Multiscale Particle-in-Cell methods and comparisons for the long time two-dimensional Vlasov-Poisson equation with strong magnetic field

Nicolas Crouseilles[a], Sever A. Hirstoaga[b], Xiaofei Zhao[a]

[a]*INRIA-Rennes Bretagne Atlantique, IPSO Project and IRMAR, Université de Rennes 1, France*
[b]*INRIA Nancy-Grand Est, TONUS Project and IRMA (UMR CNRS 7501), Université de Strasbourg, France*

Abstract

We applied different kinds of multiscale methods to numerically study the long-time Vlasov-Poisson equation with a strong magnetic field. The multiscale methods include an asymptotic preserving Runge-Kutta scheme, an exponential time discretization scheme, stroboscopic averaging method and a uniformly accurate two-scale formulation. We briefly review these methods and then adapt them to solve the Vlasov-Poisson equation under Particle-in-Cell discretization. Extensive numerical experiments are conducted to investigate and compare the accuracy, efficiency and long-time behaviour of all the methods. The methods with the best performance under different parameter regimes are identified.

Keywords: Vlasov-Poisson equation, Highly oscillatory, Particle-in-Cell, Numerical comparison, Two-scale formulation, Stroboscopic averaging, Asymptotic preserving Runge-Kutta, Exponential time discretization

1. Introduction

In this work, we consider the long time Vlasov-Poisson equation with a strong external homogeneous magnetic field in four phase space dimensions as [19, 18]. The distribution function $f^\varepsilon(t,\mathbf{x},\mathbf{v})$ depending on time $t \geq 0$, on space $\mathbf{x} = (x_1, x_2)^T$ and on velocity $\mathbf{v} = (v_1, v_2)^T$, is then the solution of the following kinetic equation

$$\partial_t f^\varepsilon + \frac{\mathbf{v}}{\varepsilon}\cdot\nabla_{\mathbf{x}} f^\varepsilon + \frac{1}{\varepsilon}\left(\mathbf{E}^\varepsilon + \frac{1}{\varepsilon}\mathbf{v}^\perp\right)\cdot\nabla_{\mathbf{v}} f^\varepsilon = 0, \quad t>0,\ \mathbf{x},\mathbf{v}\in\mathbb{R}^2, \tag{1.1a}$$

$$f^\varepsilon(0,\mathbf{x},\mathbf{v}) = f_0(\mathbf{x},\mathbf{v}), \quad \mathbf{x},\mathbf{v}\in\mathbb{R}^2, \tag{1.1b}$$

where the self-consistent electric field $\mathbf{E}^\varepsilon(t,\mathbf{x}) = -\nabla_{\mathbf{x}}\phi^\varepsilon(t,\mathbf{x})$ (with ϕ^ε the electric potential) is determined by the Poisson equation

$$-\Delta\phi^\varepsilon = \rho^\varepsilon - 1,\ \ \rho^\varepsilon(t,\mathbf{x}) := \int_{\mathbb{R}^2} f^\varepsilon(t,\mathbf{x},\mathbf{v})d\mathbf{v}, \quad t\geq 0,\ \mathbf{x}\in\mathbb{R}^2. \tag{1.2}$$

We denote $\mathbf{v}^\perp$ hereafter as

$$\mathbf{v}^\perp = J\mathbf{v} = (v_2, -v_1)^T, \quad \text{with} \quad J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The real-valued scalar function $f^\varepsilon(t,\mathbf{x},\mathbf{v})$ is the unknown representing the space-velocity distribution of electrons. The given parameter $\varepsilon > 0$ is inversely proportional to the strength of the magnetic field. The function ρ^ε is referred as the density.

Email addresses: `nicolas.crouseilles@inria.fr` (Nicolas Crouseilles), `hirstoaga@math.unistra.fr` (Sever A. Hirstoaga), `zhxfnus@gmail.com` (Xiaofei Zhao)

When $\varepsilon = 1$, it corresponds the classical regime and the equation (1.1) has been widely studied in the literature. When the parameter ε becomes very small, the solution $f^\varepsilon(t, \mathbf{x}, \mathbf{v})$ of the Vlasov-Poisson equation (1.1) is known to exhibit fast oscillations in time with wavelength $O(\varepsilon^2)$. It has been proved that f^ε weak-$*$ converges to $f(t, \mathbf{x}, \mathbf{v})$ (see [19]) which solves

$$\partial_t f + \mathbf{E}^\perp \cdot \nabla_{\mathbf{x}} f + \frac{1}{2}\Delta\phi\ \mathbf{v}^\perp \cdot \nabla_{\mathbf{v}} f = 0, \quad t > 0, \mathbf{x}, \mathbf{v} \in \mathbb{R}^2, \tag{1.3a}$$

$$\mathbf{E} = -\nabla_{\mathbf{x}}\phi, \quad -\Delta\phi(t, \mathbf{x}) = \int_{\mathbb{R}^2} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} - 1, \quad t > 0,\ \mathbf{x} \in \mathbb{R}^2, \tag{1.3b}$$

$$f(0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v}). \tag{1.3c}$$

By integrating (1.3a) with respect to $\mathbf{v} \in \mathbb{R}^2$, we further get the well-known Guiding Center equation satisfied by ρ, the limit of the density ρ^ε as $\varepsilon \to 0$

$$\partial_t \rho + \mathbf{E}^\perp \cdot \nabla_{\mathbf{x}} \rho = 0, \quad t > 0, \mathbf{x} \in \mathbb{R}^2, \tag{1.4a}$$

$$\mathbf{E} = -\nabla_{\mathbf{x}}\phi, \quad -\Delta\phi(t, \mathbf{x}) = \rho(t, \mathbf{x}) - 1, \quad t > 0,\ \mathbf{x} \in \mathbb{R}^2, \tag{1.4b}$$

$$\rho(0, \mathbf{x}) = \int_{\mathbb{R}^2} f(0, \mathbf{x}, \mathbf{v}) d\mathbf{v}. \tag{1.4c}$$

