to position the grid points.

On the other hand, in order to simplify the interpolation step, we choose to always perform it on a uniform logical grid, the position in the actual phase space of the grid points being given by an invertible mapping φ_t from the logical grid to the physical grid. We choose φ_t such that it is continuously differentiable as well as its inverse. The subscript t reminds us that φ_t can be different for different times.

In order to describe the algorithm, we need to introduce a few notations regarding the logical and physical grids. We shall denote with a $*$ the coordinates in the logical grid. Then for a given point $(\mathbf{x}, \mathbf{v})$ in the physical phase space, we have

$$(\mathbf{x}^*, \mathbf{v}^*) = \varphi_t^{-1}(\mathbf{x}, \mathbf{v}) \qquad \text{or} \qquad (\mathbf{x}, \mathbf{v}) = \varphi_t(\mathbf{x}^*, \mathbf{v}^*).$$

Let us also introduce the distribution function on the logical grid defined by $f^*(x^*, v^*, t) = f(\varphi_t(x^*, v^*), t)$. Then, the property that f is conserved along the characteristics translates into the following new conservation property for f^* that shall be used in the algorithm:

$$\begin{aligned} f^*(x^*, v^*, t) &= f(\varphi_t(x^*, v^*), t), \\ &= f(x, v, t), \\ &= f(X(s; x, v, t), V(s; x, v, t), s), \\ &= f(\varphi_s(X^*(s; x, v, t), V^*(s; x, v, t)), s), \\ &= f^*(X^*(s; x, v, t), V^*(s; x, v, t), s), \end{aligned}$$

where s parameterizes motion along the characteristics. Now, f^n being known as well as $\varphi_n (= \varphi_{t_n})$ and φ_{n+1}, the following algorithm can be used to compute f^{n+1}.

Algorithm 2:

1. Compute positions in physical phase-space of grid points where f^{n+1} is to be computed: $(x_{i,j}^{n+1}, v_{i,j}^{n+1}) = \varphi_{n+1}(x_i^*, v_j^*)$, where (x_i^*, v_j^*) are the nodes of the logical grid.

2. Compute origin of grid points $(x_{i,j}^{n+1}, v_{i,j}^{n+1})$ using algorithm 1 or similar. We denote by $(X_{i,j}^n, V_{i,j}^n)$ these origins.

3. Transform $(X_{i,j}^n, V_{i,j}^n)$ back to the logical grid at time t_n: $(X_{i,j}^{*n}, V_{i,j}^{*n}) = \varphi_n^{-1}(X_{i,j}^n, V_{i,j}^n)$.

4. Interpolate f^{*n} at origin of characteristics on logical grid to get f^{n+1}, as $f^{*(n+1)}(x_i^*, v_j^*) = f^{*n}(X_{i,j}^{*n}, V_{i,j}^{*n})$.

Steps 2 and 4 exist in any nonsplit semi-Lagrangian code. Hence the extension of such codes to moving grids can be performed easily by implementing the transform from logical to physical space (step 1) and the back transform from physical to logical space (step 3).

4.2 Coupling with the Poisson equation

One of the problems that can arise with the moving grid is that the grid points fail to be aligned along a given position $\mathbf{x}$ in physical space. Hence when velocity moments, in particular ρ, need to be computed we need to interpolate f at some specified points for the numerical integration.

In order to minimize these interpolations the numerical integration is performed using an adaptive Gauss quadrature. The grid motion could be constrained to avoid this (forcing points to line up in columns of constant $\mathbf{x}$); but if it is desired that the mesh motion track the phase space flow as closely as possible (so as to minimize numerical diffusion) such measures are needed. The trade-off for real applications has yet to be assessed.

In some systems, the requirement that the mapping from logical to physical mesh remain simple may itself limit the ability of the grid to follow the phase space flow.

5 Application to beam simulation in transverse phase space

In this kind of simulation the beam envelope can evolve greatly, leaving at any given time an large portion of a fixed grid empty and inducing much unnecessary computation. For this reason, we use a moving grid, and adapt that grid at each time step to the RMS beam envelope. Hence the transform φ is a rotation coupled to a dilation following the envelope motion. In the following, we consider x to be a dimensionless quantity, scaled to the size of the physical domain and

More precisely, the ellipse defined by its larger dimension a, its smaller dimension b and its angle θ with respect to the (Ox) axis can be defined by the RMS parameters of the beam from the relations

$$\tan 2\theta = \frac{2\langle xx' \rangle}{\langle x^2 \rangle - \langle x'^2 \rangle},$$

$$a = \sqrt{2(\cos^2\theta\langle x^2\rangle + \sin^2\theta\langle x'^2\rangle + 2\sin\theta\cos\theta\langle xx'\rangle)},$$

$$b = \sqrt{2(\sin^2\theta\langle x^2\rangle + \cos^2\theta\langle x'^2\rangle - 2\sin\theta\cos\theta\langle xx'\rangle)},$$

where for a function $\chi(x, x')$ we denote by

$$\langle \chi(x, x') \rangle = \frac{\int \chi(x, x') f(x, x')\, dxdx'}{\int f(x, x')\, dxdx'}.$$

In this calculation, we consider x to be a dimensionless quantity, scaled to the size of the physical domain and, as usual in beam physics, $x' = \frac{v_x}{v_z}$, where v_z is the longitudinal velocity of the beam.

