



**HAL**  
open science

# Two-scale semi-lagrangian simulation of a charged particle beam in a periodic focusing channel

Alexandre Mouton

► **To cite this version:**

Alexandre Mouton. Two-scale semi-lagrangian simulation of a charged particle beam in a periodic focusing channel. *Kinetic and Related Models*, 2009, 2 (2), pp.251-274. 10.3934/krm.2009.2.251 . hal-00348631

**HAL Id: hal-00348631**

**<https://hal.science/hal-00348631>**

Submitted on 19 Dec 2008

**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.

# Two-scale semi-lagrangian simulation of a charged particle beam in a periodic focusing channel

*Alexandre Mouton*

IRMA, Université Louis Pasteur, F-67084 Strasbourg

## Abstract

This paper is devoted to numerical simulation of a charged particle beam submitted to a strong oscillating electric field. For that, we consider a two-scale numerical approach as follows: we first recall the two-scale model which is obtained by using two-scale convergence techniques; then, we numerically solve this limit model by using a backward semi-lagrangian method and we propose a new mesh of the phase space which allows us to simplify the solution of the Poisson's equation. Finally, we present some numerical results which have been obtained by the new method, and we validate its efficiency through long time simulations.

**AMS subjects:** 35B27, 76X05, 65M25 (65Y20).

**Keywords:** two-scale convergence, semi-lagrangian method, Vlasov-Poisson model.

## 1 Introduction

Recent papers have proved that the two-scale convergence theory developed by Allaire in [2] and Nguetseng in [17] can be used successfully in order to develop numerical methods for solving ODEs or PDEs with oscillatory singular perturbations: for example, Frénod, Salvarani and Sonnendrücker have developed a two-scale PIC method in [13] for simulations of charged particle beams in a periodic focusing channel, and Frénod, Mouton and Sonnendrücker have developed a two-scale finite volume method in [9] for solving the weakly compressible 1D Euler equations. We can also cite the work of Ailliot, Frénod and Monbet in [1] about the simulation of tide oscillation for long term drift forecast of objects in the ocean.

On the other hand, many papers devoted to the numerical simulation of Vlasov-type problems involve Particle-In-Cell methods (see Birdsall and Langdon [3]) or Eulerian methods like semi-lagrangian schemes (see Sonnendrücker *et al.*[19], Grandgirard *et al.*[14], Filbet and Sonnendrücker [7, 8], Cheng and Knorr [5]). Since papers like [9] or [13] can be viewed as parts of a work programme which goal is the development of two-scale numerical methods for simulations of magnetic confinement fusion, it can be interesting to couple two-scale convergence results on Vlasov models such as two-scale models obtained in [12] with a semi-lagrangian scheme.

However, it is preferable in a first time to develop such a method on a more simple problem in order to study its behavior especially in the context of non-smooth solutions. The context of charged particle beams in a periodic focusing external field described in [13] offers a relatively simple framework for answering these questions.

We recall that such a phenomenon can be successfully represented by the 3D Vlasov-Maxwell system. In the same spirit of [13], we will consider non-relativistic long and thin beams, so we can replace the full three-dimensional Vlasov-Maxwell system by its paraxial approximation. To obtain this approximation, we do the following assumptions:

- the beam has already reached its stationary state,
- the beam is long and thin,
- the beam is propagating at constant velocity  $v_z$  along the longitudinal axis  $z$ ,
- the beam is sufficiently long so we can neglect the longitudinal self-consistent forces,

- the external electric field is supposed to be  $l$ -periodic in  $z$  and independent of the time,
- the beam is axisymmetric,
- the initial distribution  $f_0$  is concentrated in angular momentum.

Under all these assumptions, the 3D Vlasov-Maxwell system reduces itself to a 2D Vlasov-Poisson system where the longitudinal position coordinate  $z$  plays the role of time, and even to a 1D axisymmetric Vlasov-Poisson system of the form

$$\begin{aligned}
\partial_t f^\epsilon + \frac{v_r}{\epsilon} \partial_r f^\epsilon + (E_r^\epsilon + \Xi_r^\epsilon) \partial_{v_r} f^\epsilon &= 0, \\
f^\epsilon(r, v_r, t = 0) &= f_0(r, v_r), \\
\frac{1}{r} \partial_r (r E_r^\epsilon) &= \int_{\mathbb{R}} f^\epsilon dv_r, \\
\Xi_r^\epsilon(r, t) &= -\frac{1}{\epsilon} H_0 r + H_1 \left( \omega_1 \frac{t}{\epsilon} \right) r,
\end{aligned} \tag{1.1}$$

where  $r \geq 0$  is the radial component of the position vector in the transverse plane to the propagation direction,  $v_r \in \mathbb{R}$  is the projection of the transverse velocity in the transverse plane to the propagation direction,  $\epsilon$  is the ratio between the characteristic transverse radius of the beam and the characteristic longitudinal length of the beam,  $f^\epsilon = f^\epsilon(r, v_r, t)$  is the distribution function of the particles,  $E_r^\epsilon = E_r^\epsilon(r, t)$  is the radial part of the transverse self-consistent electric field,  $\Xi_r^\epsilon = \Xi_r^\epsilon(r, t)$  is the radial part of the transverse external electric field defined with the dimensionless real constant  $\omega_1$  and the tension functions  $H_0$  and  $H_1$  which are respectively constant and periodic. All these quantities and variables are dimensionless. This system is naturally defined for  $r \geq 0$  but we can extend it to  $r \in \mathbb{R}$  by using the conventions  $f^\epsilon(-r, -v_r, t) = f^\epsilon(r, v_r, t)$ ,  $E_r^\epsilon(-r, t) = -E_r^\epsilon(r, t)$ , and  $\Xi_r^\epsilon(-r, t) = -\Xi_r^\epsilon(r, t)$ .

The aim of the paper is to simulate the system (1.1) with a two-scale semi-lagrangian scheme when  $\epsilon \rightarrow 0$ . Inspired by Frénod, Salvarani and Sonnendrücker [13], and Frénod, Mouton and Sonnendrücker [9], we derive the model (1.1) by using the two-scale convergence theory developed in Allaire [2] and Nguetseng [17] and we obtain a new model which is independent of  $\epsilon$ . Then, instead of discretizing the model (1.1) with a classical semi-lagrangian scheme, we discretize the new model with a semi-lagrangian method in order to obtain an approximation of a function  $F = F(r, v_r, \tau, t)$ , where  $\tau$  is the second time scale, which verifies  $f^\epsilon(r, v_r, t) \sim 2\pi F(r, v_r, \frac{t}{\epsilon}, t)$ . Proceeding in such a way presents the advantage that there is no longer  $\frac{1}{\epsilon}$ -oscillations in the limit model, so a very small time step is no longer required in order to simulate these oscillations. However, since semi-lagrangian schemes are based on interpolation on a phase space mesh, we have to pay attention to the contributions of the second time scale  $\tau$  in the limit model. For this reason, in the same spirit of the work of Lang *et al.*[16], we introduce a  $\tau$ -dependent moving mesh adapted for this two-scale numerical method the goal of which is to reduce the number of interpolations during the simulation.

The paper is organized as follows: in section 2, we recall the procedure leading from the paraxial approximation of the complete 3D Vlasov-Maxwell system to the model (1.1), and we will recall some two-scale convergence results about this model. In section 3, we build the semi-lagrangian method on the limit model and we see how to simplify it by considering a particular mesh. Section 4 is devoted to numerical results obtained with the two-scale semi-lagrangian method: on one hand, we compare them to some results obtained from a classical semi-lagrangian method on the system (1.1) in terms of quality of results and CPU time, and on the other hand, we see in this section the consequences of the use of the new mesh in the same terms.

## 2 The two-scale model

Firstly, we recall in this section the way to obtain the system (1.1) from the paraxial approximation of the 3D Vlasov-Maxwell equations. Then, we present a theorem about the two-scale convergence

of the solutions of (1.1) which has been proved by Frénod, Salvarani and Sonnendrücker in [13].

## 2.1 Scaling of the paraxial model

By applying the hypothesis about the considered beam which are mentioned in the introduction, the full three-dimensional Vlasov-Maxwell is reduced to

$$\begin{aligned} v_z \partial_z f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m} (\mathbf{E} + \boldsymbol{\Xi}) \cdot \nabla_{\mathbf{v}} f &= 0, \\ f(\mathbf{x}, z = 0, \mathbf{v}) &= f_0(\mathbf{x}, \mathbf{v}), \\ -\nabla_{\mathbf{x}} \phi &= \mathbf{E}, \quad -\Delta_{\mathbf{x}} \phi = \frac{\rho}{\varepsilon_0} = \frac{q}{\varepsilon_0} \int_{\mathbb{R}^2} f d\mathbf{v}, \end{aligned} \quad (2.1)$$

where  $z$  is the longitudinal position coordinate,  $\mathbf{x} = (x, y)$  is the transverse position vector,  $v_z$  is the constant longitudinal speed,  $\mathbf{v} = (v_x, v_y)$  is the transverse speed vector,  $f = f(\mathbf{x}, z, \mathbf{v})$  is the distribution function of particles whose charge is  $q$  and mass is  $m$ ,  $\phi = \phi(\mathbf{x}, z)$  is the potential linked with the transverse self-consistent electric field  $\mathbf{E} = \mathbf{E}(\mathbf{x}, z)$ , and  $\boldsymbol{\Xi} = \boldsymbol{\Xi}(\mathbf{x}, z)$  is the transverse external electric field. More details about this derivation can be found in [6] and [8]. In this model, we assume that the external electric field is given by

$$\boldsymbol{\Xi}(\mathbf{x}, z) = -H_0 \mathbf{x} + H_1 \left( \omega_1 \frac{z}{l} \right) \mathbf{x}, \quad (2.2)$$

where  $H_0$  is a positive constant tension,  $H_1$  is a  $l$ -periodic tension and  $\omega_1$  is a dimensionless real constant.

In the same spirit of Frénod, Salvarani and Sonnendrücker [13], we build a dimensionless version of the system (2.1)-(2.2) by introducing the dimensionless variables  $\mathbf{x}'$ ,  $z'$ ,  $\mathbf{v}'$  defined by

$$\mathbf{x} = \lambda \mathbf{x}', \quad z = L z', \quad \mathbf{v} = v_z \mathbf{v}', \quad (2.3)$$

where  $\lambda$  is the characteristic transverse radius of the beam and  $L$  is the characteristic length of the beam. Moreover, we define the dimensionless quantities  $f'$ ,  $f'_0$ ,  $\mathbf{E}'$ ,  $\phi'$ ,  $H'_0$ ,  $H'_1$  by

$$\begin{aligned} f(\lambda \mathbf{x}', L z', v_z \mathbf{v}') &= \frac{m \varepsilon_0}{q^2 \lambda L} f'(\mathbf{x}', z', \mathbf{v}'), & \mathbf{E}(\lambda \mathbf{x}', L z') &= \frac{m v_z^2}{q L} \mathbf{E}'(\mathbf{x}', z'), \\ f_0(\lambda \mathbf{x}', v_z \mathbf{v}') &= \frac{m \varepsilon_0}{q^2 \lambda L} f'_0(\mathbf{x}', \mathbf{v}'), & H_0 &= \overline{H_0} H'_0, \\ \phi(\lambda \mathbf{x}', L z') &= \frac{m \lambda v_z^2}{q L} \phi'(\mathbf{x}', z'), & H_1(\tau) &= \overline{H_1} H'_1(\tau). \end{aligned} \quad (2.4)$$