Due to high oscillations in time, if one wants to do accurate simulation of the problem (1.1) using classical numerical solvers, one needs to consider small time steps (typically smaller than ε^2) which leads to prohibitive time computations in the asymptotic regime. A way to avoid this restriction is to use asymptotic non stiff models like (1.3) or (1.4); such model can be simulated using ε-independent time steps since the models do not contain any stiffness. Obviously, these models are only valid in the asymptotic regime $\varepsilon << 1$ and can not correctly describe the dynamic of (1.1) for intermediate regime or when $\varepsilon \approx 1$. As a consequence, multiscale approaches turns out to be very competitive since they are able to handle small or large values for ε without adapting the numerical parameters. Then, the CPU time is independent of ε. Our goal in this work is to compare, both in terms of accuracy and of CPU time, different multiscale methods recently introduced in the literature for numerically solving highly oscillatory problems like (1.1).

Many multiscale techniques have been devoted to design numerical integrators dedicated to the solving of highly oscillatory problems under different context. For kinetic models in high dimensions, the multiscale schemes are usually performed on the characteristics under the Particle-in-Cell discretization [24, 20]. To tackle the Vlasov-Poisson equation (1.1), a class of semi-implicit Runge-Kutta (SIRK) methods has been proposed in [15]. The SIRK methods could have high order of accuracy and superior stability, and they have been shown in [15] as asymptotic preserving schemes for (1.1) which solves consistently as $\varepsilon \to 0$ the Guiding Center limit model (1.4). Very recently, a two-scale formulation (TSF) integrator has been proposed in [11] for solving (1.1). The TSF approach separates the fast variable out in the equation and solves it as an extra degree-of-freedom, which benefits to reduce the oscillation. The TSF integrator has been illustrated in [11] to offer uniform second order accuracy in solving (1.1) for all $\varepsilon \in (0, 1]$, while extra degree-of-freedom potentially increases the computational cost. An exponential time discretization (ETD) scheme has been proposed in [18] for solving the Vlasov-Poisson equation under a short time scaling, which means equivalently to consider the model (1.1) for $0 < t \leq \varepsilon$. The results in [18] show that the ETD scheme is very accurate as $0 < \varepsilon \ll 1$. The good performance of ETD for the short time problem gives motivations to consider it for the long-time Vlasov-Poisson equation (1.1). On the other hand, under the hierarchy of the Heterogeneous Multiscale Methods [1, 14], the stroboscopic averaging method (SAM) has been presented to solve highly oscillatory equations under abstract framework in [3]. It has been recently implemented for integrating a highly oscillatory Schrödinger equation in [4, 7], and the success there of SAM in the dispersive equation motivates their prospectiveness in solving highly oscillatory kinetic models.

In this work, we are going to perform the above mentioned methods: SAM, ETD, SIRK and TSF to the Vlasov-Poisson equation (1.1) under the framework of Particle-in-Cell method, and carry out systematical comparisons between them. For SIRK and TSF, we shall briefly present and review them. For ETD and SAM, we shall adapt and derive them in details based on different formulations of the characteristics. We shall test the performance of all the methods for a wide range of ε, and investigate their accuracy and efficiency with comparison carried out aiming to identify the best one among all in different regimes of the problem.

The rest of the paper is organized as follows. First, we recall the framework of PIC method and derive different formulations of characteristics in Section 2 as preparations for presenting the schemes. The mentioned different kinds of multiscale schemes are given/reviewed in Section 3. Section 4 presents the results of numerical tests and comparisons. Conclusions are drawn in Section 5.

2. Particle-in-Cell and framework

In this section, we are going to briefly review the Particle-in-Cell (PIC) discretization for the Vlasov-Poisson equation (1.1). We shall first present the PIC framework and then focus on the integrations of the characteristics. We shall derive three different formulations of the characteristics in order to perform different kinds of numerical integrators and obtain corresponding PIC methods.

2.1. PIC framework

The PIC discretization consists in approximating the unknown distribution $f^\varepsilon(t, \mathbf{x}, \mathbf{v})$ of (1.1) by a sum of Dirac masses centred at $(\mathbf{x}_k(t), \mathbf{v}_k(t))$ with weight $\omega_k > 0$ for $k = 1, \ldots, N_p$ and $N_p \in \mathbb{N}$ as

$$f_p^\varepsilon(t, \mathbf{x}, \mathbf{v}) = \sum_{k=1}^{N_p} \omega_k \delta(\mathbf{x} - \mathbf{x}_k(t))\delta(\mathbf{v} - \mathbf{v}_k(t)), \quad t \geq 0, \ \mathbf{x}, \mathbf{v} \in \mathbb{R}^2. \tag{2.1}$$

Plugging (2.1) into (1.1), in the sense of distribution, for $k = 1, \ldots, N_p$, one gets that each particle k obeys the characteristics equation

$$\dot{\mathbf{x}}_k(t) = \frac{\mathbf{v}_k(t)}{\varepsilon}, \tag{2.2a}$$

$$\dot{\mathbf{v}}_k(t) = \frac{\mathbf{E}^\varepsilon(t, \mathbf{x}_k(t))}{\varepsilon} + \frac{\mathbf{v}_k^\perp(t)}{\varepsilon^2}, \quad t > 0, \tag{2.2b}$$

$$\mathbf{x}_k(0) = \mathbf{x}_{k,0}, \quad \mathbf{v}_k(0) = \mathbf{v}_{k,0}. \tag{2.2c}$$

