An exponential integrator for a highly oscillatory Vlasov equation

Emmanuel Frenod, Sever Adrian Hirstoaga, Eric Sonnendrücker

▶ To cite this version:

An exponential integrator for a highly oscillatory Vlasov equation

Emmanuel Frénod*   Sever A. Hirstoaga†   Eric Sonnendrücker‡

Abstract

In the framework of a Particle-In-Cell scheme for some 1D Vlasov-Poisson system depending on a small parameter, we propose a time-stepping method which is numerically uniformly accurate when the parameter goes to zero. Based on an exponential time differencing approach, the scheme is able to use large time steps with respect to the typical size of the fast oscillations of the solution.

1   Introduction

The problem of studying a charged particle beam focused by external electric or magnetic fields is important in many applications. The motion of such particle beams is governed by the interactions between the electric field generated by the particles themselves and by the external focusing electromagnetic field. The modelling framework consists in coupling a kinetic equation with Maxwell equations. Disregarding the collisions between particles, the kinetic modelling is performed by means of the Vlasov equation. In this paper we will consider only non-relativistic long and thin beams. Therefore, instead of studying the phenomenon by means of the full Vlasov-Maxwell system, we can use its paraxial approximation (see [4] for mathematical modelling and numerical simulation of focused particle beams dynamics). In this framework, the effects of the self-consistent magnetic and electric fields can be both taken into account by solving a single Poisson equation. Finally, we are led to solve a two dimensional phase space Vlasov-Poisson system with a parameter ε. This small parameter acts on the time variable (in fact the longitudinal variable of a thin beam, see [6] for details and physical meaning of the parameter), producing the highly oscillatory behaviour in phase space. In this framework, the aim of this paper is to propose a numerical scheme able to study efficiently the evolution of a beam over a large number of fast periods.

Although more precise alternatives exist, we have chosen in this work to perform the numerical solution of Vlasov equation by particle methods, which consist in approximating the distribution function by a finite number of macroparticles. The trajectories of these particles are computed from the characteristic curves of the Vlasov equation, whereas the self-consistent electric field is computed on a mesh in the physical space. This method allows to obtain satisfying results with a small number of particles (see [1]). The contribution of this paper is to propose a new numerical method for solving the characteristic curves, or equivalently, for computing the

---

*Université Européenne de Bretagne, LMBA (UMR CNRS 6205), Université de Bretagne-Sud & Inria Nancy-Grand Est, CALVI Project & IRMA (UMR CNRS 7501), Université de Strasbourg, France

†Inria Nancy-Grand Est, CALVI Project & IRMA (UMR CNRS 7501), Université de Strasbourg, France

‡Max Planck Institute for Plasma Physics, EURATOM Association, Boltzmannstr. 2, 85748 Garching, Germany

macroparticles' trajectories. Namely, we are faced with solving the following stiff differential equation

$$u'(t) = u(t)/\varepsilon + F(t, u(t)), \qquad u(0) = u_0, \tag{1.1}$$

for several small values of the parameter ε and where F represents a nonlinear term which plays the role of the self-consistent electric field. The difficulty arising in the numerical solution for this equation relies on the ability of the scheme to be uniformly stable and accurate when ε goes to zero. Following the survey article [7], we are encouraged to use efficient numerical methods like exponential integrators in order to describe the dynamics of equation (1.1). The basic idea behind these methods is to make use of the solution's exact representation given by the variation-of-constants formula

$$u(t) = e^{t/\varepsilon} u_0 + \int_0^t e^{(t-\tau)/\varepsilon} F(\tau, u(\tau)) \, d\tau. \tag{1.2}$$

Applying this formula from one time step to another has the merit to solve exactly the linear (stiff) part. Classical numerical schemes fail to capture the stiff behaviour regardless of the size of the time step with respect to the small parameter. One may consult [7] for construction, mathematical analysis and implementation of exponential integrators in the two classical types of stiff problems encapsulated in equation (1.1). As a specific implementation, one may cite [9], where, in the context of laser-plasma interactions, an exponential integrator is used for a Particle-In-Cell (PIC) method in order to model the high-frequency plasma response. Once the stiff part is exactly solved, one may use an exponential time differencing (ETD) method (see [3]) for the specific numerical treatement of the nonlinear term. The ETD schemes turn out to outperform many other schemes when treating problems like (1.1); see [3, 8] for comparisons of ETD against the Implicit-Explicit method, the Integrating Factor method, etc.

In the present paper we construct and implement an exponential integrator in order to solve the characteristics of the highly oscillatory Vlasov-Poisson problem (2.2). The aim is to use a scheme with large time steps compared to the fast period that arises from the linear term without loosing the accuracy when the small parameter vanishes. The novelty of this method is in the numerical approximation of the integral term in (1.2). More precisely, when the time step is much larger than the rapid period, the idea of the algorithm is the following: we first finely solve the ODEs over one fast period by means of a high-order solver (we have used explicit 4th order Runge-Kutta). Then, thanks to formula (1.2), we may compute an approximation of the solution over a large integral number of periods. We have also found that using a more accurate period, instead of the period of the solution to the system (1.1) without the nonlinear term, leads to more accurate simulations. In addition, we have checked if the scheme gives accurate solutions starting with an initial condition which lies on the slow manifold or not. We cite [2] for a "definition" of the slow manifold: "The slow manifold is that particular solution which varies only on the slow time scale; the general solution to the ODE contains fast oscillations also."

The remainder of the paper is organized as follows. Following [6], we briefly recall in Section 2 the paraxial approximation together with the axisymmetric beam assumption. In Section 3 we describe the PIC method for the Vlasov-Poisson system in which we are interested. Then, Section 4 is devoted to the construction of the new numerical scheme as an exponential integrator for solving the time-stepping part of the PIC algorithm. Eventually, in Section 5, we implement and test our method on several test cases related to the Vlasov-Poisson system.

2 The paraxial model: an axisymmetric Vlasov equation

The paraxial approximation relies on a scaled Vlasov-Poisson system in a phase space of dimension four, $2D$ for space variable $\mathbf{x}$ and $2D$ for velocity variable $\mathbf{v}$. This simplified model of the full Vlasov-Maxwell system is particularly adapted to the study of long and thin beams of charged particles and it describes their evolution in the plane transverse to their direction of propagation.

Subject of many research investigations, the paraxial model was derived in a number of papers, see e.g. [4, 6]. In this work we are interested in solving numerically a paraxial model with some additional hypotheses, see (2.2) below. The solution of system (2.2) is represented by a beam of particles in phase space. The beam evolves by rotating around the origin in the phase space, and in long times a bunch forms around the center of the beam from which filaments of particles are going out. These filaments are difficult to capture with classical numerical methods.