With these new variables, the system (2.1)-(2.2) can be rewritten under the form

$$\begin{aligned} \partial_{z'} f' + \frac{L}{\lambda} \mathbf{v}' \cdot \nabla_{\mathbf{x}'} f' + \left[ \mathbf{E}' - \frac{\overline{H_0} \lambda q L}{v_z^2 m} H'_0 \mathbf{x}' + \frac{\overline{H_1} \lambda q L}{v_z^2 m} H'_1 \left( \frac{\omega_1 L z'}{l} \right) \mathbf{x}' \right] \cdot \nabla_{\mathbf{v}'} f' &= 0, \\ f'(\mathbf{x}', z' = 0, \mathbf{v}') &= f'_0(\mathbf{x}', \mathbf{v}'), \\ -\nabla_{\mathbf{x}'} \phi' &= \mathbf{E}', \quad -\Delta_{\mathbf{x}'} \phi' = \int_{\mathbb{R}^2} f' d\mathbf{v}'. \end{aligned} \quad (2.5)$$

Since the beam is supposed to be long and thin, it is natural to take the ratio

$$\frac{\lambda}{L} = \epsilon. \quad (2.6)$$

Furthermore, as we want to simulate the beam over a large number of periods of the external electric field, we also consider the ratio

$$\frac{l}{L} = \epsilon. \quad (2.7)$$

Finally, we suppose that the external electric field is much stronger than the self-consistent electric field and that its oscillations in  $z$  direction are of the same order as  $\mathbf{E}$ , so we consider that

$$\frac{\overline{H_0} \lambda q L}{v_z^2 m} = \frac{1}{\epsilon}, \quad \frac{\overline{H_1} \lambda q L}{v_z^2 m} = 1. \quad (2.8)$$

Under all these hypothesis the system (2.5) reduces to

$$\begin{aligned} \partial_t f^\epsilon + \frac{1}{\epsilon} \mathbf{v} \cdot \nabla_{\mathbf{x}} f^\epsilon + (\mathbf{E}^\epsilon + \mathbf{\Xi}^\epsilon) \cdot \nabla_{\mathbf{v}} f^\epsilon &= 0, \\ f^\epsilon(\mathbf{x}, \mathbf{v}, t = 0) &= f_0(\mathbf{x}, \mathbf{v}), \\ -\nabla_{\mathbf{x}} \phi^\epsilon &= \mathbf{E}^\epsilon, \quad -\Delta_{\mathbf{x}} \phi^\epsilon = \int_{\mathbb{R}^2} f^\epsilon d\mathbf{v}, \\ \mathbf{\Xi}^\epsilon(\mathbf{x}, t) &= -\frac{1}{\epsilon} H_0 \mathbf{x} + H_1 \left( \frac{\omega_1 t}{\epsilon} \right) \mathbf{x}, \end{aligned} \quad (2.9)$$

where the primed notations for the variables and the initial distribution have been eliminated, and where  $z'$  has been replaced by  $t$ ,  $f'$  by  $f^\epsilon$ ,  $\mathbf{E}'$  by  $\mathbf{E}^\epsilon$ ,  $\phi'$  by  $\phi^\epsilon$ , and  $\mathbf{\Xi}'$  by  $\mathbf{\Xi}^\epsilon$ .

We introduce the polar coordinates  $(r, \theta, v_r, v_\theta)$  linked with  $(\mathbf{x}, \mathbf{v})$  by the relations

$$\begin{aligned} x &= r \cos \theta, & v_r &= v_x \cos \theta + v_y \sin \theta, \\ y &= r \sin \theta, & v_\theta &= v_y \cos \theta - v_x \sin \theta. \end{aligned} \quad (2.10)$$

Since the beam is supposed to be axisymmetric, the system does not depend on  $\theta$ . Furthermore, we assume that the initial distribution is concentrated in angular momentum. Then the system (2.9) reduces to (1.1).

## 2.2 Two-scale convergence results

Since the aim of the paper is to develop a two-scale numerical method in order to simulate the model (1.1), we have to establish that the solution  $f^\epsilon$  of this model two-scale converges to a function  $F = F(r, v_r, \tau, t)$  in a certain Banach space, and we have to find a system of equations verified by  $F$ . These results have been proved in [13] and are recalled in the theorem below.

**Theorem 1.** *We assume that  $H_0 = 1$  and that the initial distribution  $f_0$  of (1.1) satisfies the following properties:*

- (i)  $f_0 \in L^1(\mathbb{R}^2; |r| dr dv_r) \cap L^p(\mathbb{R}^2; |r| dr dv_r)$  with  $p \geq 2$ ,
- (ii)  $f_0(r, v_r) \geq 0$  for all  $(r, v_r) \in \mathbb{R}^2$ ,
- (iii)  $\int_{\mathbb{R}^2} (r^2 + v_r^2) f_0(r, v_r) |r| dr dv_r < +\infty$ .

*Then, considering a sequence of solutions  $(f^\epsilon, \mathbf{E}_r^\epsilon)$  of (1.1) and extracting a subsequence from it, we can say that  $f^\epsilon$  two-scale converges to  $F \in L^\infty([0, T] \times [0, 2\pi]; L^2(\mathbb{R}^2; |r| dr dv_r))$  and  $\mathbf{E}_r^\epsilon$  two-scale converges to  $\mathcal{E}_r \in L^\infty([0, T] \times [0, 2\pi]; W^{1,3/2}(\mathbb{R}; |r| dr))$ . Furthermore, there exists a function  $G \in L^\infty([0, T]; L^2(\mathbb{R}^2; |q| dq du_r))$  such that*

$$F(r, v_r, \tau, t) = G(\cos(\tau) r - \sin(\tau) v_r, \sin(\tau) r + \cos(\tau) v_r, t), \quad (2.11)$$

and  $(G, \mathcal{E}_r)$  is solution of

$$\begin{aligned}
\partial_t G - \left[ \int_0^{2\pi} \sin(\sigma) \mathcal{E}_r(\cos(\sigma) q + \sin(\sigma) u_r, \sigma, t) d\sigma \right] \partial_q G \\
- \left[ \int_0^{2\pi} \sin(\sigma) \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma) q + \sin(\sigma) u_r) d\sigma \right] \partial_q G \\
+ \left[ \int_0^{2\pi} \cos(\sigma) \mathcal{E}_r(\cos(\sigma) q + \sin(\sigma) u_r, \sigma, t) d\sigma \right] \partial_{u_r} G \\
+ \left[ \int_0^{2\pi} \cos(\sigma) \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma) q + \sin(\sigma) u_r) d\sigma \right] \partial_{u_r} G = 0, \\
G(q, u_r, t = 0) = \frac{1}{2\pi} f_0(q, u_r), \\
\frac{1}{r} \partial_r (r \mathcal{E}_r(r, \tau, t)) = \int_{\mathbb{R}} G(\cos(\tau) r - \sin(\tau) v_r, \sin(\tau) r + \cos(\tau) v_r, t) dv_r,
\end{aligned} \tag{2.12}$$

where  $I_{\mathbb{Q}}(\omega_1)$  is equal to 1 if  $\omega_1 \in \mathbb{Q}$ , and 0 otherwise.

Of course, such a result exists for the solution of the system (2.9) and can be found in [13].

### 3 The two-scale semi-lagrangian method

In this section, we develop a two-scale semi-lagrangian method in order to approach the solution  $f^\epsilon$  of (1.1), in the case where  $H_0 = 1$ . As it has been suggested in the introduction, the strategy is to discretize the model (2.11)-(2.12) in order to obtain a good approximation of  $F$  which can be used for approaching  $f^\epsilon(r, v_r, t) \sim 2\pi F(r, v_r, \frac{t}{\epsilon}, t)$ . As in the two-scale PIC-method developed by Frénod, Salvarani and Sonnendrücker in [13], there is an advantage by proceeding in such a way: since there is no longer  $\frac{1}{\epsilon}$ -frequency oscillations in the system (2.12), we do not need a very small time step for good simulation. In a first time, we recall the basis of a semi-lagrangian method. Then we present a motivation for the development of a two-scale semi-lagrangian method through the description of a classical backward semi-lagrangian method on the model (1.1). Finally, we describe the two-scale numerical method itself and we suggest a new mesh in order to simplify it.

#### 3.1 The semi-lagrangian method

In this paragraph, we recall the way to discretize the abstract model

$$\partial_t f(\mathbf{x}, t) + \mathbf{U}(\mathbf{x}, t) \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, t) = 0 \tag{3.1}$$

with a semi-lagrangian method. For this, we have to consider the characteristics of (3.1), which are the solutions of

$$\partial_t \mathbf{X}(t) = \mathbf{U}(\mathbf{X}(t), t). \tag{3.2}$$

It is an easy game to remark that  $f$  is constant along the characteristics, i.e.

$$\partial_t (f(\mathbf{X}(t), t)) = 0, \tag{3.3}$$

so we can write

$$f(\mathbf{X}(t; \mathbf{x}, s), t) = f(\mathbf{x}, s), \tag{3.4}$$

where  $\mathbf{X}(t; \mathbf{x}, s)$  is the solution (3.2) with the condition  $\mathbf{X}(s) = \mathbf{x}$ .

This property of  $f$  is used in the semi-lagrangian method as follows: assuming that we know the value of  $f$  at time  $t_n - \Delta t$  on the mesh points  $(\mathbf{x}_i)_{i=0, \dots, N}$ , we use the property (3.4) to say that

$$f(\mathbf{x}_i, t_n + \Delta t) = f(\mathbf{X}(t_n - \Delta t; \mathbf{x}_i, t_n + \Delta t), t_n - \Delta t), \tag{3.5}$$

so we have to compute the point  $\mathbf{X}(t_n - \Delta t; \mathbf{x}_i, t_n + \Delta t)$  first, then compute  $f(\mathbf{X}(t_n - \Delta t; \mathbf{x}_i, t_n + \Delta t), t_n - \Delta t)$  by interpolating  $f(\cdot, t_n - \Delta t)$  on the points  $(\mathbf{x}_i)_{i=0, \dots, N}$  in order to obtain an approximation of  $f(\mathbf{x}_i, t_n + \Delta t)$ . Sonnendrücker *et al.* have suggested in [19] a way to compute a second order approximation of  $\mathbf{X}(t_n - \Delta t; \mathbf{x}_i, t_n + \Delta t)$ : they discretize the equation (3.2) with a finite difference method in order to obtain the following approximation:

$$\mathbf{X}(t_n - \Delta t; \mathbf{x}_i, t_n + \Delta t) = \mathbf{x}_i - 2 \mathbf{d}_i, \quad (3.6)$$

where  $\mathbf{d}_i$  is solution of

$$\mathbf{d}_i = \Delta t \mathbf{U}(\mathbf{x}_i - \mathbf{d}_i, t_n). \quad (3.7)$$

In many cases,  $\mathbf{U}(\cdot, t_n)$  is only known at points  $\mathbf{x}_i$ . Then we have to replace (3.7) by

$$\mathbf{d}_i = \Delta t \Pi \mathbf{U}(\mathbf{x}_i - \mathbf{d}_i, t_n), \quad (3.8)$$

where  $\Pi$  is an interpolation operator on points  $(\mathbf{x}_i)_i$ . Assuming that the polynomial function  $\mathbf{x} \mapsto \Pi \mathbf{U}(\mathbf{x}, t_n)$  is regular enough, we write the following expansion of (3.8):

$$\mathbf{d}_i = \Delta t \mathbf{U}(\mathbf{x}_i, t_n) - \Delta t \nabla_{\mathbf{x}}(\Pi \mathbf{U})(\mathbf{x}_i, t_n) \mathbf{d}_i + \mathcal{O}(\Delta t^2), \quad (3.9)$$

because, in the expansion of  $\Pi \mathbf{U}$  in  $\mathbf{x}$ , we get a  $\mathcal{O}(|\mathbf{d}_i|^2)$  term, which is a  $\mathcal{O}(\Delta t^2)$ . Then, we obtain a second order accurate approximation of  $\mathbf{d}_i$  given by

$$\mathbf{d}_i = \Delta t (\mathbf{Id} + \Delta t \nabla_{\mathbf{x}}(\Pi \mathbf{U})(\mathbf{x}_i, t_n))^{-1} \times \mathbf{U}(\mathbf{x}_i, t_n). \quad (3.10)$$