The weight w_k and initial values of the particles $\mathbf{x}_{k,0}, \mathbf{v}_{k,0}$ for $k = 0, \ldots, N_p$ are prescribed according to the given initial distribution $f_0(\mathbf{x}, \mathbf{v})$ in (1.1b). To determine the weight, by integrating (2.1) at $t = 0$ in whole space we require

$$\sum_{k=1}^{N_p} \omega_k = \int_{\mathbb{R}^2 \times \mathbb{R}^2} f_0(\mathbf{x}, \mathbf{v}) d\mathbf{x} d\mathbf{v}.$$

Thus, a simple choice of uniform weight for all particles would be

$$w_k = \frac{1}{N_p} \int_{\mathbb{R}^2 \times \mathbb{R}^2} f_0(\mathbf{x}, \mathbf{v}) d\mathbf{x} d\mathbf{v}, \quad k = 1, \ldots, N_p.$$

The initializing $\mathbf{x}_{k,0}, \mathbf{v}_{k,0}$ for $k = 0, \ldots, N_p$ is a rather well-established and classical sampling issue. It could be done by the Monte Carlo type rejection sampling method as a general approach. The detailed process can be found in standard statistics textbooks. As a consequence, this approach introduces a noise of order $1/\sqrt{N_p}$ (see [24]). When f_0 is of variable separation form, i.e. $f_0(\mathbf{x}, \mathbf{v}) = \chi_1(\mathbf{x})\chi_2(\mathbf{v})$, especially for the cold plasma case as widely considered in the literatures where χ_1 is an one-dimensional distribution and χ_2 is a Gaussian, the initial positions of particles could be given by the so-called inversion of cumulative distribution function. This approach is a deterministic way and a detailed description can be found in [2]. We omit the details here for brevity. The characteristics equations for particles (2.2) are coupled to the

Poisson equation (1.2) through the electric field $\mathbf{E}^\varepsilon$. Once the positions $\{\mathbf{x}_k(t)\}_{k=1,\ldots,N_p}$ of the particles are obtained at time $t>0$, one needs to evaluate the approximated density

$$\rho_p^\varepsilon(t,\mathbf{x}) = \sum_{k=1}^{N_p} \omega_k \delta(\mathbf{x}-\mathbf{x}_k(t)) \approx \rho^\varepsilon(t,\mathbf{x}) \quad \mathbf{x}\in\mathbb{R}^2, \tag{2.3}$$

then solve the Poisson equation

$$-\Delta\phi_p^\varepsilon(t,\mathbf{x}) = \rho_p^\varepsilon(t,\mathbf{x}) - 1, \quad \mathbf{E}_p^\varepsilon(t,\mathbf{x}) = -\nabla\phi_p^\varepsilon(t,\mathbf{x}), \quad \mathbf{x}\in\mathbb{R}^2, \tag{2.4}$$

on a mesh grid of $\mathbf{x}$ in $\mathbb{R}^2$ for $\mathbf{E}_p^\varepsilon(t,\mathbf{x}) \approx \mathbf{E}^\varepsilon(t,\mathbf{x})$, and finally interpolate for values of $\mathbf{E}^\varepsilon(t,\mathbf{x}_k(t))$ at each particle position. In practice, the Dirac delta function $\delta(\mathbf{x})$ is approximated by some regularized basis $S(\mathbf{x})$. For example in 1D, it is approximated by the B-spline function $S^m(x)$ of order $m\in\mathbb{N}$ (see [24]):

$$S^0(x) := \begin{cases} \dfrac{1}{\Delta x}, & |x| \le \dfrac{\Delta x}{2}, \\ 0, & \text{else}, \end{cases}, \qquad S^m(x) := \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} S^{m-1}(y)dy, \quad m \ge 1. \tag{2.5}$$

The case in two dimensions is done by tensor product. The B-spline function S^m is defined locally in space which is preferred from numerical point of view, but globally it is only a C^{m-1} function for $m\ge 1$. A smooth but global basis has been considered in [20] to get high order accuracy. The classical PIC scheme is hence completed by a standard numerical integrator towards (2.2).

2.2. Reformulations of the characteristics equations

In order to design some of the multiscale PIC methods, one needs to reformulate the characteristics equations (2.2). For instance, one could filter out the main oscillations in (2.2) as

$$\mathbf{y}_k(t) = \mathrm{e}^{-tJ/\varepsilon^2}\mathbf{v}_k(t), \quad t\ge 0, \quad \text{with} \quad \mathrm{e}^{sJ} = \begin{pmatrix} \cos(s) & \sin(s) \\ -\sin(s) & \cos(s) \end{pmatrix}, \tag{2.6}$$

then one gets a first formulation of the characteristic equations (2.2) well-adapted for the ETD approach

$$\dot{\mathbf{x}}_k(t) = \frac{1}{\varepsilon}\mathrm{e}^{tJ/\varepsilon^2}\mathbf{y}_k(t), \tag{2.7a}$$

$$\dot{\mathbf{y}}_k(t) = \frac{1}{\varepsilon}\mathrm{e}^{-tJ/\varepsilon^2}\mathbf{E}^\varepsilon(t,\mathbf{x}_k(t)), \quad t>0, \tag{2.7b}$$

$$\mathbf{x}_k(0) = \mathbf{x}_{k,0}, \quad \mathbf{y}_k(0) = \mathbf{v}_{k,0}. \tag{2.7c}$$

