Comparison of two Eulerian solvers for the four-dimensional Vlasov equation: Part I

N. Crouseilles* M. Gutnic† G. Latu‡ E. Sonnendrücker§

Abstract

This paper presents two methods for solving the four-dimensional Vlasov equation on a grid of the phase space. The two methods are based on the semi-Lagrangian method which consists in computing the distribution function at each grid point by following the characteristic curve ending there. The first method reconstructs the distribution function using local splines which are well suited for a parallel implementation. The second method is adaptive using wavelets interpolation: only a subset of the grid points are conserved to manage data locality. Numerical results are presented in the second part.

1 Introduction

The Vlasov equation describes the evolution of a system of charged particles under the effects of external or self-consistent forces. The unknown f is the distribution function of particles in the phase space which depends on the time $t \geq 0$, the physical space $\mathbf{x} \in \mathbb{R}^d$ and the velocity $\mathbf{v} \in \mathbb{R}^d$, where d is the dimension $d = 1, 2, 3$. This kind of model can be used for the study of beam propagation, collisionless or gyrokinetic plasmas.

The numerical resolution of Vlasov type equations, the solution of which depends at least on six variables plus time, is performed most of the time using PIC (Particle In Cell) methods where the plasma is approached by a finite number of macro-particles (see [1]). Nevertheless it is well known that for some applications, the numerical noise inherent to the particle methods becomes too significant. Consequently, methods which discretize the Vlasov equation on a phase space grid have been proposed (see [3, 5, 9]). Among these Eulerian methods, the semi-Lagrangian method consists in computing directly the distribution function on a Cartesian grid of the phase space. The computation is done by integrating the characteristic curves backward at each time step and interpolating the value at the feet of the characteristics by some interpolation techniques (Lagrange, Hermite or cubic splines for example). We refer the

*INRIA Lorraine (CALVI Project), IECN Nancy and IRMA Strasbourg, France. crouseil@math.u-strasbg.fr

†IRMA Strasbourg and INRIA Lorraine (CALVI Project), France. gutnic@math.u-strasbg.fr

‡LSIIT Strasbourg and INRIA Lorraine (CALVI Project), France. latu@icps.u-strasbg.fr

§IRMA Strasbourg and INRIA Lorraine (CALVI Project), France. sonnen@math.u-strasbg.fr

reader to [9] for more details on the semi-Lagrangian method and to [5] for a comparison of Eulerian solvers dedicated to the Vlasov equation.

Eulerian methods, particularly well suited to trace halo formation generally badly described by PIC methods, have proven their efficiency on uniform meshes in two dimensional phase space, but when the dimensionality increases, the number of points on a uniform grid becomes very important which makes numerical simulations challenging. 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). Hence, methodologies have been recently implemented to avoid the increasing of points using uniform grid: such adaptive methods decrease the computational cost by keeping only a subset of all grid points. Such methods use moving distribution function grids well suited to manage data locality. For more details, we refer the reader to [2, 8].

This work is devoted to the comparison of two solvers that simulate the four-dimensional Vlasov equation. Even if the two methods are based on a semi-Lagrangian algorithm, the first one requires a uniform mesh whereas the second method is an adaptive method. The uniform method reconstructs the distribution function using local cubic splines which are well suited for parallel simulations. Indeed, the phase space domain is decomposed into patches; each patch is devoted to a processor and computes its own cubic spline coefficients by solving reduced linear systems. Finally, some boundary conditions are imposed at the boundary of the patches to get a $\mathcal{C}^1$ global solution (see section 3.1). This method may be particularly efficient for problems where low density regions are of crucial importance from the physical point of view, as in gyrokinetic simulations for example. In the adaptive method, the distribution function is decomposed in wavelet basis at different levels such that the coefficients called details are small whereas the function is regular. Therefore, according to a prediction procedure and a given threshold, only the significant details are computed. The distribution function can then be determined on the corresponding adaptive grid (see section 3.2). Adaptivity is particularly well suited when thin structures of the distribution function occur, as in particle beam applications or in laser-plasma interaction for example.

The model we consider throughout this paper is the scaled nonrelativistic four-dimensional Vlasov equation coupled self-consistently with the Poisson's equation

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = 0, \tag{1.1}$$

where $\mathbf{x} = (x, y)$, $\mathbf{v} = (v_x, v_y)$ and with initial condition $f(0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v})$. The self-consistent electric field is computed from Poisson's equations

$$-\Delta\phi = \rho(t, \mathbf{x}) = \int_{\mathbf{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}, \quad E(t, \mathbf{x}) = -\nabla_{\mathbf{x}} \phi(t, \mathbf{x}). \tag{1.2}$$

The magnetic field can be considered external and known, but can also be governed by the Maxwell equations. This system is standardly normalized for plasma applications using characteristic quantities as the plasma frequency ω_p in time, the Debye length in space and the electron thermal velocity v_{th} in the velocity space. For the normalization of the paraxial model used in particle beam applications, we refer to [6].