Considering this approach, the semi-lagrangian method writes:

1. knowing  $f$  at time  $t_n - \Delta t$  and  $\mathbf{U}$  at time  $t_n$ , we compute  $(\mathbf{d}_i)_{i=0, \dots, N}$  by using the relation (3.10) for each  $i$ ,
2. we compute  $f$  at time  $t_n + \Delta t$  as follows:

$$f(\mathbf{x}_i, t_n + \Delta t) = \Pi f(\mathbf{x}_i - 2 \mathbf{d}_i, t_n - \Delta t). \quad (3.11)$$

### 3.2 Implementation of the non-homogenized model

In this paragraph, we describe a classical semi-lagrangian method on the system (1.1). Since the electric fields  $E_r^\epsilon$  and  $\Xi_r^\epsilon$  do not depend on  $v_r$ , we can do a time-splitting on the first equation of the model, i.e. solving separately at each time step the equations

$$\partial_t f^\epsilon + \frac{v_r}{\epsilon} \partial_r f^\epsilon = 0, \quad (3.12)$$

and

$$\partial_t f^\epsilon + (E_r^\epsilon + \Xi_r^\epsilon) \partial_{v_r} f^\epsilon = 0, \quad (3.13)$$

with a second order in time numerical scheme instead of solving the complete equation with the same scheme. As a consequence, we only do 1D interpolations instead of 2D interpolations. Then, denoting  $f_n^\epsilon$  with the approximation of  $f^\epsilon(\cdot, \cdot, t_n)$  and  $E_n^\epsilon$  with the approximation of  $E_r^\epsilon(\cdot, t_n)$  on the uniform mesh  $(r_i, v_{rj})_{i,j}$  where  $\Delta r$  is the size of one cell in  $r$  direction and  $\Delta v_r$  is the size of one cell in  $v_r$  direction, an iteration of the method is organized as follows:

1. Knowing  $f_n^\epsilon$  and  $E_n^\epsilon$ , we do a backward advection of  $\frac{\Delta t}{2}$  in  $v_r$  direction and we define  $f_*^\epsilon$  by

$$f_*^\epsilon(r_i, v_{rj}) = \Pi_{v_r} f_n^\epsilon \left( r_i, v_{rj} - \frac{\Delta t}{2} (E_n^\epsilon(r_i) + \Xi_r^\epsilon(r_i, t_n)) \right), \quad (3.14)$$

where  $\Pi_{v_r}$  is a 1D cubic spline interpolation operator on the points  $(v_{rj})_j$ .

2. We do an advection of  $\Delta t$  in  $r$  direction and we define  $f_{**}^\epsilon$  by

$$f_{**}^\epsilon(r_i, v_{rj}) = \Pi_r f_*^\epsilon\left(r_i - \frac{\Delta t}{\epsilon} v_{rj}, v_{rj}\right), \quad (3.15)$$

where  $\Pi_r$  is a 1D cubic spline interpolation operator on the points  $(r_i)_i$ .

3. We compute  $E_{n+1}^\epsilon$  by discretizing the formula

$$E_{n+1}^\epsilon(r_i) = \begin{cases} \frac{1}{r_i} \int_0^{r_i} \int_{\mathbb{R}} s f_{**}^\epsilon(s, v_r) dv_r ds & \text{if } r_i \neq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (3.16)$$

with the trapezoidal rule on the points  $(r_i, v_{rj})_{i,j}$ .

4. We compute  $f_{n+1}^\epsilon$  by doing a last advection of  $\frac{\Delta t}{2}$  in  $v_r$  direction:

$$f_{n+1}^\epsilon(r_i, v_{rj}) = \Pi_{v_r} f_{**}^\epsilon\left(r_i, v_{rj} - \frac{\Delta t}{2} (E_{n+1}^\epsilon(r_i) + \Xi_r^\epsilon(r_i, t_{n+1}))\right). \quad (3.17)$$

If we use such a method, we have to guarantee the accuracy of the scheme, especially we consider the  $\frac{1}{\epsilon}$ -frequency oscillations of the external electric field. A solution is to assume that the time step  $\Delta t$  satisfies

$$\begin{cases} r_i - \Delta r \leq r_i - \frac{\Delta t}{\epsilon} v_{rj} \leq r_i + \Delta r, \\ v_{rj} - \Delta v_r \leq v_{rj} - \frac{\Delta t}{2} (E_n^\epsilon(r_i) + \Xi_r^\epsilon(r_i, t_n)) \leq v_{rj} + \Delta v_r, \end{cases} \quad (3.18)$$

for all  $i, j, n$ . Then, in order to obtain good results with this method,  $\Delta t$  has to be of the same order of  $\epsilon$  which penalizes the method in terms of CPU time cost when we consider a very small  $\epsilon$ .

### 3.3 Implementation of the two-scale model

In this paragraph, we adapt the semi-lagrangian method we have described in the paragraph 3.1 to the model (2.12). In this case, the characteristics of the system are the solutions of

$$\begin{cases} \partial_t Q(t) = \langle \mathcal{E}_1 \rangle(Q(t), U_r(t), t), \\ \partial_t U_r(t) = \langle \mathcal{E}_2 \rangle(Q(t), U_r(t), t), \end{cases} \quad (3.19)$$

where  $\langle \mathcal{E}_1 \rangle$  and  $\langle \mathcal{E}_2 \rangle$  are defined by

$$\begin{cases} \langle \mathcal{E}_1 \rangle(q, u_r, t) = - \int_0^{2\pi} \sin(\sigma) \left[ \mathcal{E}_r(\cos(\sigma) q + \sin(\sigma) u_r, \sigma, t) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma) q + \sin(\sigma) u_r) \right] d\sigma, \\ \langle \mathcal{E}_2 \rangle(q, u_r, t) = \int_0^{2\pi} \cos(\sigma) \left[ \mathcal{E}_r(\cos(\sigma) q + \sin(\sigma) u_r, \sigma, t) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma) q + \sin(\sigma) u_r) \right] d\sigma. \end{cases} \quad (3.20)$$

As in the paragraph 3.1, we remark that the solution  $G$  of (2.12) is constant along the characteristics, so we can write:

$$G(q, u_r, t_{n+1}) = G(Q(t_{n-1}; q, t_{n+1}), U_r(t_{n-1}; u_r, t_{n+1}), t_{n-1}), \quad (3.21)$$

where  $t_n = n \Delta t$ , and where  $(Q(t_{n-1}; q, t_{n+1}), U_r(t_{n-1}; u_r, t_{n+1}))$  is the solution of (3.19) with the condition  $(Q(t_{n+1}), U_r(t_{n+1})) = (q, u_r)$ .



Firstly, we define the mesh in  $q$  and  $u_r$  directions by considering the points  $q_i = i \Delta q$  and  $u_{rj} = j \Delta u_r$  ( $i = -P_q, \dots, P_q, j = -P_{u_r}, \dots, P_{u_r}$ ). We also consider the uniform mesh  $\tau_m = m \Delta \tau$  on  $[0, 2\pi]$  ( $m = 0, \dots, P_\tau$ ). Finally, we fix a time step  $\Delta t$  for the entire simulation. Then, denoting  $G^n$  with the approximation of  $G$  at time  $t_n$ , an iteration of the semi-lagrangian method is organized as follows:

1. Assuming that we know the value of  $G^n$  and  $G^{n-1}$  on the mesh  $(q_i, u_{rj})_{i,j}$ , we compute  $\mathcal{E}_r^n$  with the formula

$$\mathcal{E}_r^n(q_i, \tau_m) = \begin{cases} \frac{1}{q_i} \int_0^{q_i} \int_{\mathbb{R}} s G^n (\cos(\tau_m) s - \sin(\tau_m) v_r, \\ \sin(\tau_m) s + \cos(\tau_m) v_r) ds dv_r & \text{if } i \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3.22)$$

Since  $G^n$  is only known at points  $(q_i, u_{rj})$ , we have to interpolate  $G^n$ . Assuming that the support of  $G^n$  is included in  $[-(P_q + 1)\Delta q, (P_q + 1)\Delta q] \times [-(P_{u_r} + 1)\Delta u_r, (P_{u_r} + 1)\Delta u_r]$  with  $P_q, P_{u_r} \in \mathbb{N}$  large enough, we use the trapezoidal rule in order to approach the integral above. We obtain

$$\begin{aligned} \mathcal{E}_r^n(q_i, \tau_m) \approx & \frac{\Delta q \Delta u_r}{2} \sum_{j=-P_{u_r}}^{P_{u_r}} \left( \Pi_2 G^n (\cos(\tau_m) q_i - \sin(\tau_m) u_{rj}, \sin(\tau_m) q_i + \cos(\tau_m) u_{rj}) \right. \\ & \left. + \frac{2}{i} \sum_{k=1}^{i-1} k \Pi_2 G^n (\cos(\tau_m) q_k - \sin(\tau_m) u_{rj}, \sin(\tau_m) q_k + \cos(\tau_m) u_{rj}) \right), \end{aligned} \quad (3.23)$$

where  $\Pi_2$  is a cubic spline interpolation operator on the points  $(q_i, u_{rj})$ .

2. We compute  $\langle \mathcal{E}_1^n \rangle$  and  $\langle \mathcal{E}_2^n \rangle$  at points  $(q_i, u_{rj})$ : since  $\mathcal{E}_r^n$  is only known at points  $(q_i, \tau_m)$ , we have to interpolate  $\mathcal{E}_r^n$ . By using the trapezoidal rule for approximating the integrals in (3.20), we obtain

$$\begin{cases} \langle \mathcal{E}_1^n \rangle(q_i, u_{rj}) \approx -\Delta \tau \sum_{m=0}^{P_\tau} \sin(\tau_m) \left[ \Pi_1 \mathcal{E}_r^n (\cos(\tau_m) q_i + \sin(\tau_m) u_{rj}, \tau_m) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \tau_m) (\cos(\tau_m) q_i + \sin(\tau_m) u_{rj}) \right], \\ \langle \mathcal{E}_2^n \rangle(q_i, u_{rj}) \approx \Delta \tau \sum_{m=0}^{P_\tau} \cos(\tau_m) \left[ \Pi_1 \mathcal{E}_r^n (\cos(\tau_m) q_i + \sin(\tau_m) u_{rj}, \tau_m) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \tau_m) (\cos(\tau_m) q_i + \sin(\tau_m) u_{rj}) \right], \end{cases} \quad (3.24)$$

where  $\Pi_1$  is a cubic spline interpolation operator on the points  $(q_i)_i$ .

3. We compute the shifts  $(d_{i,j}^1, d_{i,j}^2)$  by using the following formula:

$$\begin{pmatrix} d_{i,j}^1 \\ d_{i,j}^2 \end{pmatrix} = \Delta t \mathbf{A}_{i,j}^{-1} \begin{pmatrix} \langle \mathcal{E}_1^n \rangle(q_i, u_{rj}) \\ \langle \mathcal{E}_2^n \rangle(q_i, u_{rj}) \end{pmatrix}, \quad (3.25)$$

where the matrix  $\mathbf{A}_{i,j}$  is defined by

$$\mathbf{A}_{i,j} = \mathbf{Id} + \Delta t \begin{pmatrix} \partial_q (\Pi_2 \langle \mathcal{E}_1^n \rangle)(q_i, u_{rj}) & \partial_{u_r} (\Pi_2 \langle \mathcal{E}_1^n \rangle)(q_i, u_{rj}) \\ \partial_q (\Pi_2 \langle \mathcal{E}_2^n \rangle)(q_i, u_{rj}) & \partial_{u_r} (\Pi_2 \langle \mathcal{E}_2^n \rangle)(q_i, u_{rj}) \end{pmatrix}. \quad (3.26)$$

4. We compute  $G^{n+1}$  by interpolating  $G^{n-1}$  at the points  $(q_i - 2d_{i,j}^1, u_{rj} - 2d_{i,j}^2)$ :

$$G^{n+1}(q_i, u_{rj}) = \Pi_2 G^{n-1}(q_i - 2d_{i,j}^1, u_{rj} - 2d_{i,j}^2). \quad (3.27)$$

5. We save the approximation of  $f^\epsilon$  at time  $t_{n+1}$  given by

$$f^\epsilon(r, v_r, t_{n+1}) \sim 2\pi \Pi_2 G^{n+1} \left( \cos\left(\frac{t_{n+1}}{\epsilon}\right) r - \sin\left(\frac{t_{n+1}}{\epsilon}\right) v_r, \sin\left(\frac{t_{n+1}}{\epsilon}\right) r + \cos\left(\frac{t_{n+1}}{\epsilon}\right) v_r \right). \quad (3.28)$$

In order to initialize this two time step advance, we have to compute  $G^1$  from  $G^0$  which is given as an initial data: for this purpose, we perform a complete iteration such as described above where  $\Delta t$  is replaced by  $\frac{\Delta t}{2}$  and where we assume that  $G^{1/2} = G^0$ .