In order to later apply the SAM and TSF methods, we need to reformulate the latter equations into a suitable form. To do so, we first introduce the following new unknown

$$\mathbf{u}_{k,+}(t) := \mathbf{x}_k(t) + \varepsilon J\mathrm{e}^{tJ/\varepsilon^2}\mathbf{y}_k(t), \quad \mathbf{u}_{k,-}(t) := -\varepsilon J\mathbf{y}_k(t), \qquad t\ge 0, \tag{2.8}$$

from which we recover the original ones thanks to

$$\mathbf{x}_k(t) = \mathbf{u}_{k,+}(t) + \mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_{k,-}(t), \quad \text{and} \quad \mathbf{v}_k(t) = \mathrm{e}^{tJ/\varepsilon^2}\mathbf{y}_k(t) = \frac{\mathrm{e}^{tJ/\varepsilon^2}}{\varepsilon}J\mathbf{u}_{k,-}(t).$$

Then from (2.7) we get

$$\begin{cases} \dot{\mathbf{u}}_{k,+}(t) = J\mathbf{E}^\varepsilon\left(t, \mathbf{u}_{k,+}(t) + \mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_{k,-}(t)\right), \\ \dot{\mathbf{u}}_{k,-}(t) = -J\mathrm{e}^{-tJ/\varepsilon^2}\mathbf{E}^\varepsilon\left(t, \mathbf{u}_{k,+}(t) + \mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_{k,-}(t)\right), \quad t>0. \end{cases} \tag{2.9}$$

Keeping in mind that with PIC method, the electric field $\mathbf{E}^\varepsilon$ is given by (2.3) and (2.4), and for simplification we denote

$$-\nabla_{\mathbf{x}} \cdot \mathbf{E}^\varepsilon(t, \mathbf{x}) = \sum_{k=1}^{N_p} \omega_k \delta\left(\mathbf{x} - \left(\mathbf{u}_{k,+}(t) + \mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_{k,-}(t)\right)\right) - 1, \tag{2.10}$$

where here $\mathbf{E}^\varepsilon$ is always interpreted as a conservative field generated by a potential. If we denote

$$\mathbf{u}_\pm(t) := \left(\mathbf{u}_{1,\pm}(t), \ldots, \mathbf{u}_{N_p,\pm}(t)\right)^T,$$

as a column vector, then we could interpret the t variable in $\mathbf{E}^\varepsilon(t, \mathbf{x})$ as defined by $\mathbf{u}_\pm(t)$ and the fast time variable t/ε^2 through the nonlinear relation (2.10), i.e. there is a function $\mathbf{F}^\varepsilon$ that reads as

$$\mathbf{F}^\varepsilon(t/\varepsilon^2, \mathbf{u}_+(t), \mathbf{u}_-(t), \mathbf{x}) := \mathbf{E}^\varepsilon\left(t, \mathbf{x}\right).$$

Hence, (2.9) can be seen as a system for all k and then can be written as

$$\dot{\mathbf{u}}_\pm(t) = \mathbf{F}_\pm\left(t/\varepsilon^2, \mathbf{u}_+(t), \mathbf{u}_-(t)\right), \tag{2.11}$$

where $\mathbf{F}_\pm := \left(\mathbf{F}_{1,\pm}, \ldots, \mathbf{F}_{N_p,\pm}\right)^T$ and

$$\begin{aligned} \mathbf{F}_{k,+}\left(s, \mathbf{u}_+, \mathbf{u}_-\right) &= J\mathbf{F}^\varepsilon\left(s, \mathbf{u}_+, \mathbf{u}_-, \mathbf{u}_{k,+} + \mathrm{e}^{sJ}\mathbf{u}_{k,-}\right), \\ \mathbf{F}_{k,-}\left(s, \mathbf{u}_+, \mathbf{u}_-\right) &= -J\mathrm{e}^{-sJ}\mathbf{F}^\varepsilon\left(s, \mathbf{u}_+, \mathbf{u}_-, \mathbf{u}_{k,+} + \mathrm{e}^{sJ}\mathbf{u}_{k,-}\right). \end{aligned} \tag{2.12}$$

Denoting the initial data in (2.7c) as column vectors $\mathbf{x}_0 = (\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N_p,0})^T$ and $\mathbf{y}_0 = (\mathbf{v}_{1,0}, \ldots, \mathbf{v}_{N_p,0})^T$, we summarise the reformulated characteristics as

$$\begin{cases} \dot{\mathbf{u}}_\pm(t) = \mathbf{F}_\pm\left(t/\varepsilon^2, \mathbf{u}_+(t), \mathbf{u}_-(t)\right), \qquad 0 < t \leq T, \\ \mathbf{u}_+(0) = \mathbf{x}_0 + \varepsilon J\mathbf{v}_0, \quad \mathbf{u}_-(0) = -\varepsilon J\mathbf{v}_0. \end{cases} \tag{2.13}$$

This ODE is well-posed on $[0, T]$, where the final time $T > 0$ of the above initial value problem is independent of ε and the function $\mathbf{F}_\pm(s, \mathbf{u}_+, \mathbf{u}_-)$ is clearly 2π-periodic in s. Hence, (2.13) meets the requirement to perform the multiscale methods like SAM or TSF method. After solving (2.13), one uses

$$\mathbf{x}(t) = \mathbf{u}_+(t) + \mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_-(t), \qquad \mathbf{v}(t) = \frac{J}{\varepsilon}\mathrm{e}^{tJ/\varepsilon^2}\mathbf{u}_-(t),$$

to recover the original particle variables.