The computing box at time t_{n+1} is determined using a, b and θ obtained from RMS values of the beam computed at time t_n.

6 Numerical results

In order to validate our method and pinpoint its advantages, we applied it in cases where the RMS envelope motion is important, namely first in the case of a mismatched beam in a uniform focusing channel and then in the case of a matched beam in a periodic focusing channel.

We considered the model problem of the transverse axisymmetric Vlasov-Poisson equation with vanishing canonical angular momentum. This problem reads

$$\frac{\partial f}{\partial t} + v_r \frac{\partial f}{\partial r} + (F_{app} + \frac{q}{m} E_r) \frac{\partial f}{\partial v_r} = 0,$$

$$\frac{1}{r}\frac{d}{dr}(rE_r) = \rho = \int f\, dv_r.$$

6.1 Mismatched Gaussian beam in a uniform focusing channel

We consider a mismatched Gaussian beam

$$f_0(r, v_r) = n_0 e^{-(r^2/a^2 + v^2/v_{th}^2)}$$

with a linear applied field of the form αr.

We represent snapshots of the beam and the moving computing box in Figure 1. The results are very satisfying as the computing box follows very precisely the global motion of the beam.

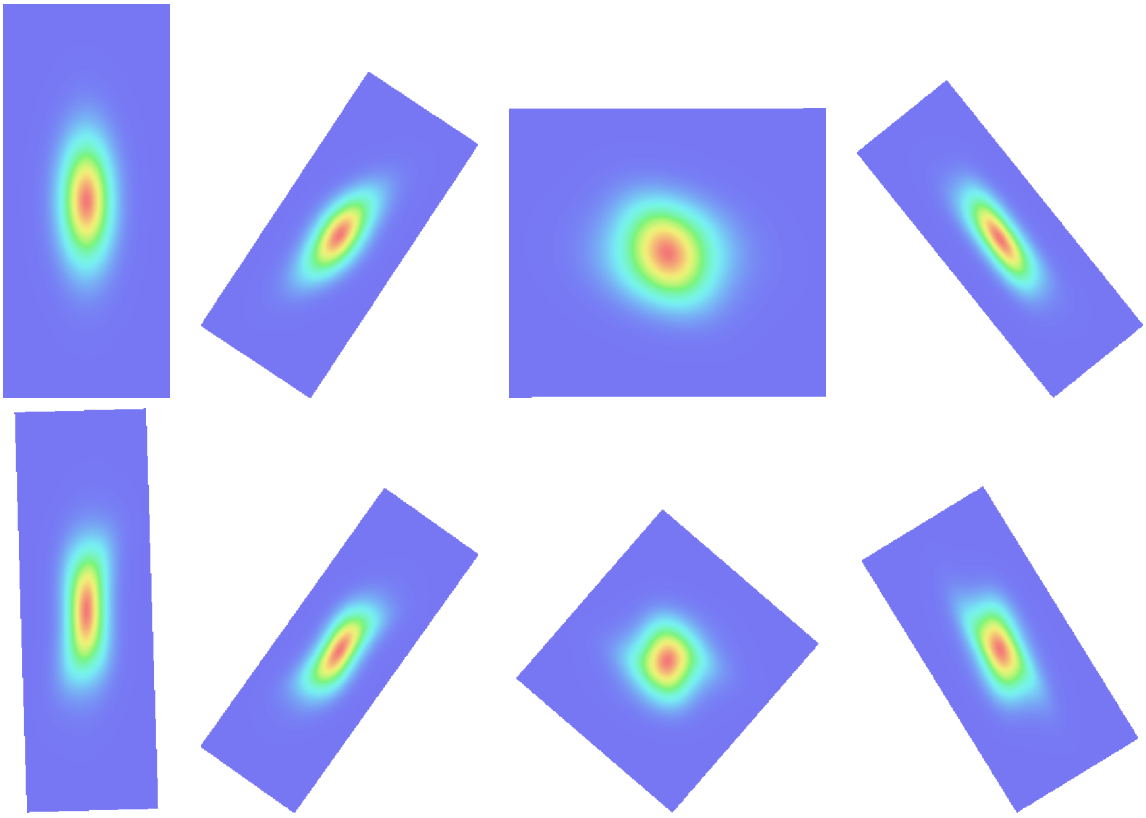

Figure 1: Snapshots of the motion of a mismatched Gaussian beam in a uniform focusing channel

6.2 Matched Gaussian beam in a periodic focusing channel

We consider here a matched Gaussian beam

$$f_0(r, v_r) = n_0 e^{-(r^2/a^2 + v^2/v_{th}^2)}$$

with a linear applied field of the form $\alpha(z)r$, where $\alpha(z)$ is a piecewise constant function switching beetween 0 and some fixed value, the pattern repeating periodically (the repetition length is called a lattice period).

