Adaptive Vlasov simulations of intense beams

Eric Sonnendrücker*, Michael Gutnic*, Matthieu Haefele† and Jean-Louis Lemaire**

*IRMA, Université Louis Pasteur Strasbourg and Projet CALVI, INRIA Lorraine
†LSIIT, Université Louis Pasteur Strasbourg and Projet CALVI, INRIA Lorraine
**CEA, Bruyères-Le-Châtel

Abstract. Most simulations of intense particle beams are performed nowadays using Particle In Cell (PIC) techniques. Direct grid based Vlasov methods have also been used but mostly for 1D simulations as they become very costly in higher dimensions when using uniform phase space grids. We have recently introduced adaptive mesh refinement techniques that allow us to automatically concentrate the grid points at places where the distribution function is varying most. In this paper we shall introduce this technique and show how it can be used to improve the efficiency of grid based Vlasov solvers.

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. On the other hand they suffer from numerical diffusion and 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, all the more when the dimension of the simulated phase space gets higher.

In order to optimize the number of necessary grid points for our phase space grids we investigate adaptive grid techniques which enable us to keep only the grid points which are really necessary to achieve a given error tolerance. The use of such adaptive grid techniques should make the computational complexity of grid-based solver more comparable to that of particle methods and thus render this technique a serious alternative to PIC simulations.

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

$$\frac{\partial f}{\partial z} + \frac{\mathbf{v}}{v_b} \cdot \nabla_{\mathbf{x}} f + \frac{q}{\gamma_b v_b m} (E + \mathbf{v} \times B) \cdot \nabla_{\mathbf{v}} f = 0, \quad (1)$$

with initial condition $f(\mathbf{x},\mathbf{v},0) = f_0(\mathbf{x},\mathbf{v})$. Here v_b denotes the beam velocity and $\gamma_b = (1-(v_b/c)^2)^{-1/2}$. The self electric field $\mathbf{E}$ is computed from Poisson's equations

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

The magnetic field is external and considered to be known.

The paper is organized as follows: We first recall the semi-Lagrangian method which is the underlying method, then present the idea of hierarchical approximation of a function. We then show how this idea can be applied in the setting of the semi-Lagrangian method to get an adaptive grid. And finally, we present some numerical simulation of beams in transverse phase-space using the adaptive methods.

BUILDING AN ADAPTIVE GRID VLASOV SOLVER

The semi-Lagragian method, first introduced for the Vlasov equation in [1], is based on the feature of the Vlasov equation that the particle distribution function is conserved along the particle trajectories. Hence if (x_n, v_n) and (x_{n+1}, v_{n+1}) are the positions of the same particle following the Lorentz equations of motion at times $t_n = n\Delta t$ and $t_{n+1} = (n+1)\Delta t$ respectively. Then the distribution

function f satisfies $f(t_n, x_n, v_n) = f(t_{n+1}, x_{n+1}, v_{n+1})$. This property is used to construct the semi-Lagrangian numerical method which computes the distribution function on a grid of phase space using the following two steps:

1. Compute the positions (x_n, v_n) at time t_n of the particles ending at the phase space grid points (x_i, v_j) at time t_{n+1}.
2. Interpolate the old values f_n of the distribution function at the origins of the trajectories. Then, using the conservation property $f_{n+1}(x_i, v_j) = f_n(x_n, v_n)$.

In order for this method to become adaptive, we shall try and find a minimal set of interpolation points depending on the actual values of f_n to get an approximation of $f(t_n)$ within some given accuracy criterion. Then if such a set of points can be predicted for f_{n+1}, only the trajectories of the particles ending at these points will need to be computed. As this set of points will be considerably smaller than the number of points on the uniform fine grid, we shall be able to reduce the computation time considerably.

Hierarchical approximation

In order to optimize the number of grid points used for the interpolation step of the semi-Lagrangian method, we construct a hierarchical approximation of a function using a sequence of nested grids.

Let us describe this construction precisely in the 1D case. The extension to several dimensions is straightforward using a tensor product construction.

For any integer value of j, we consider a uniform grid G^j of step 2^{-j}. The grid points are located at $x_k^j = k2^{-j}$. This defines a sequence of grids that we denote by $(G^j)_{j_0 \le j \le j_1}$, and j will be called the level of the grid.