In the following section, we are going to present PIC schemes by performing different numerical integrators on either (2.2) for SIRK, (2.7) for ETD or (2.13) for SAM and TSF. For simplicity of notation and convenience of comparison, we shall fix all the involved local solvers in the numerical integrators as second order approximations to present the schemes.

3. Numerical methods

In this section, we present and review the numerical methods we intend to compare in the next section.

3.1. Semi-implicit Runge-Kutta

A class of semi-implicit Runge-Kutta (SIRK) schemes have been proposed in [15] for solving the characteristics. These schemes are derived based on (2.2). Among the (second order) SIRK schemes considered in [15], there is a second order L-stable scheme which has been shown to outperform the others in accuracy and stability.

Denoting $\gamma = 1 - 1/\sqrt{2}$, the second order L-stable SIRK method for the approximation of (2.2) reads

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{(1-\gamma)\Delta t}{\varepsilon} F^{(1)} + \frac{\gamma \Delta t}{\varepsilon} F^{(2)}, \quad \mathbf{x}^{n+1} = \mathbf{x}^n + \frac{(1-\gamma)\Delta t}{\varepsilon} \mathbf{v}^{(1)} + \frac{\gamma \Delta t}{\varepsilon} \mathbf{v}^{n+1}, \tag{3.1}$$

where

$$t^{(1)} = t_n + \frac{\Delta t}{2\gamma}, \quad \mathbf{x}^{(1)} = \mathbf{x}^n + \frac{\Delta t}{2\gamma\varepsilon} \mathbf{v}^{(1)}, \quad \mathbf{v}^{(1)} = \mathbf{v}^n + \frac{\gamma \Delta t}{\varepsilon} F^{(1)},$$
$$F^{(1)} = \frac{J\mathbf{v}^{(1)}}{\varepsilon} + \mathbf{E}^\varepsilon(t_n, \mathbf{x}^n), \quad F^{(2)} = \frac{J\mathbf{v}^{n+1}}{\varepsilon} + \mathbf{E}^\varepsilon(t^{(1)}, \mathbf{x}^{(1)}).$$

The above SIRK is uniformly linearly stable in integrating (2.2) and the higher order versions are also derived in [15]. The SIRK (3.1) has been shown in [15] to solve consistently the Guiding Center equation (1.4) in the limit regime, i.e. for any fixed $\Delta t > 0$ and $n \geq 0$, $\mathbf{x}^n \to \widetilde{\mathbf{x}}^n$ as $\varepsilon \to 0$, where $\widetilde{\mathbf{x}}^n$ is the numerical solution of the characteristics of (1.4) as

$$\widetilde{\mathbf{x}}^{n+1} = \widetilde{\mathbf{x}}^n + (1-\gamma)\Delta t J\mathbf{E}(t_n, \widetilde{\mathbf{x}}^n) + \gamma \Delta t J\mathbf{E}\left(t^{(1)}, \widetilde{\mathbf{x}}^n + \frac{\Delta t}{2\gamma} J\mathbf{E}(t_n, \widetilde{\mathbf{x}}^n)\right).$$

In this sense, SIRK is asymptotic preserving in the approximations of the particle positions $\mathbf{x}(t)$. Thus, by analyzing the SIRK on the limit model (2.2), an error bound can be

$$|\mathbf{x}^n - \mathbf{x}(t_n)| \leq C(\Delta t^2 + \varepsilon^q), \qquad 0 \leq n \leq \frac{T}{\Delta t}, \tag{3.2}$$

where q appears to be 1 from our the numerical investigations of the convergence of the Vlasov-Poisson equation (1.1) as $\varepsilon \to 0$.

3.2. Exponential time discretization

An exponential time discretization (ETD) PIC method is proposed in [18] for the Vlasov-Poission equation with strong magnetic field under short time scaling. Here we extend the ETD scheme to the long-time scaling problem (1.1) by using the characteristics formulation (2.7).

By applying the Duhamel's principle between t_n and t_{n+1} to the filtered characteristics formulation (2.7), we get

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \int_{t_n}^{t_{n+1}} \frac{1}{\varepsilon} \mathrm{e}^{tJ/\varepsilon^2} \mathbf{y}(t) dt, \quad \mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \frac{1}{\varepsilon} \mathrm{e}^{-tJ/\varepsilon^2} \mathbf{E}^\varepsilon(t, \mathbf{x}(t)) dt \quad n \geq 0. \tag{3.3}$$

For a chosen (macro) time step $\Delta t = H > 0$, we denote

$$H = N \cdot (2\pi\varepsilon^2) + r\varepsilon^2, \quad N \in \mathbb{N}, \ 0 \leq r < 2\pi \quad \text{and} \quad t_n^* = t_n + N \cdot (2\pi\varepsilon^2). \tag{3.4}$$

When $H \geq 2\pi\varepsilon^2$, i.e. $N > 0$, we write the integral equations (3.3) as