We represent snapshots of the beam and the moving computing box in Figure 2. Here as well, although some filaments are generated by the nonlinear forces, the computing box obtained from the RMS envelope of the beam does a good job in helping to determine the region of non vanishing f.

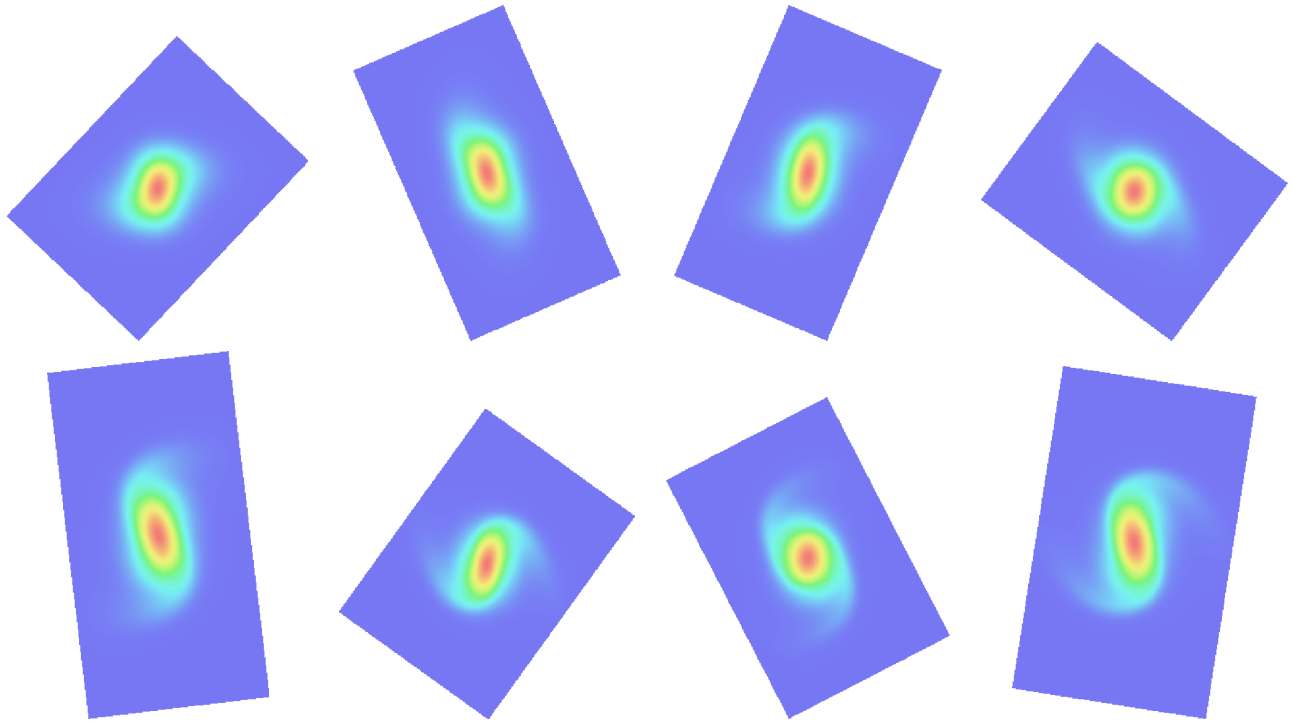

Figure 2: Snapshots of the motion of a matched Gaussian beam in a periodic focusing channel

7 Conclusion

The transform method appears very promising for beam simulations. Use of a very simple transform given by the RMS motion of the beam allows the grid to follow more closely the global motion of the beam, and thus reduces considerably the size of the grid necessary for the simulation. This method has been implemented on a 1D model problem. The next step will be to assess its usefulness in more realistic cases, including 2D transverse simulations and/or 1D longitudinal simulations. Finally, we note that many other applications of such a method outside of beam physics can be envisioned.

References

[1] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The Semi-Lagrangian Method for the Numerical Resolution of Vlasov Equations. *J. Comput. Phys.* 149: (1998) pp. 201–220.

[2] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space. *J. Comput. Phys.* 22: (1976) pp. 330–348.

[3] E. Sonnendrücker, J. Barnard, A. Friedman, D. Grote, S. Lund, Simulation of heavy ion beams with a semi-Lagrangian Vlasov solver, *Nuclear Inst. and Methods in Physics Research*, Section A, 464, no. 1-3, (2001) 470-476.