To compare an approximation defined on a coarse grid, say G^j, with an approximation defined on the finer grid G^{j+1}, we need a way to extend grid functions defined on G^j to all the points of G^{j+1}. This can be done by using a Lagrange interpolation of odd degree (to keep symmetry). Let us denote by (c_k^j) the values of a function f^j defined on the grid G^j at the grid points $k2^{-j}$. Then f^j can be defined at the midpoints of the cells of G^j using Lagrange interpolation of odd degree $2N-1$

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

where for example $\alpha_0 = \alpha_1 = 1/2$ for linear interpolation $(N = 1)$ and $\alpha_{-1} = \alpha_2 = -1/16$ and $\alpha_0 = \alpha_1 = 9/16$ for cubic interpolation $(N = 2)$. This procedure can be repeated successively on the mid-points of the cells of the finer grid and thus enables us to define a value for a function defined on grid G^j on the whole hierarchy of nested grids.

Now, as we have defined a procedure to extend a coarse grid approximation on a finer grid, we can also define a projection of the fine grid values to the coarse grid. The simplest way is to just consider a restriction. We can then compare the coarse grid approximation with the fine grid approximation. Let us denote by f_{j+1} the approximation defined on G_{j+1}. It is characterized by the values (c_k^{j+1}) of the function at the grid points of G_{j+1}. In the same way f_j is the approximation defined on G_j which is characterized by its values c_k^j at the grid points of G_j. Note that only the even points of G_{j+1} also exist on G_j. The odd points do not exist but can be computed using our interpolation procedure. Hence we have $c_{2k}^{j+1} = c_k^j$ and the difference of the predicted value on G^j and the actual value on G^{j+1} at the next odd point will be denoted by

$$\begin{aligned} d_k^j &= c_{2k+1}^{j+1} - P_{2N-1}(x_{2k+1}^{j+1}) \\ &= c_{2k+1}^{j+1} - \sum_{n=1-N}^{N} \alpha_n c_{k+n}^j = c_{2k+1}^{j+1} - \sum_{n=1-N}^{N} \alpha_n c_{2k+2n}^{j+1}, \end{aligned} \tag{3}$$

where P_{2N-1} stands for the centered Lagrange interpolation polynomial of degree $2N-1$. It is easy to see that the (c_k^{j+1}) on the one hand and the (c_k^j) and the (d_k^j) on the other hand convey the same information expressed differently as formula (3) enables to go from the (c_k^{j+1}) to the (c_k^j) and the (d_k^j) and the formula can be inverted in the following way

$$c_{2k}^{j+1} = c_k^j, \quad c_{2k+1}^{j+1} = d_k^j + \sum_{n=1-N}^{N} \alpha_n c_{k+n}^j,$$

to go the other way.

This procedure enables us to define a hierarchical approximation of a function. Indeed, the (c_k^j) define a coarse grid uniform approximation of f, the (c_k^{j+1}) a fine grid uniform approximation of f and the (d_k^j) carry the missing piece to go from the coarse grid approximation to the fine grid approximation.

The differences d_k^j will be small at points where the prediction using the values of the coarser grid do a good job. This is the idea that we shall use. We shall keep the values at the midpoints of the finer grid only where they cannot be predicted within a given accuracy from the coarser grid. Note that when a coefficient d_k^j which small is set to zero, we lose a small amount of particles, which is clearly unphysical. However, we can remedy this problem by modifying the c_k^j coefficients such that

they carry all the mass. In this case after computing d_k^j, we set

$$\tilde{c}_k^j = c_k^j + \frac{1}{4}(d_k^j + d_{k+1}^j).$$

This technique can also be used to conserve any number of moments. See [2] for more details.

In contrast to a uniform approximation where all coefficients cary similar information, in the hierarchical approximation the coefficients corresponding to finer grid points only carry a correction to the approximation on the coarse grid points, which is small when the approximation on the coarse grid is already good enough. This gives us an indication on which grid points, we really need to keep, indeed only those where the coefficient is larger than some given threshold. Hence hierarchical approximation gives us a way to approximate a function at the expense of only a small additional error using considerably fewer grid points. The method we have described can be cast in the framework of biorthogonal wavelets which can be used for a mathematical analysis of the method.