### 3.4 The two-scale mesh

In most of test cases, we assume that the support of the initial distribution  $f_0$  is compact and is included in some  $\Omega = [-R, R] \times [-v_R, v_R] \subset \mathbb{R}^2$  for  $R > 0$ ,  $v_R > 0$  large enough. Then, if we follow the algorithm presented in the previous paragraph, the first thing we have to compute is  $\mathcal{E}_r^0$  by approximating the integral in (3.22): for numerical reasons, we have to reduce the integral on  $\mathbb{R}$  to an integral on a compact interval. Furthermore, since we can only say that the support of  $(r, v_r) \mapsto f_0(\cos(\tau)r - \sin(\tau)v_r, \sin(\tau)r + \cos(\tau)v_r)$  is included in  $\Omega' = [-R - v_R, R + v_R] \times [-R - v_R, R + v_R]$  as it is illustrated in Figure 1, the integral on  $\mathbb{R}$  in (3.22) is reduced to an integral on  $[-R - v_R, R + v_R]$ . As a consequence, we have to do all the simulation on  $\Omega'$  instead of  $\Omega$  in order to avoid losing some data and we have to increase the number of mesh points in order to keep good interpolation results, even if the distribution function will be equal to 0 at many new points.

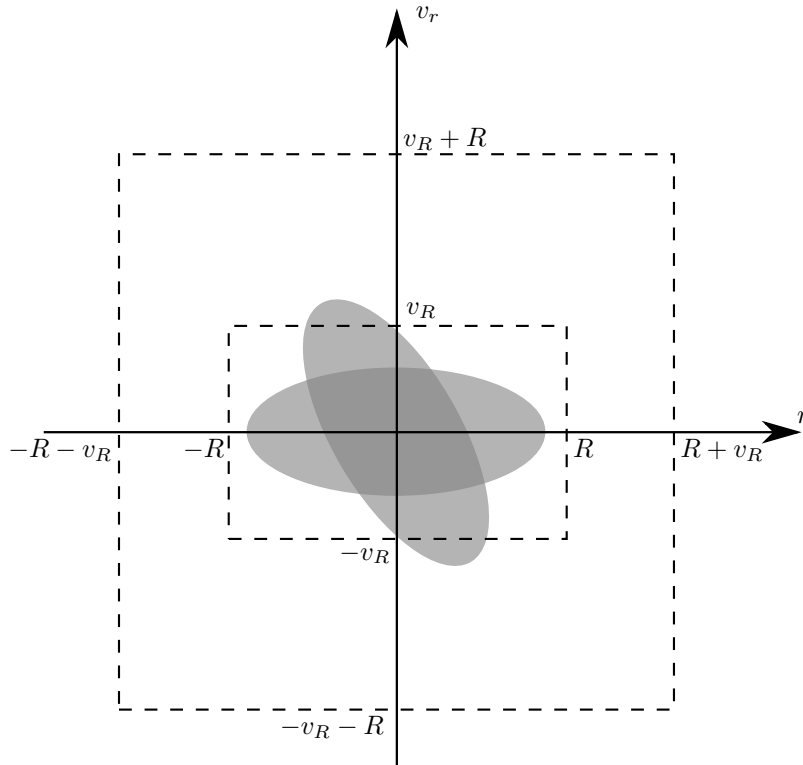


Figure 1: Support of  $(r, v_r) \mapsto f_0(r, v_r)$  and  $(r, v_r) \mapsto f_0(\cos(\tau)r - \sin(\tau)v_r, \sin(\tau)r + \cos(\tau)v_r)$  with  $\tau = \frac{2\pi}{3}$ .

In this paragraph, we present a different approach in order to avoid this extension of the simulation domain and its mesh. Before explaining the main idea of this new method, we consider

the following meshes on  $\Omega$  and  $[0, 2\pi]$ :

$$\begin{aligned}
M(\Omega) &= \left\{ (r_i, v_{rj}) = (i \Delta r, j \Delta v_r) : i = -P_r, \dots, P_r, j = -P_{v_r}, \dots, P_{v_r} \right\}, \\
M([0, 2\pi]) &= \left\{ \tau_m = m \Delta \tau : m = 0, \dots, P_\tau \right\}, \\
M([-R, R]) &= \left\{ r_i = i \Delta r : i = -P_r, \dots, P_r \right\},
\end{aligned} \tag{3.29}$$

where  $\Delta r = \frac{R}{P_r+1}$ ,  $\Delta v_r = \frac{v_R}{P_{v_r}+1}$  and  $\Delta \tau = \frac{2\pi}{P_\tau+1}$ . Considering the function  $\gamma$  defined by

$$\begin{aligned}
\gamma : \mathbb{R}^2 \times [0, 2\pi] &\longrightarrow \mathbb{R}^2 \\
(r, v_r, \tau) &\longmapsto (\cos(\tau) r - \sin(\tau) v_r, \sin(\tau) r + \cos(\tau) v_r) \quad ,
\end{aligned} \tag{3.30}$$

we define  $\Omega(\tau)$  and  $M(\Omega(\tau))$  by

$$\begin{aligned}
\Omega(\tau) &= \gamma(\Omega \times \{\tau\}) \subset \mathbb{R}^2, \\
M(\Omega(\tau)) &= \left\{ \gamma(r_i, v_{rj}, \tau) : i = -P_r, \dots, P_r, j = -P_{v_r}, \dots, P_{v_r} \right\}.
\end{aligned} \tag{3.31}$$

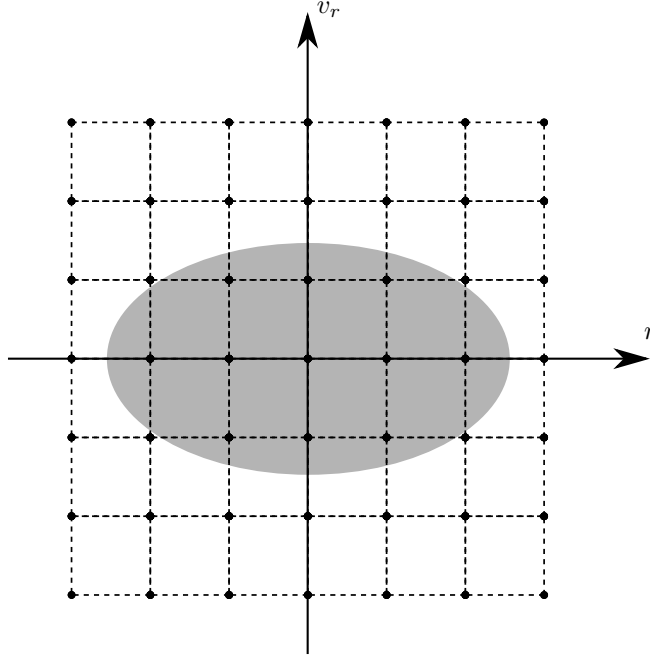


Figure 2: Mesh  $M(\Omega) = M(\Omega(0))$  and support of  $(r, v_r) \mapsto f_0(r, v_r)$ .

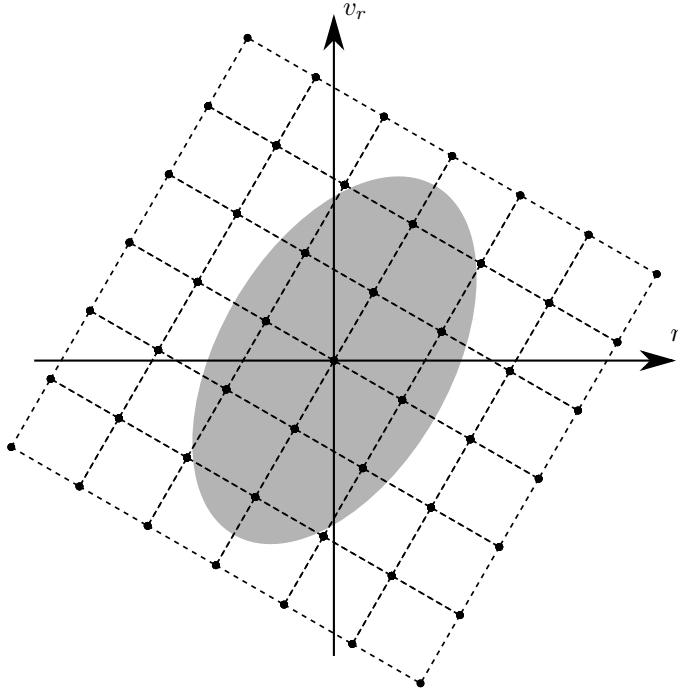


Figure 3: Mesh  $M(\Omega(\frac{\pi}{3}))$  and support of  $(r, v_r) \mapsto f_0 \circ \gamma(r, v_r, \frac{\pi}{3})$ .

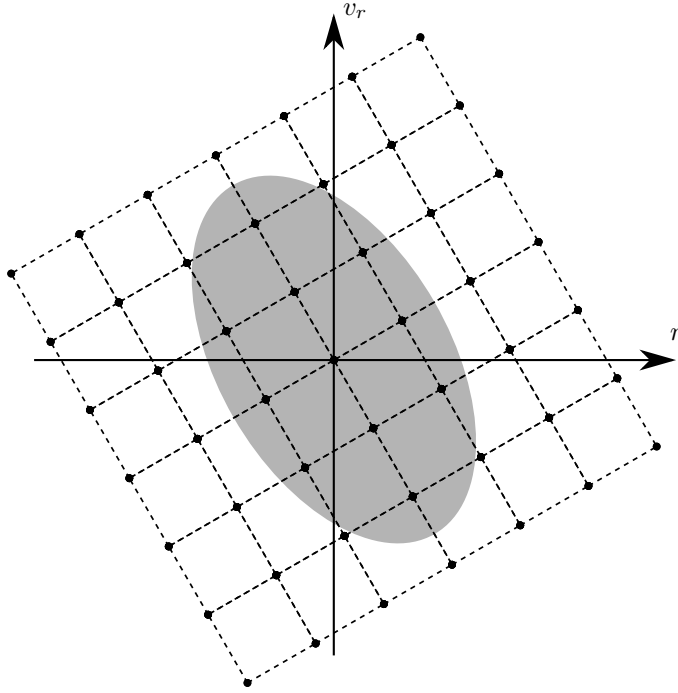


Figure 4: Mesh  $M(\Omega(\frac{2\pi}{3}))$  and support of  $(r, v_r) \mapsto f_0 \circ \gamma(r, v_r, \frac{2\pi}{3})$ .

The main idea of our new method is to compute  $(r, v_r) \mapsto F(r, v_r, \tau_m, t_n)$  at points  $(r_i, v_{rj}) \in M(\Omega)$  and to compute  $r \mapsto \mathcal{E}_r(r, \tau_m, t_n)$  at points  $r_i \in M([-R, R])$ , whereas the function  $(q, u_r) \mapsto G(q, u_r, t_n)$  is computed at points  $\gamma(r_i, v_{rj}, \tau_m) \in M(\Omega(\tau_m))$  for every  $\tau_m \in M([0, 2\pi])$ . This approach is similar as the time-dependent moving grid described by Lang *et al.* in [16], even if, in our case, the mesh  $M(\Omega(\tau))$  only depends on  $M(\Omega)$  and  $\tau$ , and is completely defined before the beginning of the simulation.