We now introduce the paraxial model that we aim to solve. In the additional axisymmetric beam assumption (i.e. invariant beam under rotation in $\mathbb{R}^2$ for $\mathbf{x}$), we are led to change the $\mathbf{x}$ coordinate in the polar frame. We thus write the model in polar coordinates (r, θ), where $r = |\mathbf{x}|$ and $\theta \in [0, 2\pi)$ is such that $x_1 = r \cos\theta$ and $x_2 = r \sin\theta$. Then we use new velocity variables $v_r = \mathbf{v} \cdot \mathbf{x}/r$ and $v_\theta = \mathbf{v} \cdot \mathbf{x}^\perp/r$, where $\mathbf{x}^\perp = (-x_2, x_1)$. Assuming in addition, as in [6], that f_ε is concentrated in angular momentum, i.e. $r v_\theta = 0$, the paraxial model becomes

$$
\begin{cases}
\dfrac{\partial f_\varepsilon}{\partial t} + \dfrac{1}{\varepsilon} v_r \dfrac{\partial f_\varepsilon}{\partial r} + \left(\mathbf{E}_r^\varepsilon + \Xi_r^\varepsilon \right) \dfrac{\partial f_\varepsilon}{\partial v_r} = 0, \\[2ex]
\dfrac{1}{r} \dfrac{\partial (r\, \mathbf{E}_r^\varepsilon)}{\partial r} = \rho_\varepsilon, \quad \rho_\varepsilon(t, r) = \displaystyle\int_{\mathbb{R}} f_\varepsilon(t, r, v_r)\, dv_r, \\[2ex]
f_\varepsilon(t = 0, r, v_r) = f_0(r, v_r),
\end{cases}
\tag{2.1}
$$

where the external force Ξ_r^ε writes

$$
\Xi_r^\varepsilon(t, r) = \left(-\frac{1}{\varepsilon} H_0 + H_1\left(\frac{t}{\varepsilon}\right) \right) r,
$$

with $H_0 > 0$ and $H_1(t)$ some 2π-periodic function with zero mean value. In this paper we assume that there is no time oscillation in the external field but only the strong uniform focusing. We thus take $H_0 = 1$ and $H_1 = 0$ and the Vlasov-Poisson system in which we are interested writes

$$
\begin{cases}
\dfrac{\partial f_\varepsilon}{\partial t} + \dfrac{1}{\varepsilon} v_r \dfrac{\partial f_\varepsilon}{\partial r} + \left(\mathbf{E}_r^\varepsilon - \dfrac{r}{\varepsilon} \right) \dfrac{\partial f_\varepsilon}{\partial v_r} = 0, \\[2ex]
\dfrac{1}{r} \dfrac{\partial (r\, \mathbf{E}_r^\varepsilon)}{\partial r} = \displaystyle\int_{\mathbb{R}} f_\varepsilon(t, r, v_r)\, dv_r, \\[2ex]
f_\varepsilon(t = 0, r, v_r) = f_0(r, v_r).
\end{cases}
\tag{2.2}
$$

In order to test the numerical method that we propose, we first consider two numerically simpler test cases where the electric field is not issued from Poisson equation, but it has analytical forms: in the first case $\mathbf{E}_r^\varepsilon(t, r) = -r$ and in the second one $\mathbf{E}_r^\varepsilon(t, r) = -r^3$. Unlike the second case, for the first one, we can analytically compute the solution to (2.2)(a).

3   A Particle-In-Cell method for Vlasov-Poisson system

We solve the Vlasov-Poisson system (2.2) by using a Particle-In-Cell (PIC) method [1]. We thus introduce the following Dirac mass sum approximation of f_ε

$$f_\varepsilon^{N_p}(r, v, t) = \sum_{k=1}^{N_p} w_k \delta(r - R_k(t)) \, \delta(v - V_k(t)),$$

where N_p is the number of macroparticles and $\big(R_k(t), V_k(t)\big)$ is the position in phase space of macroparticle k moving along a characteristic curve of equation (2.2)(a). Therefore, the problem is to find the positions and velocities (R_k^{n+1}, V_k^{n+1}) at time t_{n+1} from their values at time t_n, by solving

$$\left\{ \begin{array}{l} R'(t) = V(t)/\varepsilon, \\ V'(t) = -R(t)/\varepsilon \, + \, E_\varepsilon(t, R(t)), \\ R(t_n) = R_k^n, \, V(t_n) = V_k^n. \end{array} \right. \tag{3.1}$$

In this case, the standard PIC algorithm writes as follows: (1) deposit particles on a spatial grid, leading to the grid density; (2) solve Poisson equation on the grid, leading to the grid electric field; (3) interpolate the grid electric field in each particle; (4) push particles with the previously obtained electric field.

The first three steps are classically treated. The first one deals with the computing of the grid density by convoluting $\rho_\varepsilon^{N_p}$ defined by

$$\rho_\varepsilon^{N_p}(r, t) = \sum_{k=1}^{N_p} w_k \delta(r - R_k(t)),$$

with a first order spline (this corresponds to the Cloud-in-Cell method in [1]). Then, we solve Poisson equation (2.2)(b) on a uniform one-dimensional grid by using finite differences. In our case this amounts to only discretize some space integral. We have done this by using the trapezoidal rule. As for the third step, we used the same convolution function as for the deposition step in order to get the particle electric field (this corresponds to a linear interpolation of the grid electric field on each cell).

Eventually, the major issue is the fourth step of the PIC algorithm, consisting in the numerical integration of system (3.1). Here is the main focus of this paper, taking into account that we want to propose a stable and accurate scheme using large time steps with respect to the fast oscillation. To this end, we introduce in the next section a method based on exponential time differencing.

4   The exponential integrator for the Particle-In-Cell method

We now describe the exponential numerical integrator that we have implemented to solve the fourth step of the PIC algorithm. We first write down the exponential time differencing (ETD) method in the case of the stiff ODE system we are interested in. Then, in Section 4.2, we develop an algorithm based on the exponential time differencing in the framework of the PIC method.

4.1 Exponential time differencing for a highly oscillatory ODE

The so-called exponential time differencing scheme arose originally in the field of computational electrodynamics but has been reinvented many times over the years (see [3] and the references therein). We take details of the ideas behind the various ETD schemes from the comprehensive paper by Cox and Matthews [3].

Recall the stiff system of ODEs that we have to solve:

$$\begin{cases} R'(t) = \dfrac{1}{\varepsilon} V(t), \\ V'(t) = -\dfrac{1}{\varepsilon} R(t) + E_\varepsilon(t, R(t)), \end{cases} \tag{4.1}$$

with some initial condition (R_0, V_0). In this section we assume that the electric field E_ε is given. As exposed in [3], the stiffness comes from the two scales on which the solution evolves: the rapid oscillations due to the linear term and a slower evolution due to the nonlinear (electric) term. Thus, while any explicit scheme is limited to a small time step, of order ε, a fully implicit one requires nonlinear problems to be solved and is therefore slow. A suitable time-stepping scheme for (4.1) should be able to avoid the small time steps when treating the stiffness. The essence of the ETD methods is to solve the stiff linear part exactly and to derive appropriate approximations when integrating numerically the slower nonlinear term [3, 8].