An adaptive algorithm for the Vlasov equation

The hierarchical approximation described before enables us to construct an adaptive grid based Vlasov solver using a lot fewer grid points than a uniform grid. The idea is the following (the full algorithm is described in [3]). First create an initial adaptive grid by construction a hierarchical approximation of the initial distribution f_0 and remove the grid points corresponding to coefficients less than the threshold. Then the time loop is based on the following steps: Predict the adaptive grid for the new time step by transporting the grid points along the particle orbits and adding one level of grid points to allow refinement. After that, compute the new value of the distribution function at the predicted grid points and perform the hierarchical decomposition to remove again unneeded grid points.

Numerical results

We validate our method on a simulation of a beam of potassium ions in a periodic focusing channel. The applied focusing field is of the form $\alpha(1+\cos 2\pi z/S)$, the beam energy is $80\,keV$ and the tune depression is 0.17. A snapshot of the distribution is given in figure 1 and the corresponding adaptive grid in figure 2. One can see that the grid points are distributed mostly in areas where the distribution function is varying.

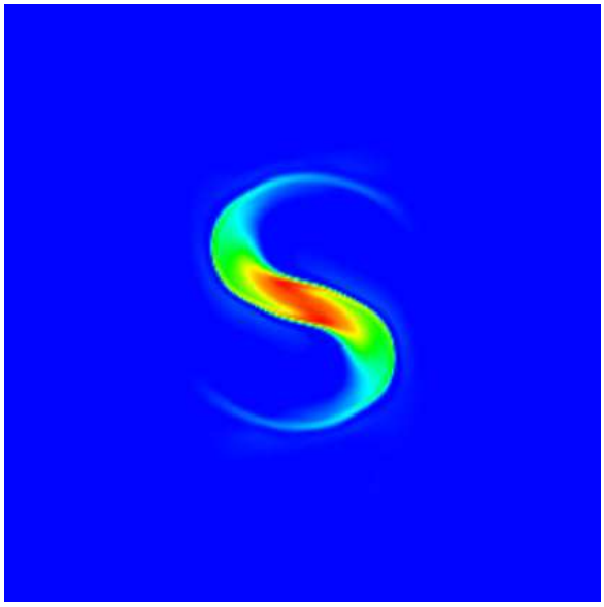

FIGURE 1. Snapshot of the distribution function

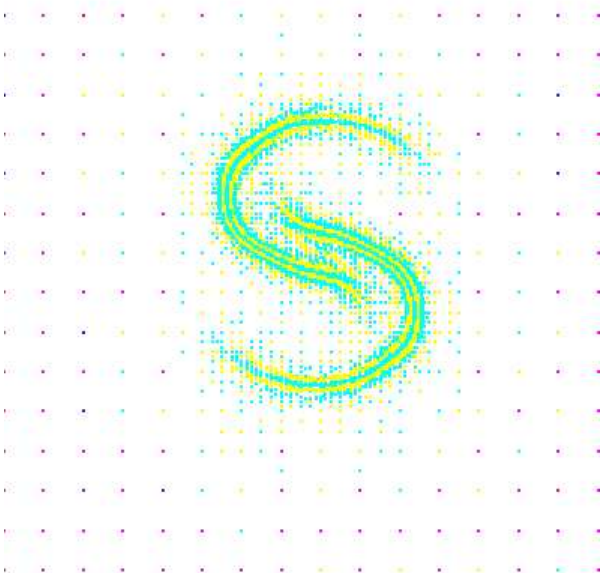

FIGURE 2. Adaptive grid

REFERENCES

1. Sonnendrücker, E., Roche, J., Bertrand, P., and Ghizzo, A., *J. Comput. Phys.*, **49**, 201–220 (1999).
2. Gutnic, M., Haefele, M., Paun, I., and Sonnendrücker, E., "An adaptive semi-Lagrangian method for the Vlasov equation based on a multiresolution analysis with moments conservation," in *Proceedings of the ICAP conference, Saint-Petursburg 2004*, 2004.
3. Gutnic, M., Haefele, M., Paun, I., and Sonnendrücker, E., *Comput. Phys. Comm.*, **164**, 214–219 (2004).