The rest of the paper is organized as follows: we first recall the basic principles of the semi-Lagrangian method applied to the Vlasov equation. Next, the algorithms of the two methods are described.

2 The semi-Lagrangian method for the Vlasov equation

The semi-Lagrangian method consists in computing approximation of the Vlasov equation (1.1) solution, on a phase space grid using the conservation property of the distribution function f along characteristics. More precisely, for any times s and t, we have

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

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

$$\frac{dX}{ds} = V, \qquad \frac{dV}{ds} = E(s, X(s)) + V(s) \times B(s, X(s)), \tag{2.1}$$

with initial conditions

$$X(t) = \mathbf{x}, \quad V(t) = \mathbf{v}.$$

From this property and f being known at time $t^n = n\Delta t$ where Δt is the time step, we can compute the distribution function at grid points $(\mathbf{x}_i, \mathbf{v}_j)$ at time t^{n+1} in the following way:

1. Find the starting point of the characteristic ending at $(\mathbf{x}_i, \mathbf{v}_j)$, *i.e.* $X^n = X(t^n; t^{n+1}, \mathbf{x}_i, \mathbf{v}_j)$ and $V^n = V(t^n; t^{n+1}, \mathbf{x}_i, \mathbf{v}_j)$.

2. Compute $f(t^n, X^n, V^n)$ by interpolation, f being known only at mesh points at time t^n.

In order to deal with the first step, a second order time discretization of (2.1) can be performed using a time splitting procedure (see [3]), which allows the explicit solve of the characteristics. To treat the second step, an interpolation operator has to be chosen. Even if the semi-Lagrangian method does not require any specific interpolation scheme, numerical experiences dictate the use of a high enough order method so that diffusion is limited. Cubic splines or Lagrange polynomials for example present some interesting properties, and seem to be a good compromise between a small diffusivity and a low computational cost. The semi-Lagrangian method also permits the use of a different interpolation grid for each time step, which is the basis of adaptive methods.

3 Two numerical methods for the Vlasov equation

3.1 The local spline interpolation

In this section, we present our interpolation technique based on a cubic spline method (see [9]). Even if the cubic spline approach is quite standard for solving Vlasov equations, it remains a global method since it requires the values of the distribution function on all the domain, which is an inconvenient from a parallelization point of view. Our approach avoids this globality. Indeed, we decompose the phase space into several patches, each patch being devoted to one processor. The strategy is based on adapted boundary conditions which allow to get a global continuously derivable reconstructed solution.

Let us consider a function f which is defined on a global domain $[x_{\min}, x_{\max}] \subset \mathbb{R}$. This domain is decomposed into several subdomains called generically $[x_0, x_N]$; each subdomain will be devoted to a processor. In the following, we will use the notation $x_i = x_0 + ih$, where h is the mesh size: $h = (x_N - x_0)/(N+1)$.

Let us now restrict the study of $f : x \to f(x)$ on the generic interval $[x_0, x_N]$, $N \in \mathbb{N}$, where x_0 and x_N are to be chosen, according to the decomposition domain. The projection s of f onto the cubic spline basis reads

$$f(x) \simeq s(x) = \sum_{\nu=-1}^{N+1} \eta_\nu B_\nu(x),$$

where B_ν is the cubic B-spline

$$B_\nu(x) = \frac{1}{6h^3} \begin{cases} (x - x_{\nu-2})^3 & \text{if } x_{\nu-2} \le x \le x_{\nu-1}, \\ h^3 + 3h^2(x - x_{\nu-1}) + 3h(x - x_{\nu-1})^2 & \\ \quad -3(x - x_{\nu-1})^3 & \text{if } x_{\nu-1} \le x \le x_\nu, \\ h^3 + 3h^2(x_{\nu+1} - x) + 3h(x_{\nu+1} - x)^2 & \\ \quad -3(x_{\nu+1} - x)^3 & \text{if } x_\nu \le x \le x_{\nu+1}, \\ (x_{\nu+2} - x)^3 & \text{if } x_{\nu+1} \le x \le x_{\nu+2}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The interpolating spline s is uniquely determined by $(N+1)$ interpolating conditions

$$f(x_i) = s(x_i), \quad \forall i = 0, ..., N, \tag{3.2}$$

and the boundary conditions on the first derivative at both ends of the interval in order to obtain a $\mathcal{C}^1$ global approximation

$$f'(x_0) \simeq s'(x_0), \quad f'(x_N) \simeq s'(x_N). \tag{3.3}$$

The only cubic B-spline not vanishing at point x_i are $B_{i\pm1}(x_i) = 1/6$ and $B_i(x_i) = 2/3$. Hence (3.2) yields

$$f(x_i) = \frac{1}{6}\,\eta_{i-1} + \frac{2}{3}\,\eta_i + \frac{1}{6}\,\eta_{i+1}, \quad i = 0, ..., N. \tag{3.4}$$