To derive the exponential time differencing (ETD) method for this system we first apply $r(-t/\varepsilon)$ to (4.1) and then integrate the obtained equation from t_n to $t_{n+1} = t_n + dt$ to deduce that

$$\begin{pmatrix} R_{n+1} \\ V_{n+1} \end{pmatrix} = r\left(\frac{dt}{\varepsilon}\right) \begin{pmatrix} R_n \\ V_n \end{pmatrix} + r\left(\frac{dt}{\varepsilon}\right) \int_{t_n}^{t_{n+1}} r\left(\frac{t_n - \tau}{\varepsilon}\right) \begin{pmatrix} 0 \\ E_\varepsilon(\tau, R(\tau)) \end{pmatrix} d\tau, \tag{4.2}$$

where

$$r(\tau) = \begin{pmatrix} \cos(\tau) & \sin(\tau) \\ -\sin(\tau) & \cos(\tau) \end{pmatrix}. \tag{4.3}$$

This formula is exact. In addition, it is also useful to write (4.2) by replacing (t_n, t_{n+1}) by any couple (s, t) such that $s < t$. More precisely, we have

$$\begin{pmatrix} R(t) \\ V(t) \end{pmatrix} = r\left(\frac{t - s}{\varepsilon}\right) \begin{pmatrix} R(s) \\ V(s) \end{pmatrix} + r\left(\frac{t - s}{\varepsilon}\right) \int_{s}^{t} r\left(\frac{s - \tau}{\varepsilon}\right) \begin{pmatrix} 0 \\ E_\varepsilon(\tau, R(\tau)) \end{pmatrix} d\tau. \tag{4.4}$$

Now the main question is how to derive approximations to the integral term in (4.2). All the ETD schemes are results of this process. In this spirit, it was shown in [3] that ETD methods can be extended to any order by using multistep or Runge-Kutta methods and explicit formulae for such arbitrary order ETD methods were derived. In particular, explicit coefficients for ETD Runge-Kutta methods of order up to four have been computed. The authors also illustrated on several ODEs and PDEs that ETD is superior over the Integrating Factor and Implicit-Explicit methods, two other classical schemes able to avoid the small time step. Nevertheless, in the form written in [3], a high-order ETD scheme (e.g. ETDRK4) suffers from numerical instability as explained in [8]. The problem has been solved in [8] by using a contour integral method for evaluating the coefficients. Then, this modified ETDRK4 scheme has been tested in [8] against five other 4th order schemes on several PDEs, and ETDRK4 has been found the best in terms of errors. These results encouraged us to use an ETD scheme in order to solve stiff ODEs.

In this paper we do not use a high-order ETD method as described in the references above but merely the formula (4.2) for justifying the derivation of Algorithm 4.1. More precisely, we adopt a different approach for approximating the integral term in (4.2), which is justified by the remark in the next paragraph and by our aim to use very large time steps compared to the fast oscillations, say $dt = 100\,\varepsilon$ when $\varepsilon = 10^{-2}$. This is the core of the method described in the following section.
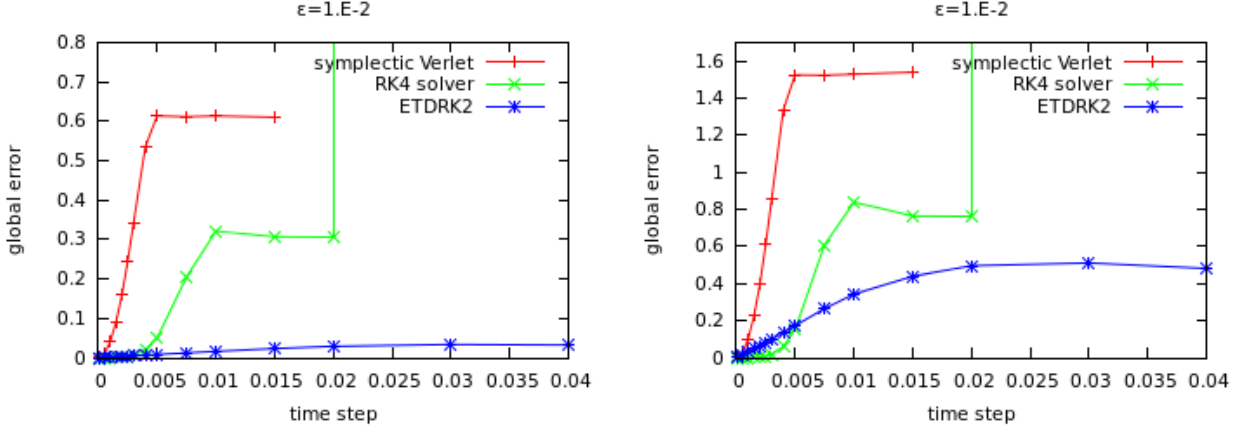

Figure 1: A nonlinear case: the global Euclidean error at time 3.14 for two initial conditions: on (left) and off (right) the slow manifold

However, we have done tests with ETDRK2 (see formula (22) in [3]) against symplectic Verlet and RK4 schemes, in the case $E_\varepsilon(t, R) = -R^3$ with $\varepsilon = 10^{-2}$ (see the test case in Section 5.2). Thus, we have noticed that this ETD scheme has smaller errors for some range for the time step and that, unlike its competitors, it is stable when $dt \geq 2\varepsilon$ (see Fig. 1). But these simulations also show that when dt is too large with respect to ε (e.g. $dt \geq 3\varepsilon$) the global error is significant for some initial condition (particles off the slow manifold, see Section 5). The reason is that for dt large enough with respect to ε, the nonlinear force term cannot be accurately taken into account by any high-order Runge-Kutta solver.

4.2   The ETD PIC method with large time steps

In this section we describe our time-stepping scheme starting from the equation (4.2). The main idea is to remark that the time for one fast grand tour is approximated by $2\pi\varepsilon$, the period of the solution to (4.1) without nonlinear term. Therefore, since we want to build a scheme with a time step dt much larger than the fast oscillation, we first need to find the unique positive integer N and the unique real $\Delta \in [0, 2\pi\varepsilon)$ such that

$$dt = N \cdot (2\pi\varepsilon) + \Delta. \qquad (4.5)$$

Thus the integral term in (4.2) becomes

$$\int_{t_n}^{t_{n+1}} d\tau = \sum_{j=0}^{N-1} \int_{t_n + 2\pi\varepsilon\, j}^{t_n + 2\pi\varepsilon\, (j+1)} d\tau + \int_{t_{n+1} - \Delta}^{t_{n+1}} d\tau, \qquad (4.6)$$

6

that we approximate by

$$N \int_{t_n}^{t_n+2\pi\varepsilon} d\tau + \int_{t_{n+1}-\Delta}^{t_{n+1}} d\tau. \tag{4.7}$$

In this approximation, we make some assumptions that we develop in Remark 4.2. Computing the integrals

$$\mathcal{I}_1 = \int_{t_n}^{t_n+2\pi\varepsilon} r\left(\frac{t_n-\tau}{\varepsilon}\right) \begin{pmatrix} 0 \\ E_\varepsilon(\tau,R(\tau)) \end{pmatrix} d\tau \text{ and } \mathcal{I}_2 = \int_{t_{n+1}-\Delta}^{t_{n+1}} r\left(\frac{t_n-\tau}{\varepsilon}\right) \begin{pmatrix} 0 \\ E_\varepsilon(\tau,R(\tau)) \end{pmatrix} d\tau$$

knowing (R_n, V_n) will thus lead to (R_{n+1}, V_{n+1}). Next we describe this method in 3 steps. The electric field is discretized explicitely, i.e. it can be computed at any time solving the Poisson equation (2.2)(b) with a right-hand side obtained from depositing the particles on the grid.