As a first consequence, considering that the support of  $(r, v_r) \mapsto f_0(r, v_r)$  is included in  $\Omega$  is equivalent to considering that the support of  $(r, v_r) \mapsto f_0(\gamma(r, v_r, \tau))$  is included in  $\Omega(\tau)$  for any  $\tau \in [0, 2\pi]$  such as illustrated in Figures 2,3, and 4, where the support of a Kapchinsky-Vladimirsky distribution is represented (see [15], [18] or [8] for more details about this distribution). Then we do not have to extend  $\Omega$  in order to avoid losing some data. Furthermore, the equation (2.11) reads

$$F(r, v_r, \tau, t) = G(\gamma(r, v_r, \tau), t). \quad (3.32)$$

Considering this approach, we fix a time step  $\Delta t$  for all the simulation. Then, an iteration of this new semi-lagrangian method is organized as follows:

1. Assuming that  $G^{n-1}$  and  $G^n$  are known on the mesh  $M(\Omega(\tau_m))$  for all  $\tau_m \in M([0, 2\pi])$ , we compute  $\mathcal{E}_r^n$  at points  $(r_i, \tau_m) \in M([-R, R]) \times M([0, 2\pi])$ . With the new notations, the equation (3.22) simplifies itself to

$$\mathcal{E}_r^n(r_i, \tau_m) = \begin{cases} \frac{1}{r_i} \int_0^{r_i} \int_{\mathbb{R}} s G^n(\gamma(s, v_r, \tau_m)) dv_r ds & \text{if } i \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3.33)$$

Assuming that the support of  $G^n(\gamma(\cdot, \cdot, \tau_m))$  is included in  $\Omega(\tau_m)$  for each  $\tau_m$  (which is equivalent to assume that the support of  $G^n(\gamma(\cdot, \cdot, 0))$  is included in  $\Omega$ ), we use the trapezoidal rule to approximate the integral above:

$$\mathcal{E}_r^n(r_i, \tau_m) \approx \frac{\Delta r \Delta v_r}{2} \sum_{j=-P_{v_r}}^{P_{v_r}} \left( G^n(\gamma(r_i, v_{rj}, \tau_m)) + \frac{2}{i} \sum_{k=1}^{i-1} k G^n(\gamma(r_k, v_{rj}, \tau_m)) \right). \quad (3.34)$$

We remark here that, contrary to the computation done in (3.23), we do not have to interpolate  $G^n$ .

2. We compute  $\langle \mathcal{E}_1^n \rangle$  and  $\langle \mathcal{E}_2^n \rangle$  at points  $\gamma(r_i, v_{rj}, \tau_m) \in M(\Omega(\tau_m))$ :

$$\begin{cases} \langle \mathcal{E}_1^n \rangle(\gamma(r_i, v_{rj}, \tau_m)) = - \int_0^{2\pi} \sin(\sigma) \left[ \mathcal{E}_r^n(\cos(\sigma - \tau_m) r_i + \sin(\sigma - \tau_m) v_{rj}, \sigma) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma - \tau_m) r_i + \sin(\sigma - \tau_m) v_{rj}) \right] d\sigma, \\ \langle \mathcal{E}_2^n \rangle(\gamma(r_i, v_{rj}, \tau_m)) = \int_0^{2\pi} \cos(\sigma) \left[ \mathcal{E}_r^n(\cos(\sigma - \tau_m) r_i + \sin(\sigma - \tau_m) v_{rj}, \sigma) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \sigma) (\cos(\sigma - \tau_m) r_i + \sin(\sigma - \tau_m) v_{rj}) \right] d\sigma. \end{cases} \quad (3.35)$$

We approximate the integrals with the trapezoidal rule. However, since  $\mathcal{E}_r^n$  is only known at points of  $M([-R, R]) \times M([0, 2\pi])$ , we have to interpolate it: for that we choose a cubic spline interpolation operator on the mesh  $M([-R, R])$  and we denote it with  $\Pi_1$ . Then we have the following approximations:

$$\begin{cases} \langle \mathcal{E}_1^n \rangle(\gamma(r_i, v_{rj}, \tau_m)) \approx -\Delta\tau \sum_{k=0}^{P_\tau} \sin(\tau_k) \left[ \Pi_1 \mathcal{E}_r^n(\cos(\tau_k - \tau_m) r_i + \sin(\tau_k - \tau_m) v_{rj}, \tau_k) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \tau_k) (\cos(\tau_k - \tau_m) r_i + \sin(\tau_k - \tau_m) v_{rj}) \right], \\ \langle \mathcal{E}_2^n \rangle(\gamma(r_i, v_{rj}, \tau_m)) \approx \Delta\tau \sum_{k=0}^{P_\tau} \cos(\tau_k) \left[ \Pi_1 \mathcal{E}_r^n(\cos(\tau_k - \tau_m) r_i + \sin(\tau_k - \tau_m) v_{rj}, \tau_k) \right. \\ \quad \left. + \frac{I_{\mathbb{Q}}(\omega_1)}{2\pi} H_1(\omega_1 \tau_k) (\cos(\tau_k - \tau_m) r_i + \sin(\tau_k - \tau_m) v_{rj}) \right]. \end{cases} \quad (3.36)$$

3. We compute the shifts  $\mathbf{d}(r_i, v_{rj}, \tau_m)$  verifying

$$\mathbf{d}(r_i, v_{rj}, \tau_m) = \Delta t \begin{pmatrix} \langle \mathcal{E}_1^n \rangle (\gamma(r_i, v_{rj}, \tau_m) - \mathbf{d}(r_i, v_{rj}, \tau_m)) \\ \langle \mathcal{E}_2^n \rangle (\gamma(r_i, v_{rj}, \tau_m) - \mathbf{d}(r_i, v_{rj}, \tau_m)) \end{pmatrix}. \quad (3.37)$$

For that, we consider the cubic spline interpolation operator  $\Pi_2^m$  on the mesh  $M(\Omega(\tau_m))$  and, inspired by (3.10), we consider the following approximation of  $\mathbf{d}(r_i, v_{rj}, \tau_m)$ :

$$\mathbf{d}(r_i, v_{rj}, \tau_m) = \Delta t \mathbf{A}_{i,j,m}^{-1} \begin{pmatrix} \langle \mathcal{E}_1^n \rangle (\gamma(r_i, v_{rj}, \tau_m)) \\ \langle \mathcal{E}_2^n \rangle (\gamma(r_i, v_{rj}, \tau_m)) \end{pmatrix}, \quad (3.38)$$

where the matrix  $\mathbf{A}_{i,j,m}$  is defined by

$$\mathbf{A}_{i,j,m} = \mathbf{Id} + \Delta t \begin{pmatrix} \partial_q(\Pi_2^m \langle \mathcal{E}_1^n \rangle)(\gamma(r_i, v_{rj}, \tau_m)) & \partial_{u_r}(\Pi_2^m \langle \mathcal{E}_1^n \rangle)(\gamma(r_i, v_{rj}, \tau_m)) \\ \partial_q(\Pi_2^m \langle \mathcal{E}_2^n \rangle)(\gamma(r_i, v_{rj}, \tau_m)) & \partial_{u_r}(\Pi_2^m \langle \mathcal{E}_2^n \rangle)(\gamma(r_i, v_{rj}, \tau_m)) \end{pmatrix}. \quad (3.39)$$

4. We compute  $G^{n+1}$  on the meshes  $M(\Omega(\tau_m))$  for all  $\tau_m \in M([0, 2\pi])$ :

$$G^{n+1}(\gamma(r_i, v_{rj}, \tau_m)) = \Pi_2^m G^{n-1}(\gamma(r_i, v_{rj}, \tau_m) - 2\mathbf{d}(r_i, v_{rj}, \tau_m)) \quad (3.40)$$

5. Assuming that there exists a fixed integer  $K \in \mathbb{N}$  such that

$$\Delta t = \epsilon \Delta \tau K, \quad (3.41)$$

we save the approximation of  $f^\epsilon$  on  $M(\Omega)$  given by

$$f^\epsilon(r_i, v_{rj}, t_{n+1}) \sim 2\pi G^{n+1}(\gamma(r_i, v_{rj}, \tau_{(n+1)K})). \quad (3.42)$$

Contrary to the scheme described in the previous paragraph, we do not have to interpolate  $G^{n+1}$  to obtain an approximation of  $f^\epsilon$  at time  $t_{n+1}$ .

Concerning the initialization of this two time step advance, we compute  $G^0 = \frac{1}{2\pi} f_0$  on the meshes  $M(\Omega(\tau_m))$  for all  $\tau_m \in M([0, 2\pi])$  and, assuming that  $G^{1/2} = G^0$ , we compute  $G^1$  by performing a complete iteration such as described above with  $\Delta t$  replaced by  $\frac{\Delta t}{2}$ .

We have built another two-scale semi-lagrangian method based on a mesh depending on the variable  $\tau$ . Compared to the semi-lagrangian method described in the paragraph 3.3, this technique allows us to avoid some interpolations not only when we compute the two-scale limit of the electric field  $\mathcal{E}$  but also when we build the approximation of  $f^\epsilon$  defined by (3.42). As a consequence, we reduces the global numerical diffusion within the simulation. However, we have to compute  $G$  and  $\langle \mathcal{E} \rangle$  on each mesh  $M(\Omega(\tau_m))$ , which can be expensive in CPU time.

## 4 Numerical results

In this section, we present some numerical results obtained with the two-scale semi-lagrangian methods described in the paragraphs 3.3 and 3.4. Following the approach of [13] for validating our new methods, we study in a first time some linear cases where we can find an analytic expression of the solution  $G$  of the system (2.12). Then we test the methods on non-linear cases.

### 4.1 Linear cases

In order to validate the two-scale semi-lagrangian methods described in the paragraphs 3.3 and 3.4, we simulate some linear cases, i.e. with a self-consistent electric field set to 0. This assumption allows us to compute analytically the solution  $G$  of the system (2.12) under an adequate choice of  $\omega_1$  and  $H_1$ .

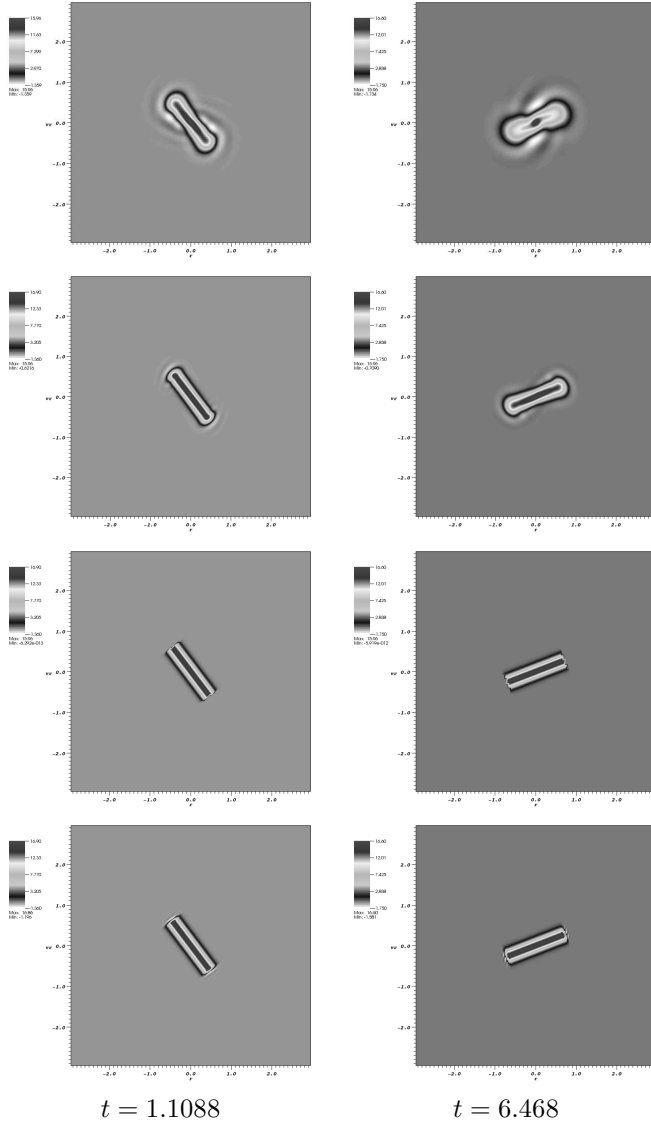


Figure 5: Simulations of type (I) (first row), (II) (second row), (III) (third row), and (IV) (fourth row) for a semi-gaussian beam without self-consistent electric field and  $\omega_1 = 4\sqrt{2}$ ,  $H_1(\tau) = \cos(\tau)$ .