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n^*) + J_{\mathbf{x}}^n, \qquad \mathbf{y}(t_{n+1}) = \mathbf{y}(t_n^*) + J_{\mathbf{y}}^n, \quad n \geq 0,$$

where

$$J_{\mathbf{x}}^n := \int_{t_n^*}^{t_{n+1}} \frac{1}{\varepsilon} \mathrm{e}^{tJ/\varepsilon^2} \mathbf{y}(t) dt, \qquad J_{\mathbf{y}}^n := \int_{t_n^*}^{t_{n+1}} \frac{1}{\varepsilon} \mathrm{e}^{-tJ/\varepsilon^2} \mathbf{E}^\varepsilon(t, \mathbf{x}(t)) dt,$$

and we approximate $\mathbf{x}(t_n^*)$ and $\mathbf{y}(t_n^*)$ as

$$\mathbf{x}(t_n^*) \approx \mathbf{x}(t_n) + N I_{\mathbf{x}}^n, \qquad \mathbf{y}(t_n^*) \approx \mathbf{y}(t_n) + N I_{\mathbf{y}}^n, \quad n \geq 0, \tag{3.5}$$

where

$$I_{\mathbf{x}}^n := \int_{t_n}^{t_n+2\pi\varepsilon^2} \frac{1}{\varepsilon}\mathrm{e}^{tJ/\varepsilon^2}\mathbf{y}(t)dt, \qquad I_{\mathbf{y}}^n := \int_{t_n}^{t_n+2\pi\varepsilon^2} \frac{1}{\varepsilon}\mathrm{e}^{-tJ/\varepsilon^2}\mathbf{E}^\varepsilon(t,\mathbf{x}(t))dt.$$

It is clear from (2.7) and by fundamental theorem of calculus,

$$\begin{aligned} I_{\mathbf{x}}^n &= \mathbf{x}(t_n+2\pi\varepsilon^2)-\mathbf{x}(t_n), & I_{\mathbf{y}}^n &= \mathbf{y}(t_n+2\pi\varepsilon^2)-\mathbf{y}(t_n),\\ J_{\mathbf{x}}^n &= \mathbf{x}(t_{n+1})-\mathbf{x}(t_n^*), & J_{\mathbf{y}}^n &= \mathbf{y}(t_{n+1})-\mathbf{y}(t_n^*). \end{aligned}$$

To evaluate $\mathbf{x}(t_n+2\pi\varepsilon^2)$ and $\mathbf{y}(t_n+2\pi\varepsilon^2)$, we solve (2.7) from t_n to $t_n+2\pi\varepsilon^2$, i.e.

$$\dot{\mathbf{x}}(t) = \frac{1}{\varepsilon}\mathrm{e}^{tJ/\varepsilon^2}\mathbf{y}(t), \quad \dot{\mathbf{y}}(t) = \frac{1}{\varepsilon}\mathrm{e}^{-tJ/\varepsilon^2}\mathbf{E}^\varepsilon(t,\mathbf{x}(t)), \quad t_n \le t \le t_n+2\pi\varepsilon^2.$$

By defining $\tilde{\mathbf{x}}^n(s) := \mathbf{x}(t_n+\varepsilon^2 s)$ and $\tilde{\mathbf{y}}^n(s) := \mathbf{y}(t_n+\varepsilon^2 s)$, we can rescale the previous differential system to get the following one

$$\begin{cases} \dot{\tilde{\mathbf{x}}}^n(s) = \varepsilon\mathrm{e}^{t_nJ/\varepsilon^2}\mathrm{e}^{sJ}\tilde{\mathbf{y}}^n(s),\\ \dot{\tilde{\mathbf{y}}}^n(s) = \varepsilon\mathrm{e}^{-t_nJ/\varepsilon^2}\mathrm{e}^{-sJ}\mathbf{E}^\varepsilon(t_n+s\varepsilon^2,\tilde{\mathbf{x}}^n(s)), \quad 0 < s \le 2\pi,\\ \tilde{\mathbf{x}}^n(0) = \mathbf{x}(t_n), \quad \tilde{\mathbf{y}}^n(0) = \mathbf{y}(t_n), \end{cases} \tag{3.6}$$

which is clearly smooth and can be discretized by a classical integrator. Thus, (3.5) can be rewritten as

$$\mathbf{x}(t_n^*) \approx \mathbf{x}(t_n) + N(\tilde{\mathbf{x}}^n(2\pi)-\mathbf{x}(t_n)), \quad \mathbf{y}(t_n^*) \approx \mathbf{y}(t_n) + N(\tilde{\mathbf{y}}^n(2\pi)-\mathbf{y}(t_n)), \quad n \ge 0. \tag{3.7}$$

At last, we push (2.7) from t_n^* to t_{n+1} to get $\mathbf{x}(t_{n+1})$ and $\mathbf{y}(t_{n+1})$. By similarly defining the rescaled variables $\tilde{\mathbf{x}}^*(s) := \mathbf{x}(t_n^*+\varepsilon^2 s)$ and $\tilde{\mathbf{y}}^*(s) := \mathbf{y}(t_n^*+\varepsilon^2 s)$, that is say the last step is to solve

$$\begin{cases} \dot{\tilde{\mathbf{x}}}^*(s) = \varepsilon\mathrm{e}^{t_n^*J/\varepsilon^2}\mathrm{e}^{sJ}\tilde{\mathbf{y}}^*(s),\\ \dot{\tilde{\mathbf{y}}}^*(s) = \varepsilon\mathrm{e}^{-t_n^*J/\varepsilon^2}\mathrm{e}^{-sJ}\mathbf{E}^\varepsilon(t_n^*+s\varepsilon^2,\tilde{\mathbf{x}}^*(s)), \quad 0 < s \le r,\\ \tilde{\mathbf{x}}^*(0) = \mathbf{x}(t_n^*), \quad \tilde{\mathbf{y}}^*(0) = \mathbf{y}(t_n^*), \end{cases} \tag{3.8}$$

which is a smooth problem as well and easy to discretise. Hence, at each time level t_n, the main ETD scheme is given by (3.7) associated with a local initial-value problem (3.6), and accomplished by another local initial-value problem (3.8) which is aiming to offer

$$\mathbf{x}(t_{n+1}) = \tilde{\mathbf{x}}^*(r), \qquad \mathbf{y}(t_{n+1}) = \tilde{\mathbf{y}}^*(r).$$