1st step: Using (4.4) with $s = t_n$ and $t = t_n + 2\pi\varepsilon$ and since $r(2\pi) = \text{Id}$, we have

$$\mathcal{I}_1 = \begin{pmatrix} R(t_n+2\pi\varepsilon)-R(t_n) \\ V(t_n+2\pi\varepsilon)-V(t_n) \end{pmatrix}. \tag{4.8}$$

Now, we finely solve (4.1) with initial conditions (R_n, V_n) by a 4th order Runge-Kutta scheme, in order to get an accurate approximation of $\big(R(t_n+2\pi\varepsilon), V(t_n+2\pi\varepsilon)\big)$. This is done by following the four steps of the standard PIC algorithm as exposed in Section 3. The time step used in the fourth step for the RK4 solver is $\sqrt{\varepsilon}\,\varepsilon$ (we explain this choice in Section 5).

2nd step: We have to calculate an approximation of $\big(R(t_n + N \cdot 2\pi\varepsilon), V(t_n + N \cdot 2\pi\varepsilon)\big)$ since it will be usefull in the computations to do in the 3rd step. Using (4.4) with $s = t_n$ and $t = t_n + N \cdot 2\pi\varepsilon$ we obtain, since $r(N \cdot 2\pi) = \text{Id}$,

$$\begin{pmatrix} R(t_n + N \cdot 2\pi\varepsilon) \\ V(t_n + N \cdot 2\pi\varepsilon) \end{pmatrix} \approx \begin{pmatrix} R_n \\ V_n \end{pmatrix} + N \cdot \mathcal{I}_1, \tag{4.9}$$

where the right integral (the first one in the right-hand side of (4.6)) was approximated as done in (4.7) by $N \cdot \mathcal{I}_1$.

3rd step: Now, we compute $\mathcal{I}_2$ by using (4.4) with $s = t_n + N \cdot 2\pi\varepsilon = t_{n+1} - \Delta$ and $t = t_{n+1}$

$$\mathcal{I}_2 = r\left(-\frac{\Delta}{\varepsilon}\right) \begin{pmatrix} \widetilde{R}(t_{n+1}) \\ \widetilde{V}(t_{n+1}) \end{pmatrix} - \begin{pmatrix} R(t_n + N \cdot 2\pi\varepsilon) \\ V(t_n + N \cdot 2\pi\varepsilon) \end{pmatrix}, \tag{4.10}$$

where $\big(\widetilde{R}(t_{n+1}), \widetilde{V}(t_{n+1})\big)$ may be found as done above for the approximation of $\big(R(t_n + 2\pi\varepsilon), V(t_n + 2\pi\varepsilon)\big)$: we follow the steps of the standard PIC algorithm, using for the particles' push a 4th order Runge-Kutta solver with initial conditions $(R(t_n + N \cdot 2\pi\varepsilon), V(t_n + N \cdot 2\pi\varepsilon))$ and with total time Δ (as for the 1st step above, we choose a small time step, $\sqrt{\varepsilon}\,\varepsilon$).

Finally, we replace (4.9) in (4.10) which we put in (4.2): the term $N \cdot \mathcal{I}_1$ will cancel and at the end we have

$$R_{n+1} = \widetilde{R}(t_{n+1}) \quad \text{and} \quad V_{n+1} = \widetilde{V}(t_{n+1}),$$

meaning that the vector $\big(\widetilde{R}(t_{n+1}), \widetilde{V}(t_{n+1})\big)$ calculated within the 3rd step above is an approximation of the solution at time t_{n+1}.

In summary, writing dt as in (4.5), the implementation of our time-stepping algorithm follows three steps:

Algorithm 4.1. *Assume that (R_n, V_n) the solution of (4.1) at time t_n is given. Then*

1. *compute (R, V) at time $t_n + 2\pi\varepsilon$ by using a fine Runge-Kutta solver with initial condition (R_n, V_n).*

2. *compute (R, V) at time $t_n + N \cdot 2\pi\varepsilon$ by the following rule*

$$\begin{pmatrix} R(t_n + N \cdot 2\pi\varepsilon) \\ V(t_n + N \cdot 2\pi\varepsilon) \end{pmatrix} = \begin{pmatrix} R_n \\ V_n \end{pmatrix} + N \begin{pmatrix} R(t_n + 2\pi\varepsilon) - R_n \\ V(t_n + 2\pi\varepsilon) - V_n \end{pmatrix}. \tag{4.11}$$

3. *compute (R, V) at time t_{n+1} by using a fine Runge-Kutta solver with initial condition (R, V) obtained at the previous step.*

In the framework of the PIC method, Algorithm 4.1 is applied to each particle.

We will call the *modified ETD-PIC scheme*, the Algorithm 4.1 where $2\pi\varepsilon$ is replaced in the first two steps with a more accurate fast time for particles. The reason for this choice is explained in Sections 5.1 and 5.4.

Remark 4.2. *In the approximation (4.7), we have made the assumptions that the solution's period does not change significantly in time and the same for the electric field E_ε. Nevertheless, although in this section E_ε in supposed to be given, in the case of the Vlasov-Poisson coupling (2.2), E_ε is the self-consistent electric field. Therefore, we expect an almost periodic trajectory of the charged particles to involve a similar time behaviour for the electric field that they generate. Thus, in such a case, we make only the assumption that the particles' period does not vary significantly in time.*

5   Validation of the numerical method

We now validate our algorithms in the test cases presented in Section 2. We illustrate their numerical performance and desired properties exposed in the Introduction with global error curves. Thus, we have checked if the method is numerically stable and accurate when ε becomes smaller. Since we solve a stiff system, we compute the errors for both types of initial particles, on and off the slow manifold (see [2, 3]). In fact, for numerical reasons, we should rather use the designations "close or far from the slow manifold" than "on or off the slow manifold". Nevertheless, in order to be in line with the literature we cite, we keep in this paper the designations "on or off the slow manifold".

In order to solve the Vlasov-Poisson model (2.2), we choose as initial distribution function the following semi-Gaussian beam (see [6])

$$f_0(r, v) = \frac{n_0}{\sqrt{2\pi}\, v_{\text{th}}} \exp\left(-\frac{v^2}{2v_{\text{th}}^2}\right)\chi_{[-0.75, 0.75]}(r), \tag{5.1}$$

where the thermal velocity is $v_{\text{th}} = 0.0727518214392$ and $\chi_{[-0.75, 0.75]}(r) = 1$ if $r \in [-0.75, 0.75]$ and 0 otherwise. We implement this distribution using a particle approximation, with $N_p = 10000$ macroparticles with equal weights $w_k = 1/N_p$.