In a first example, we suppose that  $\omega_1 \notin \mathbb{Q}$ , so we obtain that  $G$  is stationary in  $t$ , and is equal to  $\frac{1}{2\pi} f_0$ . Then, the two-scale simulation reduces itself to the computation of

$$(r, v_r, t) \mapsto f_0 \left( \cos\left(\frac{t}{\epsilon}\right) r - \sin\left(\frac{t}{\epsilon}\right) v_r, \sin\left(\frac{t}{\epsilon}\right) r + \cos\left(\frac{t}{\epsilon}\right) v_r \right), \quad (4.1)$$

for any function  $H_1$ . It is much simpler than simulating the model (1.1) with the semi-lagrangian method described in the paragraph 3.2. In Figure 5, we observe such a case, with  $\omega_1 = 4\sqrt{2}$ ,  $H_1(\tau) = \cos(\tau)$ , and  $f_0$  given by

$$f_0(r, v_r, t) = \frac{n_0}{\sqrt{2\pi} v_{th}} \exp\left(-\frac{v_r^2}{2v_{th}^2}\right) \chi_{[-r_m, r_m]}(r), \quad (4.2)$$

with  $\chi_{[-r_m, r_m]}(r) = 1$  if  $|r| \leq r_m$  and 0 otherwise. This corresponds to a semi-gaussian beam in particle accelerator physics. In these Figures, we suppose that  $r_m = 0.75$ ,  $v_{th} = 0.1$ ,  $n_0 = 4$  and  $\epsilon = 10^{-2}$ . Furthermore, the simulations (I), (II), (III) and (IV) correspond to

- simulation (I): we solve the system (1.1) with a classical semi-lagrangian method, with  $P_r = P_{v_r} = 64$  and  $R = v_R = 3$ ,
- simulation (II): we solve the system (1.1) with a classical semi-lagrangian method, with  $P_r = P_{v_r} = 128$  and  $R = v_R = 3$ ,
- simulation (III): we solve the system (2.12) with a two-scale semi-lagrangian method on a two-scale mesh, with  $P_r = P_{v_r} = 64$ ,  $P_\tau = 16$  and  $R = v_R = 3$ ,
- simulation (IV): we solve the system (2.12) with a two-scale semi-lagrangian method on a uniform mesh, with  $P_q = P_{u_r} = 128$ ,  $P_\tau = 16$ , and  $R = v_R = 3$ , and we compute the approximation (3.28) on a uniform  $129 \times 129$  grid in  $(r, v_r)$  on  $[-R, R] \times [-v_R, v_R]$ .

Since we can compute an analytic solution  $G$  of (2.12), we can compare the approximations of the solution  $f^\epsilon$  of (1.1) to the function defined by (4.1). These comparisons are summarized in Figure 6: in this figure, we present the  $L^1$  norm of the difference between the function (4.1) and

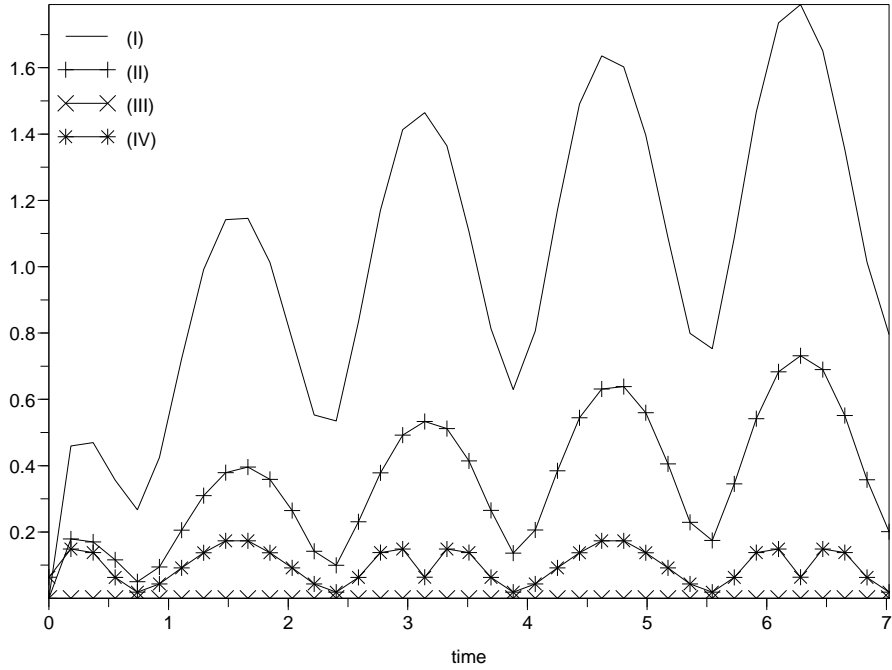


Figure 6: Evolution of the  $L^1$  norm of the difference between the function (4.1) and the approximation  $f_h$  computed with the simulations (I), (II), (III), and (IV).

In a second example, we suppose that  $\omega \in \mathbb{N}_{\geq 2}$ . Then, if we assume that  $H_1(\tau) = \cos^2(\tau)$ , the model (2.12) reduces itself to

$$\partial_t G - \frac{u_r}{4} \partial_q G + \frac{q}{4} \partial_{u_r} G = 0. \quad (4.3)$$

Knowing the initial data  $f_0$ , we can write

$$G(q, u_r, t) = \frac{1}{2\pi} f_0 \left( \cos\left(\frac{t}{4}\right) q + \sin\left(\frac{t}{4}\right) u_r, -\sin\left(\frac{t}{4}\right) q + \cos\left(\frac{t}{4}\right) u_r \right), \quad (4.4)$$

so the two-scale simulation reduces itself to the computation of

$$(r, v_r, t) \mapsto f_0 \left( \cos\left(\frac{t}{4} - \frac{t}{\epsilon}\right) r + \sin\left(\frac{t}{4} - \frac{t}{\epsilon}\right) v_r, -\sin\left(\frac{t}{4} - \frac{t}{\epsilon}\right) r + \cos\left(\frac{t}{4} - \frac{t}{\epsilon}\right) v_r \right). \quad (4.5)$$



In Figure 7, we observe such a case with  $\omega_1 = 2$ ,  $f_0$  given by (4.2) with  $r_m = 0.75$ ,  $v_{th} = 0.1$  and  $\epsilon = 10^{-2}$ .

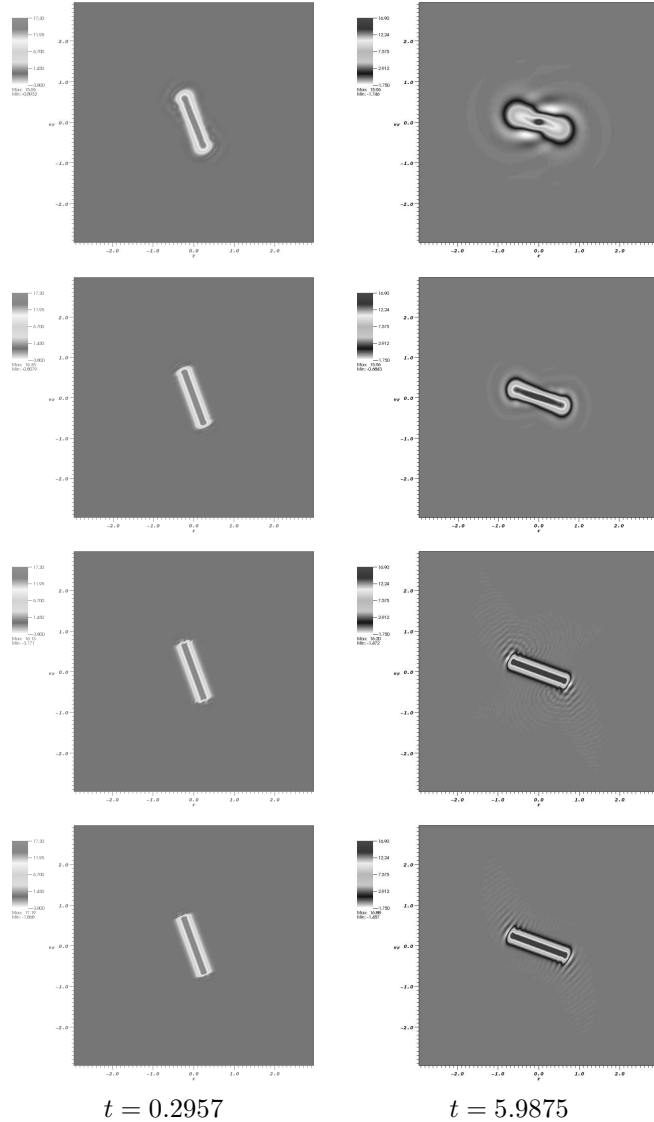


Figure 7: Simulations of type (I) (first row), (II) (second row), (III) (third row), and (IV) (fourth row) for a semi-gaussian beam without self-consistent electric field and  $\omega_1 = 2$ ,  $H_1(\tau) = \cos^2(\tau)$ .

As we can see in Figures 5 and 7, the classical semi-lagrangian method needs a very refined mesh in  $r$  and  $v_r$  directions in order to produce good results. Otherwise, it produces results which do not correspond to physics. We can explain it by the small time step induced by the condition (3.18) in order to guarantee the robustness of the method, and then a very high number of interpolations which introduce numerical diffusion. As an example, we have taken a time step  $\Delta t_{NH} \approx 1.5 \times 10^{-4}$  for the simulation of type (I) of the second case.

On the other hand, both two-scale numerical methods we have described in paragraphs 3.3 and 3.4 produce good results, even if we consider a non-refined mesh in  $r$  and  $v_r$  directions: this phenomenon can be explained by the independence of the time step in  $\epsilon$ . Consequently, we are allowed to take a time step  $\Delta t_H$  much larger than we can in the classical semi-lagrangian context, and then, we can significantly reduce the number of interpolations and the numerical diffusion which is induced. For example, for simulations of type (III) and (IV) on the second test case, we have defined  $\Delta t_H$  with the formula (3.41) where  $K = 2$ , giving us  $\Delta t_H \approx 7.4 \times 10^{-3}$ .

Finally, by observing Figure 6, we remark that the use of the two-scale mesh in the first case reduces significantly the error between the function defined by (4.1) and the approximation of  $f^\epsilon$  given by the discretization of the two-scale model (2.12): indeed, the  $L^1$  norm of this error is nearby  $10^{-6}$  when the approximation of  $f^\epsilon$  is given by the simulation (III) whereas it oscillates between 0 and 0.75 when the approximation of  $f^\epsilon$  is given by the simulation (IV). Furthermore, the oscillations of this error for simulations (I), (II), and (IV) are due to the fact that we discretize a non-smooth semi-gaussian distribution the support of which rotates in the phase space under the action of the external electric field and that we do it on a uniform phase space mesh.