On the other hand, we have $B'_{i\pm1}(x_i) = \pm 1/(2h)$ and $B'(x_i) = 0$. Thus the boundary conditions (3.3) become

$$f'(x_0) \simeq s'(x_0) = -1/(2h)\,\eta_{-1} + 1/(2h)\,\eta_1, \text{ and } f'(x_N) \simeq s'(x_N) = -1/(2h)\,\eta_{N-1} + 1/(2h)\,\eta_{N+1}. \tag{3.5}$$

Finally, $\eta = (\eta_{-1}, ..., \eta_{N+1})^T$ is the solution of the $(N+3) \times (N+3)$ system $A\eta = F$, where F is the following vector

$$F = \left[f'(x_0), f(x_0), ..., f(x_N), f'(x_N)\right]^T, \tag{3.6}$$

and A denotes the following fixed matrix

$$A = \frac{1}{6}\begin{pmatrix} -3/h & 0 & 3/h & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & & \vdots \\ 0 & 1 & 4 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & -3/h & 0 & 3/h \end{pmatrix}. \tag{3.7}$$

The resolution of this linear system benefits from the special structure of the matrix A; indeed, its LU decomposition can be computed once for all, and consequently, the resolution of the system $A\nu = F$ is linear with respect to the number of unknowns.

Let us focus now on the approximation of f'. To get accurate numerical results, boundary conditions need a special attention. Indeed, in being as closer as possible to the sequential code, the value of the derivatives has to be very close to the approximation (3.3) given by the global resolution. To that purpose, we perform a new approximation derived using the spline coefficients; we refer the reader to [4] for the details of the derivation, but the final approximation requires 21 points centered at the boundary point.

3.2 The adaptive method

In this method, our purpose is to be able of refined or derefined the phase space grid adaptively in time, which will be updated using a multi-resolution technique. We use interpolating wavelets based on Lagrange polynomial interpolation of odd degree which provide a natural way to extend the semi-Lagrangian procedure to an adaptive grid (see [8] for more details).

In 1D, we represent the distribution function on a wavelet basis functions at different scales. Denoting by φ_k^j the basis functions at level j such that $\varphi_k^j(x_k^j) = 1$ and $\varphi_k^j(x_l^j) = 0$ if $l \neq k$, we can write the grid distribution function f^j taking value c_k^j at grid point x_k^j as

$$f^j(x) = \sum_k c_k^j \, \varphi_k^j(x).$$

Then the framework of interpolating wavelets allows us to equivalently define a grid function f_{j+1} defined at a finer level $j+1$ on a grid G_{j+1} as

$$f^{j+1}(x) = \sum_k c_k^{j+1} \, \varphi_k^{j+1}(x) = \sum_k c_k^j \, \varphi_k^j(x) + \sum_k d_k^j \, \varphi_k^j(2x-1),$$

either by the sequence $(c_k^{j+1})_{k\in\mathbb{Z}}$ or by the sequences $(c_k^j)_{k\in\mathbb{Z}}$ where $c_{2k}^{j+1} = c_k^j$ and $(d_k^j)_{k\in\mathbb{Z}}$ where

$$d_k^j = c_{2k+1}^{j+1} - P_{2N-1}(x_{2k+1}^{j+1}) = c_{2k+1}^{j+1} - \sum_{n=1-N}^{N} \alpha_n \, c_{k+n}^j = c_{2k+1}^{j+1} - \sum_{n=1-N}^{N} \alpha_n \, c_{2k+2n}^{j+1},$$

where P_{2N-1} is the Lagrange interpolating polynomial of degree $N-1$. We can then compress it by eliminating detail coefficients d_k^j which are naturally small where the distribution function is regular, and accordingly remove the associated grid points.