5.1 A linear case

As a first test of validation of our scheme we consider a case where the solution is analytically known. It is the linear case where the self-consistent electric field in (4.1) is given by $E_\varepsilon(t, R) = -R$. The solution satisfying the initial condition $R(s) = R_s$, $V(s) = V_s$ is

$$\begin{cases} R(t) = V_s \dfrac{1}{\sqrt{1+\varepsilon}} \sin\left(\dfrac{\sqrt{1+\varepsilon}}{\varepsilon}(t-s)\right) + R_s \cos\left(\dfrac{\sqrt{1+\varepsilon}}{\varepsilon}(t-s)\right) \\[3mm] V(t) = V_s \cos\left(\dfrac{\sqrt{1+\varepsilon}}{\varepsilon}(t-s)\right) - R_s \sqrt{1+\varepsilon} \sin\left(\dfrac{\sqrt{1+\varepsilon}}{\varepsilon}(t-s)\right) \end{cases} \tag{5.2}$$

for all $t \geq s$. We note that the trajectory in the phase space is an ellipse since we have

$$\forall t \geq s \qquad R(t)^2 + \left(\dfrac{V(t)}{\sqrt{1+\varepsilon}}\right)^2 = R_s^2 + \left(\dfrac{V_s}{\sqrt{1+\varepsilon}}\right)^2.$$

In this particular case, we can compute exactly the time for one rapid grand tour. It is

$$t_p = \dfrac{2\pi\varepsilon}{\sqrt{1+\varepsilon}} \tag{5.3}$$

and this value (and not $2\pi\varepsilon$) was used in our simulation for the push of particles within Algorithm 4.1. In fact, using $2\pi\varepsilon$ instead of the right rapid time makes our method unstable when the final time is large enough. The reason is that $2\pi\varepsilon - t_p$ is big enough so that the accumulated in time errors issued from the rule (4.11) lead the particles to drift outward in the phase space. We also notice that, unlike general case, the rapid time does not depend on the initial condition. Thus, the beam of particles is rotating in the phase space without spiraling. This was checked numerically for our scheme.

In addition, in this case, the slow manifold can easily be determined. Knowing that it is that particular solution which varies only on the slow time scale ([2]), we deduce that the fast oscillations in the solution given by (5.2) can be removed when $R_0 = 0$ and $V_0 = 0$. The stationary solution $(R, V) = (0, 0)$ is the slow manifold in this particular case. Numerically, on and off the slow manifold particles were arbitrarly chosen with $R_0 = 0.306825$ and $V_0 \sim 7 \cdot 10^{-6}$ (close to $(0,0)$) and with $R_0 = 0.748725$ and $V_0 \sim 0.142892$, respectively.

Eventually, this simple case allows us to remark the following. When computing a reference solution, we push the particles with explicit 4th order Runge-Kutta solver using a sufficiently small time step. As a result of several numerical experiments, we found the quite optimal $dt = \sqrt{\varepsilon}\,\varepsilon$. More precisely, we have first noticed the need for the choice of a uniformly small ratio dt/ε with respect to ε in order to get an accurate solution. Indeed, we have computed a reference solution with a time step $dt = \varepsilon/10$, when $\varepsilon = 10^{-5}$. The outcome of this simulation at final times $t = 1.0$ and $t = 3.5$ is illustrated in Fig. 2, showing that the time step $dt = \varepsilon/10$ is not sufficiently small so that good accuracy for the reference solution be reached. Second, we have experimented different uniformly small time steps, $dt = \varepsilon^{5/4}$, $\varepsilon^{3/2}$, $\varepsilon^{7/4}$, ε^2. Thus, computing solutions with time steps $dt \leq \varepsilon^{7/4}$ do not lead to considerably smaller error, for small ε, and in addition it requires large CPU time. We have finally declared acceptable, at worst of order 10^{-5}, the errors obtained with $dt = \varepsilon^{3/2}$, and therefore, this time step was chosen in the following test cases when computing the reference solution.
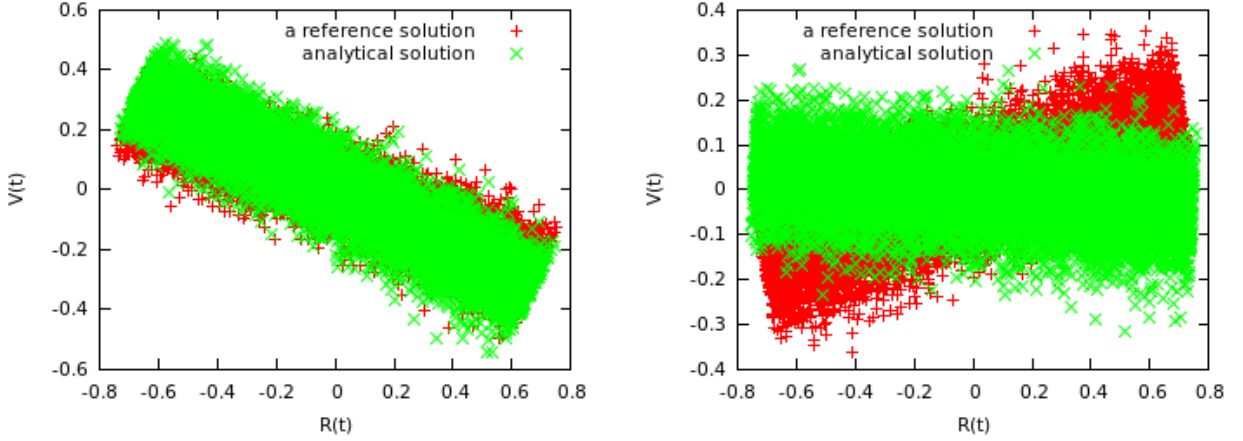
Figure 2: Computing a reference solution in phase space: use of an explicit Runge-Kutta solver with $dt = \varepsilon/10$ when $\varepsilon = 10^{-5}$ at final time 1.0 (left) and 3.5 (right)