As a first conclusion, we can say that, on linear cases, the two-scale semi-lagrangian methods we have proposed give much better results than the classical semi-lagrangian method on the same mesh in  $r$  and  $v_r$ , which is promising for non-linear cases. Furthermore, as it was announced in the paragraph 3.4, the use of the two-scale mesh reduces significantly the numerical diffusion linked with the global number of interpolations within the simulation.

## 4.2 Non-linear cases

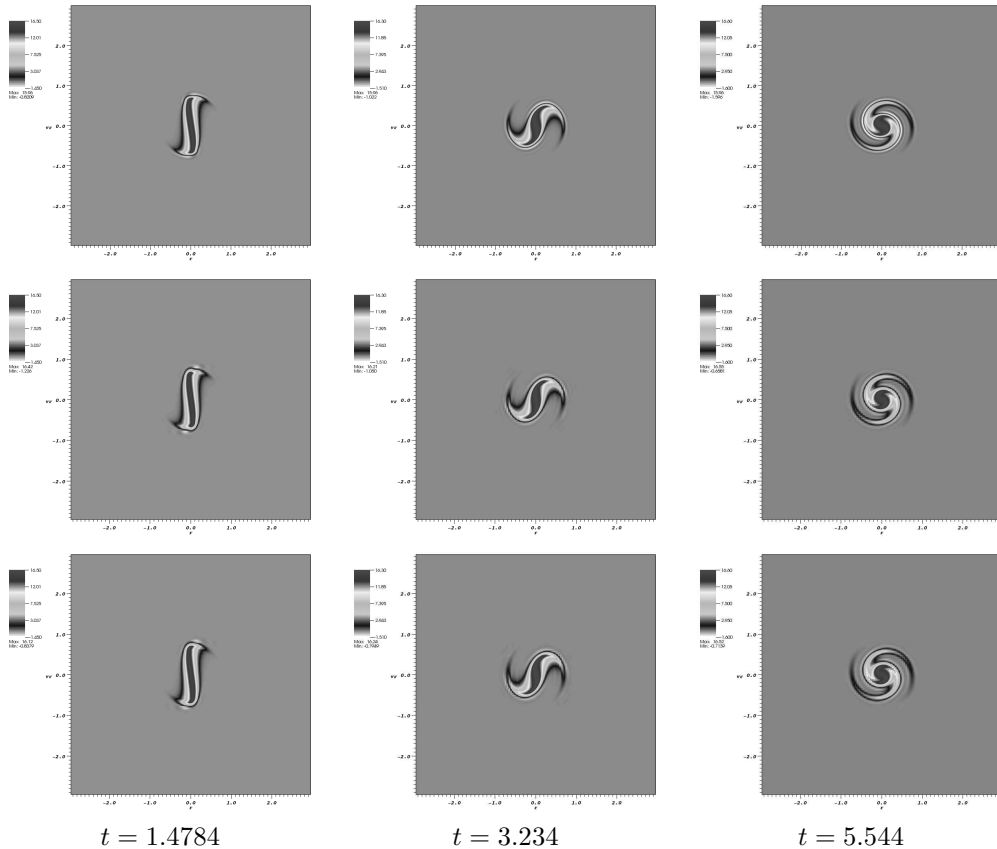


Figure 8: Simulations of type (II') (first row), (III) (second row), and (IV) (third row) for a semi-gaussian beam with  $H_1(\tau) = \cos(\tau)$  and  $\omega_1 = 4\sqrt{2}$ .

In this paragraph, we do not assume that the self-consistent electric field vanishes. Then, in most of cases, we are not able to find an analytic expression of the solution  $G$  of (2.12). So, in order to validate our two-scale methods, we have to compare their results on such a case to the results produced by a classical semi-lagrangian method on (1.1) with the same initial data, and pay attention to the development of thin structures within the beam. For that, we consider the simulations (I), (II'), (III) and (IV), where the simulation type (II') corresponds to a classical semi-lagrangian method on the system (1.1) with  $P_r = P_{v_r} = 256$  and  $r = v_r = 3$ . Since the

simulation (I) already gives bad results on linear cases, it is not really useful to present its results in terms of quality on non-linear cases. However, we have to pay attention to its CPU time cost in order to compare it to the other simulation types ones<sup>1</sup>.

In a first case, we suppose that  $H_1(\tau) = \cos(\tau)$ ,  $\omega_1 = 4\sqrt{2}$ ,  $\epsilon = 10^{-2}$ , and  $f_0$  is given by (4.2) with  $r_m = 0.75$  and  $v_{th} = 0.1$ . In order to guarantee the robustness of the two-scale schemes, we suppose that the time step  $\Delta t_H$  is computed by using the formula (3.41) with  $K = 5$ , giving us  $\Delta t_H \approx 0.0185$ , whereas the time step  $\Delta t_{NH}$  for the simulations (I) and (II') is of the form  $\frac{\Delta t_H}{N}$  with  $N$  large enough in order to verify the condition (3.18). In Figure 8, we observe the results obtained with simulations of type (II'), (III), and (IV), and we can find some results in terms of CPU time in the Table 1.

In a second case, we suppose that  $H_1(\tau) = \cos^2(\tau)$ ,  $\omega_1 = 2$ ,  $\epsilon = 10^{-2}$ , and  $f_0$  is given by (4.2) with  $r_m = 0.75$  and  $v_{th} = 0.1$ . In order to guarantee the robustness of the two-scale schemes, we suppose that  $\Delta t_H$  is defined by (3.41) with  $K = 2$ , which gives us  $\Delta t_H \approx 7.392 \times 10^{-3}$ , and  $\Delta t_{NH} = \frac{\Delta t_H}{N}$  with  $N$  large enough. In Figure 9, we observe the results obtained with simulations of type (II'), (III), and (IV) and CPU times costs for each simulation can be found in the Table 1.

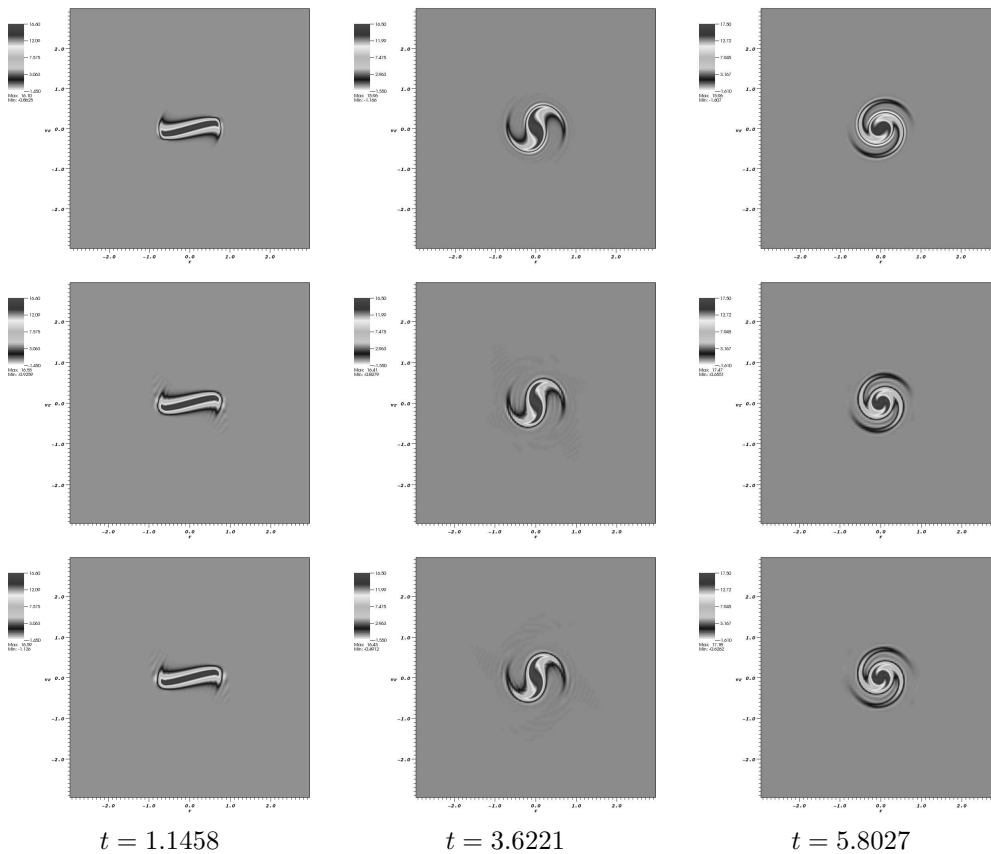


Figure 9: Simulations of type (II') (first row), (III) (second row), and (IV) (third row) for a semi-gaussian beam with  $\omega_1 = 2$ ,  $H_1(\tau) = \cos^2(\tau)$ .

As we can see in Figures 8 and 9, the three simulations (II'), (III) and (IV) produce results of the same quality, even if the mesh in  $r$  and  $v_r$  directions used in the classical semi-lagrangian method is much more refined than the one used for the two-scale simulations, which enlarge to non-linear cases the conclusion we have established at the end of the previous paragraph.

<sup>1</sup>All the simulations have been conducted on a Sun Fire X4600 server with an AMD Optreron 8220 processor (Dual Core 3000 Mhz) under SunOS 5.10 system.

Case	Simulation (I)		Simulation (II')		Simulation (III)	Simulation (IV)
	CPU time	$N$	CPU time	$N$	CPU time	CPU time
$\omega_1 = 4\sqrt{2}, H_1 = \cos$	35m	122	35h 6m 50s	480	1h 43m 39s	55m 3s
$\omega_1 = 2, H_1 = \cos^2$	37m 32s	49	38h 7m 6s	192	5h 45m 25s	2h 37m 25s

Table 1: CPU time costs: the final time is  $T = 6.93$  for the case where  $\omega_1 = 4\sqrt{2}$  and  $H_1 = \cos$ , and  $T = 6.9854$  for the case where  $\omega_1 = 2$  and  $H_1 = \cos^2$ .

Furthermore if we observe the CPU times costs of each simulations (see Table 1), we remark that both two-scale numerical methods are much slower than the classical semi-lagrangian method when they are ran on a same mesh in  $r$  and  $v_r$  directions. It is not surprising because of the high number of fixed point problems we have to solve during the two-scale simulations. But on the other hand, if we compare the CPU time costs of the simulations (II'), (III) and (IV) which give the same quality of results, we remark that both two-scale numerical methods we have described are much faster than a precise classical semi-lagrangian method. One more time, this phenomenon can be explained by the condition (3.18) imposed to the time step  $\Delta t_{NH}$  within the classical method: if we refine the mesh of the phase space, we diminish the time step, and then increase the number of iterations in time we need to reach the the final time of the simulation. As a conclusion, we can say that if we want high quality results, it is preferable in terms of CPU time cost to run one of the two-scale methods we have developed instead of the classical method.

In a last case, we suppose that  $H_1(\tau) = \cos^2(\tau)$ ,  $\omega_1 = 1$ ,  $\epsilon = 10^{-2}$ , and  $f_0$  is given by (4.2) where  $n_0 = 4$ ,  $r_m = 1.85$  and  $v_{th} = 0.1$ . As in the previous tests, we suppose that the time step  $\Delta t_H$  for the two-scale methods is given by (3.41) where  $K = 1$ , and that the time step  $\Delta t_{NH}$  for the classical semi-lagrangian method is of the form  $\frac{\Delta t_H}{N}$  with  $N$  large enough. The goal of this numerical experiment is to observe the same structures as Frénod, Salvarani and Sonnendrücker have observed in [13]. Since the structures we want to observe are quite thin, we consider the simulation types (II''), (III') and (IV')

- simulation (II''): we solve the system (1.1) with a classical semi-lagrangian method, with  $P_r = P_{v_r} = 256$  and  $R = v_R = 3$ ,
- simulation (III'): we solve the system (2.12) with a two-scale semi-lagrangian method on a two-scale mesh, with  $P_r = P_{v_r} = 128$ ,  $P_\tau = 20$  and  $R = v_R = 3$ ,
- simulation (IV'): we solve the system (2.12) with a two-scale semi-lagrangian method on a uniform mesh, with  $P_q = P_{u_r} = 256$ ,  $P_\tau = 20$ , and  $R = v_R = 3$ , and we compute the approximation (3.28) on a uniform  $257 \times 257$  grid in  $(r, v_r)$  on  $[-R, R] \times [-v_R, v_R]$ .