For detailed discretization of the ETD scheme, when $N > 0$ in (3.4), we choose an integer $n_0 > 0$ to discretise (3.6) and (3.8) in time and denote the (micro) time step $h = \frac{2\pi}{n_0}$, $h_r = \frac{r}{n_0}$, then the scheme goes as

$$\mathbf{x}^* = \mathbf{x}^n + N\left(\left(S_{\mathbf{x}}^h\right)^{n_0}(\mathbf{x}^n,\mathbf{y}^n) - \mathbf{x}^n\right), \quad \mathbf{x}^{n+1} = \left(S_{\mathbf{x}}^{h_r}\right)^{n_0}(\mathbf{x}^*,\mathbf{y}^*), \tag{3.9a}$$

$$\mathbf{y}^* = \mathbf{y}^n + N\left(\left(S_{\mathbf{y}}^h\right)^{n_0}(\mathbf{x}^n,\mathbf{y}^n) - \mathbf{y}^n\right), \quad \mathbf{y}^{n+1} = \left(S_{\mathbf{y}}^{h_r}\right)^{n_0}(\mathbf{x}^*,\mathbf{y}^*), \quad n \ge 0, \tag{3.9b}$$

where $(S_{\mathbf{x}}^h)^{n_0}(\mathbf{x}^n,\mathbf{y}^n) \approx \tilde{\mathbf{x}}^n(2\pi)$ and $(S_{\mathbf{y}}^h)^{n_0}(\mathbf{x}^n,\mathbf{y}^n) \approx \tilde{\mathbf{y}}^n(2\pi)$ are the numerical solution of (3.6) with initial data $(\mathbf{x}^n,\mathbf{y}^n)$ by a numerical integrator under time step h. $(S_{\mathbf{x}}^{h_r})^{n_r}(\mathbf{x}^*,\mathbf{y}^*)$ and $(S_{\mathbf{y}}^{h_r})^{n_r}(\mathbf{x}^*,\mathbf{y}^*)$ are defined similarly for (3.8) with initial data $(\mathbf{x}^*,\mathbf{y}^*)$ discretised by time step h_r. Here, we choose the numerical integrator $S_{\mathbf{x}}^h, S_{\mathbf{y}}^h$ as the central difference integrator for practical implementation, i.e. solving $\dot{u} = \lambda u$ with

$$u^1 = u^0 + h\lambda u^0, \quad u^{n+1} = u^{n-1} + h\lambda u^n, \quad n \ge 1.$$

When $N = 0$ in (3.4), that is the case if $H < 2\pi\varepsilon^2$, we reduce the ETD scheme to a classical central-difference method towards (2.7) with only the macro time step H. In this convention, the ETD scheme is well-defined for all H and ε.

The central difference integrator is stable on the $[-1, 1]$ interval of the imaginary axis. We remark the reason why we choose this 'mean' local integrator rather other more robust integrators such as A-stable schemes, is because we want to investigate the robustness of the whole approach.

The ETD scheme is fully explicit. For solving (2.7), i.e. a single particle, the computational cost of ETD at each (macro) time level is of order $O(n_0)$ where n_0 is the number of time grids in micro step, and the memory cost is $O(1)$. The optimal error bound or rigorous error estimates of the ETD scheme is still not clear to us. Nevertheless, the numerical results indicate it is second order accurate in the classical regime and is first order accurate in the limit regime. Sever, can we derive an estimate ?

3.3. Stroboscopic averaging method

The stroboscopic averaging method (SAM) is introduced in [3] under general framework. By a rescaling $t = \varepsilon^2 s$ and $\mathbf{w}_\pm(s) = \mathbf{u}_\pm(t)$, the model (2.13) becomes

$$\dot{\mathbf{w}}_\pm(s) = \varepsilon^2 \mathbf{F}_\pm(s, \mathbf{w}_+(s), \mathbf{w}_-(s)), \qquad 0 < s \leq \frac{T}{\varepsilon^2}. \tag{3.10}$$

As shown in [3, 4], through a smooth change of variable $\Phi_\pm(s, \Psi_+, \Psi_-)$ which is 2π-periodic in s, there exists a flow $\Psi_\pm$ and an averaged field $G_\pm$ satisfying

$$\begin{cases} \dot{\Psi}_\pm(s) = \varepsilon^2 G_\pm(\Psi_+(s), \Psi_-(s)), & 0 < s \leq \dfrac{T}{\varepsilon^2}, \\ \Psi_\pm(0) = \mathbf{w}_\pm(0), \end{cases} \tag{3.11}$$

such that for some constant $C > 0$ independent of ε nor s

$$|\mathbf{w}_\pm(s) - \Phi_\pm(s, \Psi_+(s), \Psi_-(s))| \leq C\mathrm{e}^{-C/\varepsilon^2}, \qquad 0 \leq s \leq \frac{T}{\varepsilon^2}.$$