5.2 A nonlinear case

We now take the case of $E_\varepsilon(t, R) = -R^3$. The solution to system (4.1) can be writen in terms of Jacobi elliptic functions and it cannot be put in an analytical form. We therefore compute a reference solution using a fine explicit Runge-Kutta solver. Unlike the nonlinear case in Section 5.3, we do not have numerical errors due to the computation of the electric field.
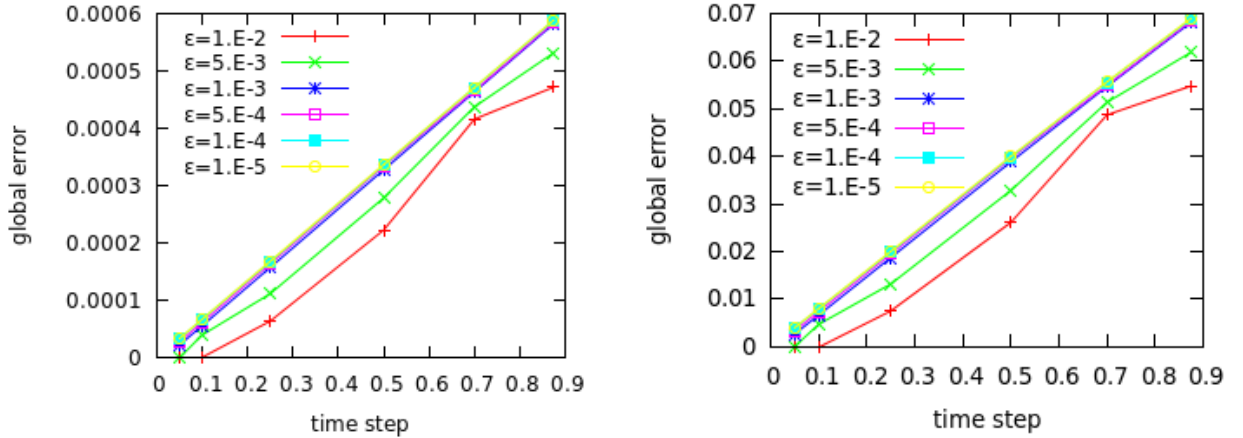

Figure 3: First nonlinear case: the global Euclidean error at time 3.5 for an initial particle on (left) and off (right) the slow manifold

We have used the initial conditions, on and off the slow manifold, mentioned in the previous section, since $(R, V) = (0, 0)$ is still a stationary solution to problem (4.1). We have applied Algorithm 4.1 for each particle. Denoting by $(R(t), V(t))$ the result of the ETD-PIC method and by $(R_{\text{ref}}(t), V_{\text{ref}}(t))$ the reference solution of problem (4.1) for some initial condition, the

10

global errors (the max value in time of the Euclidean norm of $(R(t) - R_{\text{ref}}(t), V(t) - V_{\text{ref}}(t))$) at final time 3.5 are shown in Fig. 3 for several values of ε. The error curves in the left panel are derived with the above initial condition on the slow manifold.
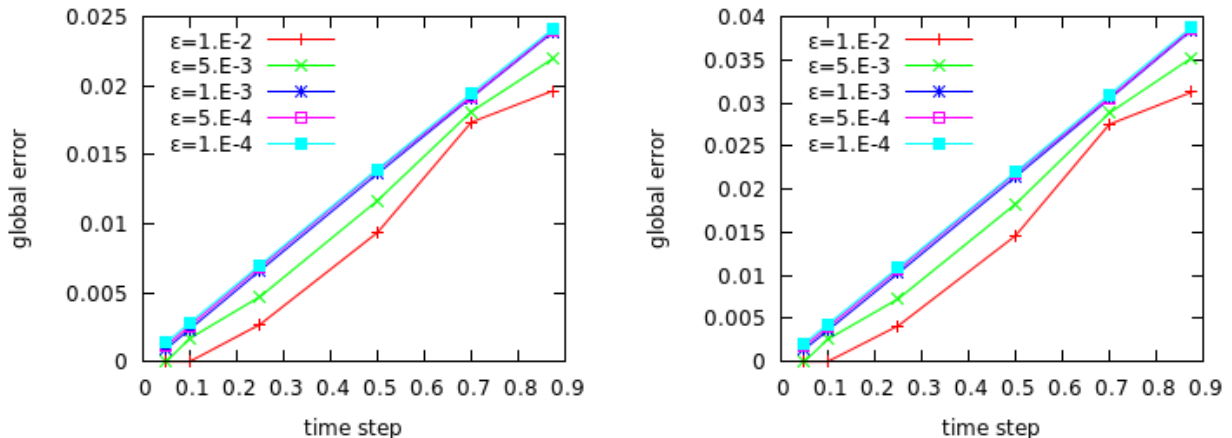

Figure 4: Vlasov-Poisson case: the global Euclidean error at time 3.5 for an initial particle on (left) and off (right) the slow manifold

5.3 The Vlasov-Poisson case

No analytical solution to system (4.1) is available in the case where E_ε is the solution of Poisson equation (2.2)(b). We solve numerically this equation by using a trapezoidal formula for the integral in r with 128 cells for the space interval. As above, the time step of the RK4 solver for computing the reference solution is $dt = \sqrt{\varepsilon}\,\varepsilon$ and we have used the same initial conditions, on and off the slow manifold. The global errors of the ETD-PIC method are shown in Fig. 4 for different values of ε.

Remark 5.1.   *1. From Figs. 3 and 4 one can see the announced property of the scheme, that is the uniform accuracy when ε goes to 0. We also observe that the order of accuracy decreases uniformly when ε goes to 0.*

   2. Note the very large time steps with respect to ε that the ETD scheme allows to use. For instance, when the time step is 0.7 this means that it goes from $70\,\varepsilon$ when $\varepsilon = 10^{-2}$ to $70000\,\varepsilon$ when $\varepsilon = 10^{-5}$.

5.4 Comments

First, we notice that for each nonlinear case above, the errors are larger when the initial condition is off the slow manifold. Then, when $\varepsilon = 10^{-3}$, we can see in Fig. 5 that the difference between errors for off and on the slow manifold particles is more significant in the first nonlinear test case.

Second, still when $\varepsilon = 10^{-3}$, we have compared the two nonlinear test cases. When simulation is done with an initial particle on the slow manifold (ONSM), the error is much more significant
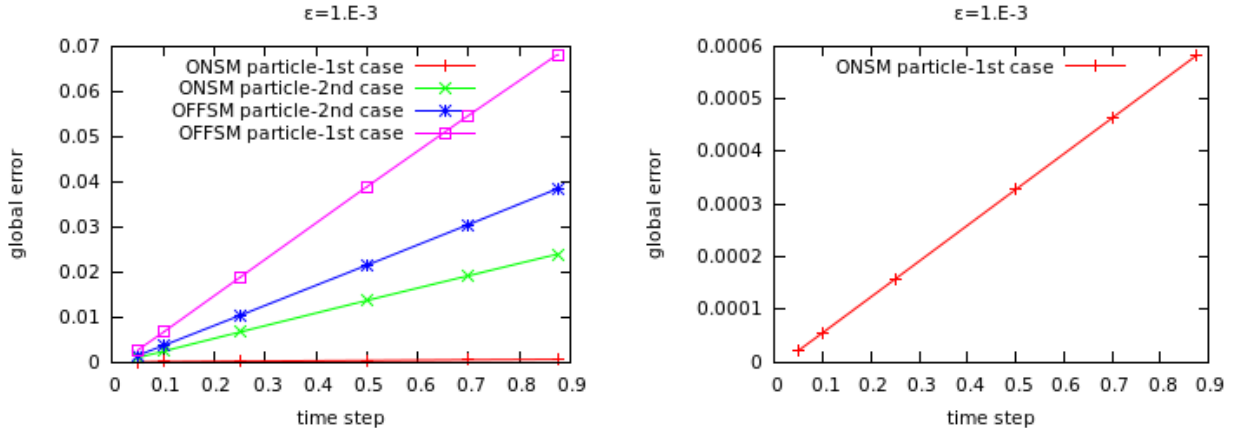
Figure 5: Global errors at time 3.5 for initial particles on and off the slow manifold in both nonlinear cases (1st case is the one in Section 5.2, 2nd case is the Vlasov-Poisson case)

for the second nonlinear case, as expected because of the additional numerical errors due to Poisson solver (see Fig. 5). Surprisingly, for an initial particle off the slow manifold (OFFSM) the error behaves conversely, although they are of the same order of magnitude.