In the Figure 10, we can observe some results obtained with the simulations (II''), (III') and (IV'). One more time, we can remark that the two-scale methods do not need a mesh as refined as the classical method's one for producing high quality results. Furthermore, we can observe in the last column of the Figure 10 that the beam simulated with the classical semi-lagrangian method becomes unfocused in long time, even if we consider a highly refined mesh. This remark is confirmed by the Figure 11 where we notice the thin structures are better determined with a two-scale method. Moreover, the result given by the classical method is slightly out of phase and more diffusive. One more time, the main reason of this problem is the condition (3.18) imposed on the time step in the classical semi-lagrangian simulation: if we consider a  $513 \times 513$  grid in  $(r, v_r)$ , this condition induces a so small value of  $\Delta t_{NH}$  (nearly  $3.84 \times 10^{-5}$ ) that the numerical diffusion introduced by the interpolations makes the beam unfocused in a long time simulation. On the other hand, the two-scale results match well with the expected long time behavior as described in [13].

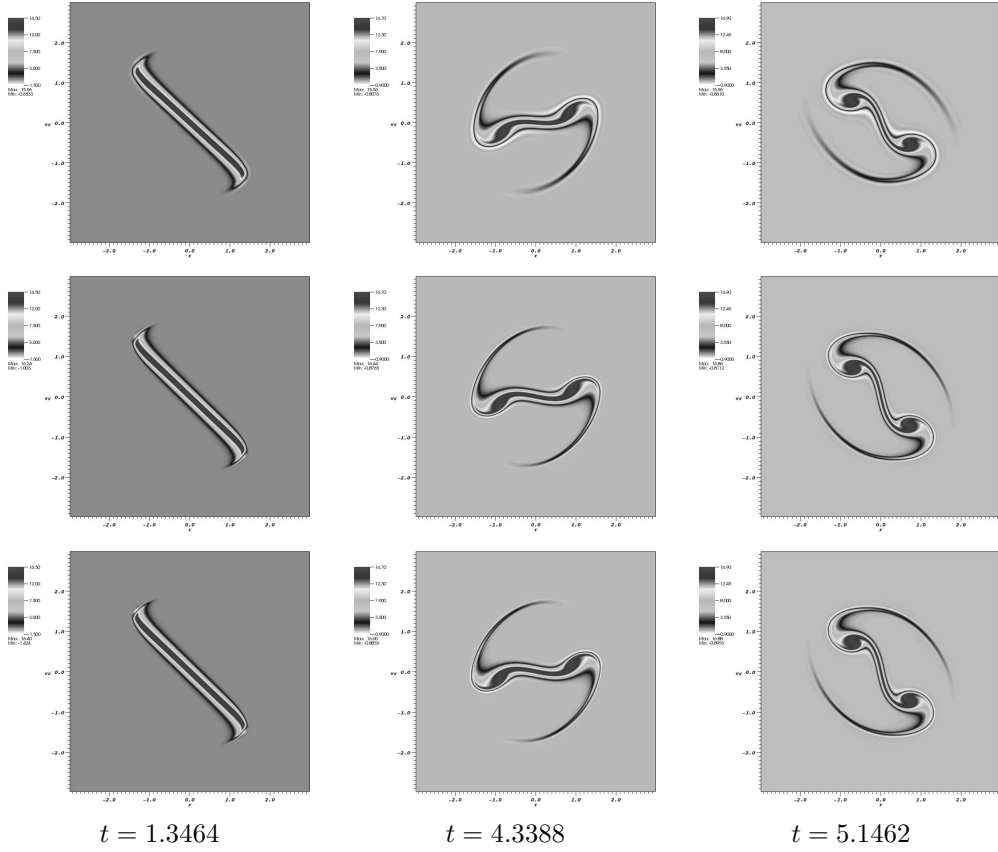


Figure 10: Simulations of type (II'') (first row), (III') (second row) and (IV') (third row) for a semi-gaussian beam with  $\omega_1 = 1$ ,  $H_1(\tau) = \cos^2(\tau)$  and  $r_m = 1.85$ .

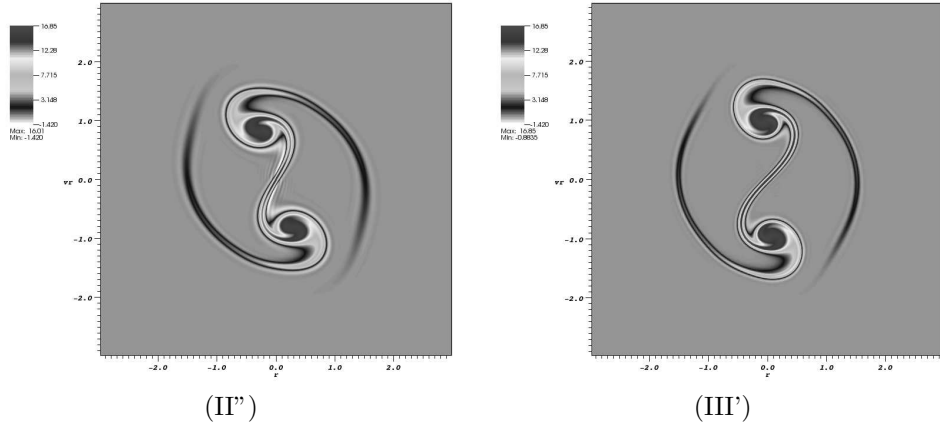


Figure 11: Simulations of type (II'') and (III') for a semi-gaussian beam at time  $t = 5.984$  with  $\omega_1 = 1$ ,  $H_1(\tau) = \cos^2(\tau)$  and  $r_m = 1.85$ .

## 5 Conclusion and perspectives

We have built a two-scale semi-lagrangian method and proposed a new mesh in order to simplify the computation of the electric field and, which leads to another two-scale semi-lagrangian method. These methods have been tested on non-smooth initial conditions, especially semi-gaussian beam initial conditions: on linear cases, we have concluded that these two-scale methods are very efficient,

even if we consider a coarse mesh in  $r$  and  $v_r$  directions, contrary to a classical semi-lagrangian scheme on the model (1.1) which needs much more points to produce good results. On non-linear cases, we have reached the same conclusion for short time simulation, not only in terms of quality of results, by also in terms of CPU time costs. These results are very promising for extensions to higher dimensional problems such as the two-scale limit models obtained by Frnod and Sonnendrecker in [10, 11, 12], the finite Larmor radius approximation obtained in [4], or other charged particle beam problems which cannot allow any time splitting. Furthermore, for long time simulation, both two-scale numerical methods we have developed give very good results, contrary to the classical semi-lagrangian method which is so penalized by its numerical diffusion for long time simulations that it produces results which do not correspond to the expected behavior. These results are also promising since they consolidate the conclusions of Frnod, Salvarani and Sonnendrücker in [13], which are that a two-scale numerical method can be successfully used in a context of non-smooth initial data.

Finally, we also have remarked that, even if they are faster than the classical semi-lagrangian method for obtaining the same quality of results, both two-scale numerical methods need a very high CPU time cost. Since this time is essentially spent for computing the fixed point problems within the methods, it can be interesting to find a way to improve this part of the methods.

**Acknowledgements** The author wishes to thank E. Frénod and E. Sonnendrücker for the discussions on the topic.

## References

- [1] P. AILLIOT, E. FRÉNOD, AND V. MONBET, *Long term object drift forecast in the ocean with tide and wind*, Multiscale Model. Simul., **5-2** (2006), 514-531.
- [2] G. ALLAIRE, *Homogenization and two-scale convergence*, SIAM J. Math. Anal., **23-6** (1992), 1482-1518.
- [3] C. K. BIRDSALL, AND A. B. LANGDON, *Plasma physics via computer simulation*, Inst. of Phys. Publishing, Bristol/Philadelphia (1991).
- [4] M. BOSTAN, *The Vlasov-Poisson system with strong external magnetic field. Finite Larmor radius regime. Rapport de recherche de l'INRIA 6159*, (2007).
- [5] C. Z. CHENG, AND G. KNORR, *The integration of the Vlasov equation in configuration space*, J. Comp. Phys., **22-3** (1976), 330-351.
- [6] P. DEGOND, AND P.-A. RAVIART, *On the paraxial approximation of the stationary Vlasov-Maxwell system*, Math. Models Methods Appl. Sci., **3-4** (1993), 513-562.
- [7] F. FILBET, AND E. SONNENDRÜCKER, *Comparison of Eulerian Vlasov solvers*, Comp. Phys. Comm., **151** (2003), 247-266.
- [8] F. FILBET, AND E. SONNENDRÜCKER, *Modeling and numerical simulation of space charge dominated beams in the paraxial approximation*, Math. Models Methods Appl. Sci., **16-5** (2006), 763-791.
- [9] E. FRÉNOD, A. MOUTON, AND E. SONNENDRÜCKER, *Two-scale numerical simulation of the weakly compressible 1D isentropic Euler equations*, Numer. Math., **108-2** (2007), 263-293.
- [10] E. FRÉNOD, AND E. SONNENDRÜCKER, *Homogenization of the Vlasov equation and of the Vlasov-Poisson system with a strong external magnetic field*, Asymp. Anal., **18-3-4** (1998), 193-214.
- [11] E. FRÉNOD, AND E. SONNENDRÜCKER, *Long time behavior of the two-dimensional Vlasov equation with a strong external magnetic field*, Math. Models Methods Appl. Sci., **10-4** (2000), 539-553.

- [12] E. FRÉNOD, AND E. SONNENDRÜCKER *The finite Larmor radius approximation*, SIAM J. Math. Anal., **32**-6 (2001), 1227-1247.
- [13] E. FRÉNOD, F. SALVARANI, AND E. SONNENDRÜCKER, *Long time simulation of a beam in a periodic focusing channel via a two-scale PIC-method*, ArXiv: 0710.3983v1, to appear in Math. Models Methods Appl. Sci.
- [14] V. GRANDGIRARD, M. BRUNETTI, P. BERTRAND, N. BESSE, X. GARBET, P. GHENDRIH, G. MANFREDI, Y. SARAZIN, O. SAUTER, E. SONNENDRÜCKER, J. VACLAVIK, AND L. VILLARD, *A drift-kinetic semi-lagrangian 4D code for ion turbulence simulation*, J. Comp. Phys., **217** (2006), 395-423.
- [15] I. M. KAPCHINSKY, AND V. V. VLADIMIRSKY, *Proceedings of the 9th international conference on high energy accelerators*, edited by L. Kowarski (CERN, Geneva) (1959), 274-.
- [16] J. LANG, W. CAO, W. HUANG, AND R. D. RUSSELL, *A two-dimensional moving finite element method with local refinement based on a posteriori error estimates*, Appl. Numer. Math., **46**-1 (2003), 75-94.
- [17] G. NGUETSENG, *A general convergence result for a functional related to the theory of homogenization*, SIAM J. Math. Anal., **20**-3 (1989), 608-623.
- [18] M. REISER, *Theory and design of charged particle beams*, Wiley, New-York (1994).
- [19] E. SONNENDRÜCKER, J.-R. ROCHE, P. BERTRAND, AND A. GHIZZO, *The semi-lagrangian method for the numerical resolution of Vlasov equations*, J. Comp. Phys., **149**-2 (1999), 201-220.

E-mail adress: [mouton@math.u-strasbg.fr](mailto:mouton@math.u-strasbg.fr)