The four-dimensional description of f is done using a tensor product representation. Defining the index set $\mathcal{I}_4 = \{\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \in \{0,1\}^4\}$, the distribution function is approximated, in a hierarchical approximation from a coarser level j_0 to a finer level j_1, by

$$f(x, y, v_x, v_y) = \sum_{\mathcal{K}} c_{\mathcal{K}}^{j_0,0}\, \phi_{\mathcal{K}}^{j_0,0}(x, y, v_x, v_y) + \sum_{j=j_0}^{j_1} \sum_{\mathcal{K}} \sum_{\substack{\alpha \in \mathcal{I} \\ \alpha \neq 0}} d_{\mathcal{K}}^{j,\alpha}\, \phi_{\mathcal{K}}^{j,\alpha}(x, y, v_x, v_y), \tag{3.8}$$

where we have $\phi_{\mathcal{K}}^{j,\alpha}(x, y, v_x, v_y) = \varphi_{k_1}^{j,\alpha_1}(x)\, \varphi_{k_2}^{j,\alpha_2}(y)\, \varphi_{k_3}^{j,\alpha_3}(v_x)\, \varphi_{k_4}^{j,\alpha_4}(v_y)$ with $\varphi_k^{j,0}(\cdot) = \varphi(2^j \cdot -k)$ and $\varphi_k^{j,1}(\cdot) = \varphi(2^j(2 \cdot -1) - k)$ for any k. Here φ denotes the scaling function at the coarsest level such that $\varphi_k^j(\cdot) = \varphi(2^j \cdot -k)$ for any k and j. From (3.8), we can compute the compressed distribution function keeping only the significant details $d_{\mathcal{K}}^{j,\alpha}$ and built the corresponding four-dimensional phase space grid.

This formula also allows us to compute adaptively the space charge density ρ in order to solve Poisson's equation (1.2) as well as the moments of order 1 and 2 in x, y, v_x and v_y in order to compute the RMS quantities needed for diagnostics. This considerably reduces the number of operations needed as it uses only points of the adaptive grid.

The numerical algorithm is then initialize by creating an initial adaptive grid by construction of a hierarchical approximation of the initial distribution f_0 and remove the grid points corresponding to coefficients less than a specific threshold.

The time loop is based on compression-prediction step and a backward advection step. We first compress the distribution function f as for f_0 and built the corresponding adaptive grid. Then we predict the adaptive grid at next time step by transporting the grid points along the particle orbits and adding all the grid points around the advanced points, in a square of width the size of the scaling function at this level, at the given level and one level finer.

In the second step, for each new grid point, we follow the characteristics backward and compute by interpolation the value of the distribution function at its origin from the old values, and therefore obtain the new values on the predicted grid. Note that, the predicted grid is at the end enhanced so that it becomes a well formed tree needed to perform the wavelet decomposition. This is an additional step compared to the usual semi-Lagrangian method.

Actually, we use a full time-splitting to only compute one-dimensional advections in x, y, v_x and v_y directions, which is inspired by the idea of Strang (in a semi-Lagrangian framework, see e.g. [3]) in order to keep the method second order in time.

Adaptivity allows us to significantly reduce the memory needed as well as the computational time. Note that we also have optimized the data structure in order to still be more efficient and in the purpose of parallelization (see e.g. [7]).

References

[1] C.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, Institute of Physics Publishing, Bristol and Philadelphia, 1991.

[2] M. Campos-Pinto, M. Merhenberger, *Adaptive numerical resolution of the Vlasov equation*, Numerical Methods for Hyperbolic and Kinetic Equation, (2004).

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

[4] N. Crouseilles, G. Latu, E. Sonnendrücker, *Hermite spline interpolation on patches for a parallel solving of the Vlasov-Poisson equation*, Rapport de Recherche INRIA-Lorraine, n° 5926, (2006).

[5] F. Filbet, E. Sonnendrücker, *Comparison of Eulerian Vlasov solvers*, Comput. Phys. Comm., **151**, pp. 247-266, (2003).

[6] F. Filbet, E. Sonnendrücker, *Modeling and numerical simulation of space charge dominated beams in the paraxial approximation*, Math. Mod. Meth. App. Sc., **16**, pp.1-29, (2006).

[7] M. Gutnic, M. Haefele, G. Latu, *A parallel Vlasov solver using a wavelet based adaptive mesh refinement*, in: 2005 International Conference on Parallel Processing (ICPP'2005), 7th Workshop on High Perf. Scientific and Engineering Computing, IEEE Computer Society Press, pp. 181-188 (2005).

[8] M. Gutnic, M. Haefele, I. Paun, E. Sonnendrücker, *Vlasov simulation on an adaptive phase space grid*, Comput. Phys. Comm., **164**, pp. 214-219, (2004).

[9] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, *The semi-Lagrangian method for the numerical resolution of the Vlasov equations*, J. Comput. Phys., **149**, pp. 201-220, (1999).