We think the reason for both comments above is in the use of $2\pi\varepsilon$ as the fast period for all the particles, which is more or less accurate. Precisely, we have experimentally found that while the period of an ONSM particle in the first nonlinear case is very close to $2\pi\varepsilon$ (thus justifying the much smaller error at the right of Fig. 5), in the second nonlinear case this period is rather close to $2\pi\varepsilon + 3\varepsilon^2$. In addition, for an OFFSM particle our experiments lead to fast times close to $2\pi\varepsilon - \varepsilon^2$ in the first nonlinear case and to $2\pi\varepsilon + \varepsilon^2$ in the second nonlinear case. Even if these numbers seem to be small to induce such significant errors, we recall that in the linear test case above, $2\pi\varepsilon - t_p$ is very close to $3\varepsilon^2$ and the use of a not accurate fast time for particles leads to an unstable simulation. That is why we test in the following section *modified ETD-PIC schemes*, where we use a more accurate fast time than $2\pi\varepsilon$.

5.5 Validation of the modified ETD-PIC numerical scheme

Finally, we present the numerical results obtained with Algorithm 4.1 where $2\pi\varepsilon$ was replaced with a more accurate period. More precisely, we first compute numerically an approximation of the period for each particle, based on the fact that particles trajectories in time are sinusoid-like almost periodic functions. The trajectories are computed with the same Runge-Kutta solver used for the reference solution's calculation. Thus, the particles are pushed during some very small time until all the particles reach their trajectory's third extremum. The criterion for finding these extrema is the velocity's change of sign. We then stated that each particle's period is the time interval between the first and the third extremum. Eventually, we compute the mean of these periods. It is by this mean that we replace $2\pi\varepsilon$ in the first two steps of Algorithm 4.1. Summarising, the implementation of the modified ETD-PIC scheme starts with the finding of the mean period and then uses this value in Algorithm 4.1 all along the simulation.

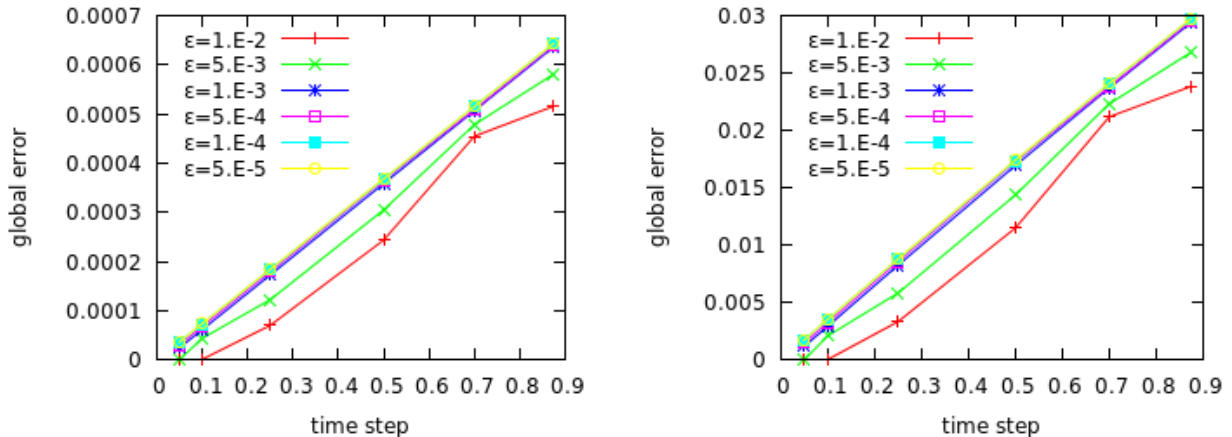Our experiments show that, at best, in the coupling with Poisson case, the error is smaller

12

Figure 6: First nonlinear case, Algorithm 4.1 with the particles mean period: the global Euclidean error at time 3.5 for an initial particle on (left) and off (right) the slow manifold

by a factor of 10 than that computed with period $2\pi\varepsilon$ (see Figs. 4 and 7). For the nonlinear case in Section 5.2, as expected (see Section 5.4), the errors are slightly larger than those computed with period $2\pi\varepsilon$ for an initial ONSM particle, but smaller for an initial OFFSM particle.

At the end, we also illustrate the results of ETD-PIC scheme vs. *modified ETD-PIC scheme* by representing the solution to Vlasov equation with an electric field, first given as in Section 5.2 (Fig. 8) and second as solution of the Poisson equation (Fig. 9). These beams of particles were obtained with the larger time step used for the errors calculus, $dt = 0.875$, meaning only 4 iterations for the ETD-PIC schemes to reach $t = 3.5$.

6 Conclusion

In this paper we have introduced a new numerical scheme for solving some stiff (highly oscillatory) differential equation. This scheme is based on exponential time differencing and can accurately handle large time steps with respect to the fast oscillation of the solution. It is applied in the framework of a Particle-In-Cell method for solving some Vlasov-Poisson equation. Since the numerical results are encouraging, several ways to explore in the future may be usefull to improve some points in the scheme development.

We have seen that the use of $2\pi\varepsilon$ as fast time within the first step of Algorithm 4.1 may lead to an unstable simulation. In addition, even in a stable simulation, the use of the particles mean period gives smaller errors than those obtained with $2\pi\varepsilon$. Therefore, we think it is important to find theoretically a more accurate approximation of the fast time.

It will be interesting to see if our numerical scheme preserves the two-scale asymptotic limit, meaning that $\varepsilon = 0$ in the numerical scheme leads to a consistent discretization of the two-scale limit model. This remark is based on the fact that the ETD discretization we have used is very close to an explicit discretization of the limit model in Theorem 1.1 of [5].

Last but not least, the ETD-PIC schemes proposed in this paper need to be tested on other systems where different types of stiff differential equations are to be solved.
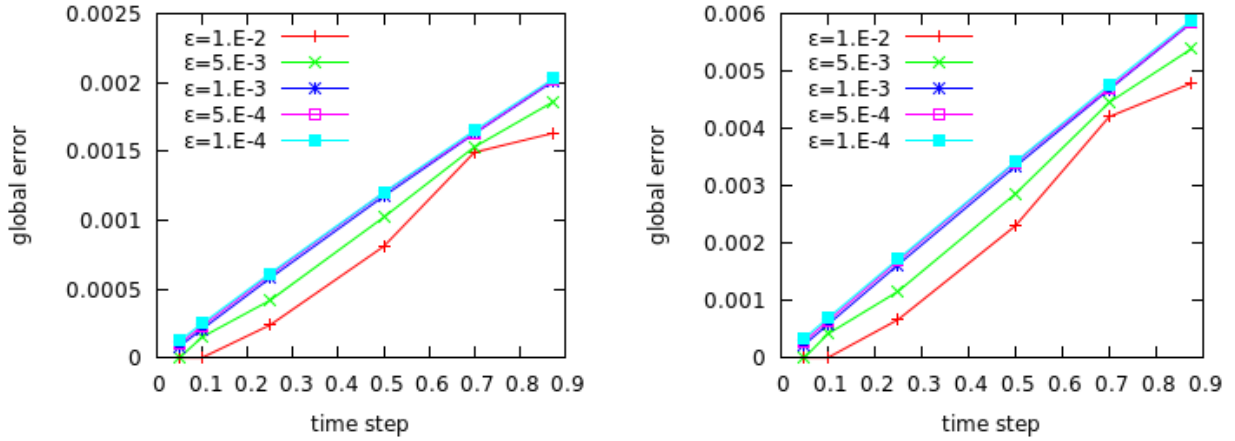
Figure 7: Vlasov-Poisson case, Algorithm 4.1 with the particles mean period: the global Euclidean error at time 3.5 for an initial particle on (left) and off (right) the slow manifold

References

[1] C. K. BIRDSALL, A. B. LANGDON, Plasma physics via computer simulation, Institute of Physics, Bristol (1991).

[2] J. P. BOYD, Chebyshev and Fourier Spectral Methods, Dover, New York (2001).

[3] S. M. COX, P. C. MATTHEWS, *Exponential Time Differencing for Stiff Systems*, J. Comput. Phys. **176** (2002), 430-455.

[4] F. FILBET, 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.

[5] E. FRÉNOD, *Application of the averaging method to the gyrokinetic plasma*, Asympt. Analysis **46** (2006), 1-28.

[6] E. FRÉNOD, F. SALVARANI, E. SONNENDRÜCKER, *Long time simulation of a beam in a periodic focusing channel via a two-scale PIC-method*, Math. Models Methods Appl. Sci. **19**-2 (2009), 175-197.

[7] M. HOCHBRUCK, A. OSTERMANN, *Exponential integrators*, Acta Numer. **19** (2010), 209-286.

[8] A.-K. KASSAM, L. N. TREFETHEN, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput. **26**-4 (2005), 1214-1233.

[9] T. TÜCKMANTEL, A. PUKHOV, J. LILJO, M. HOCHBRUCK, *Three-Dimensional Relativistic Particle-in-Cell Hybrid Code Based on an Exponential Integrator*, IEEE Trans. Plasma Sci. **38**-9 (2010), 2383-2389.
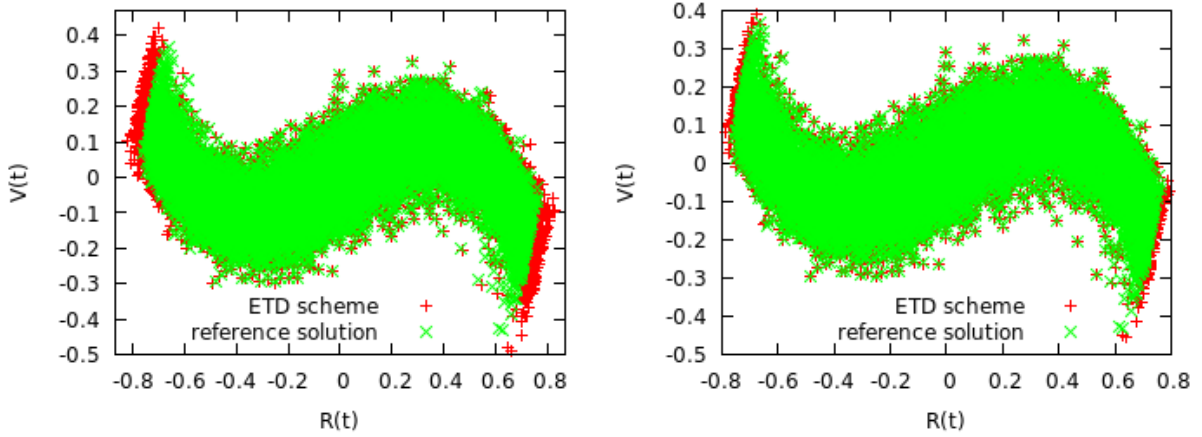
Figure 8: First nonlinear case, $\varepsilon = 10^{-4}$: phase space solution computed with a time step $8750\,\varepsilon$ at final time 3.5, using for particles period $2\pi\varepsilon$ (at left) and the mean period (at right)
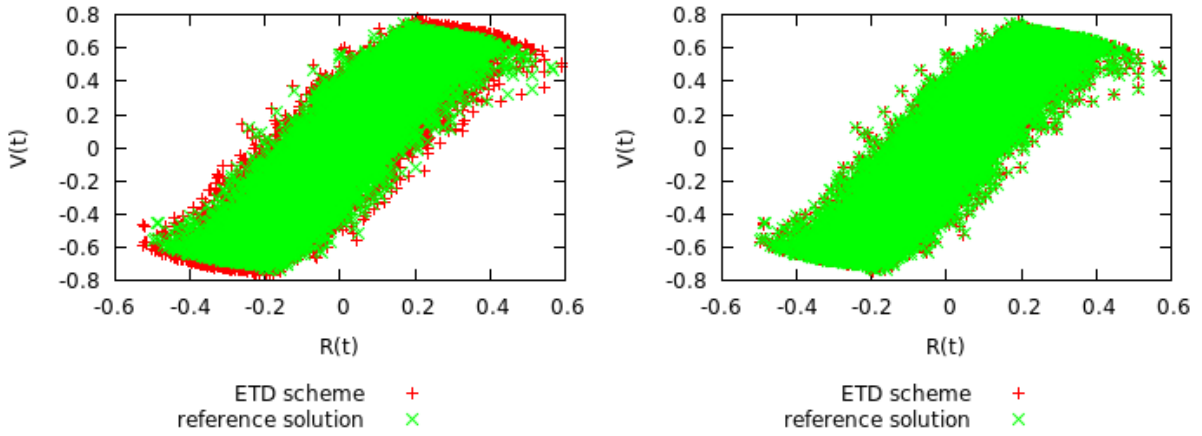

Figure 9: Vlasov-Poisson case, $\varepsilon = 10^{-4}$: phase space solution computed with a time step $8750\,\varepsilon$ at final time 3.5, using for particles period $2\pi\varepsilon$ (at left) and the mean period (at right)