For time at the multiples of period, as known as the stroboscopic times, we have the stroboscopic property,

$$\Phi_\pm(s, \Psi_+, \Psi_-) = \Psi_\pm(s), \qquad s/2\pi \in \mathbb{N}.$$

Hence solving the averaged system (3.11) becomes a good choice to approximate (3.10) at the stroboscopic time. As derived in [4, 7], the unknown function $G_\pm$ is approximated by

$$G_\pm(\Psi_+, \Psi_-) = \frac{1}{4\pi\varepsilon^2}\left(S_\pm(2\pi, \Psi_+, \Psi_-) - S_\pm(-2\pi, \Psi_+, \Psi_-)\right) + O(\varepsilon^4), \tag{3.12}$$

or

$$\begin{aligned} G_\pm(\Psi_+, \Psi_-) =& \frac{1}{24\pi\varepsilon^2}\left(-S_\pm(4\pi, \Psi_+, \Psi_-) + 8S_\pm(2\pi, \Psi_+, \Psi_-) - 8S_\pm(-2\pi, \Psi_+, \Psi_-) + S_\pm(-4\pi, \Psi_+, \Psi_-)\right) \\ & + O(\varepsilon^8), \end{aligned} \tag{3.13}$$

where for some $\Psi_\pm$ and $P > 0$, $S_\pm(P, \Psi_+, \Psi_-)$ is the solution of

$$\begin{cases} \dot{\mathbf{w}}_\pm(s) = \varepsilon^2 \mathbf{F}_\pm(s, \mathbf{w}_+(s), \mathbf{w}_-(s)), & 0 < s \leq P, \\ \mathbf{w}_\pm(0) = \Psi_\pm, \end{cases} \tag{3.14}$$

and $S_\pm(-P, \Psi_+, \Psi_-)$ is the solution of

$$\begin{cases} \dot{\mathbf{w}}_\pm(s) = \varepsilon^2 \mathbf{F}_\pm(s, \mathbf{w}_+(s), \mathbf{w}_-(s)), & -P \leq s < 0, \\ \mathbf{w}_\pm(0) = \Psi_\pm, \end{cases} \tag{3.15}$$

Then a numerical approximation to (2.13) can be constructed by solving the macro-dynamics which is a reverse rescaling on (3.11):

$$\begin{cases} \dot{\widetilde{\Psi}}_\pm(t) = G_\pm(\widetilde{\Psi}_+(t), \widetilde{\Psi}_-(t)), \qquad 0 < t \le T, \\ \widetilde{\Psi}_\pm(0) = \mathbf{u}_\pm(0), \end{cases} \tag{3.16}$$

under the help of the micro-dynamics (3.14)-(3.15) for obtaining $G_\pm$ from (3.12) or (3.13).

Assume $T/\varepsilon^2 \in 2\pi\mathbb{N}$, choose a macro time step $\Delta t = H > 0$ and a micro step h for discretising the time domains $N_0 = \frac{T}{H}$, $n_0 = \frac{2\pi}{h}$, and denote the numerical solution to $\mathbf{u}_\pm^n = \mathbf{u}_\pm(t_n) \approx \widetilde{\Psi}(t_n)$ for some $t_n = nH, 0 \le n \le N_0$. A central-difference time discretization for both macro part and micro part reads: choose $\mathbf{u}_\pm^0 = \mathbf{u}_\pm(0)$ and

$$\begin{aligned} &\mathbf{u}_\pm^1 = \mathbf{u}_\pm^0 + HG_\pm^0, \\ &\mathbf{u}_\pm^{n+1} = \mathbf{u}_\pm^{n-1} + 2HG_\pm^n, \quad n = 1, \ldots, N_0, \end{aligned} \tag{3.17}$$

where

$$G_\pm^n = \frac{1}{4\pi\varepsilon^2}\left((S_\pm^h)^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n) - (S_\pm^{-h})^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n)\right), \quad n \ge 0, \tag{3.18}$$

or

$$G_\pm^n = \frac{1}{24\pi\varepsilon^2}\left(-(S_\pm^h)^{2n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n) + 8(S_\pm^h)^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n) - 8(S_\pm^{-h})^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n) + (S_\pm^{-h})^{2n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n)\right), \tag{3.19}$$

and $(S_\pm^h)^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n)$ is the numerical flow of (3.14) with $P = 2\pi, \Psi_\pm = \mathbf{u}_\pm^n$ by the central-difference discretization with time step h, and $(S_\pm^{-h})^{n_0}(\mathbf{u}_+^n, \mathbf{u}_-^n)$ is similarly the numerical flow of (3.15) with time step $-h$.

We call the SAM (3.17) with (3.18) as SAM2, and address the SAM (3.17) with (3.19) as SAM4 for short. As studied in [7, 3], with a given smooth enough function $\mathbf{F}_\pm$ in (3.10), SAM2 for integrating (3.10) gives an error bound

$$|\mathbf{u}_\pm(t_n) - u_\pm^n| \le C(H^2 + \varepsilon^2 h^2 + \varepsilon^4), \quad 0 \le n \le \frac{T}{H}, \tag{3.20}$$

and SAM4 gives an error bound

$$|\mathbf{u}_\pm(t_n) - u_\pm^n| \le C(H^2 + \varepsilon^2 h^2 + \varepsilon^8), \quad 0 \le n \le \frac{T}{H}. \tag{3.21}$$

At each (macro) time level, the memory cost of SAM2 and SAM4 is $O(1)$ and the computational cost is $O(n_0)$ where n_0 is the number of time grids in micro step. SAM4 is twice more expensive than SAM2.

3.4. Two-scale formulation

The two-scale formulation (TSF) method is a general approach to tackle highly oscillatory problems [5]. It has recently been developed to solve the kinetic models [10, 11] on characteristics.

To solve (1.1) under PIC, the two-scale formulation is performed for (2.13) as

$$\partial_t U_\pm(t,\tau) + \frac{1}{\varepsilon^2}\partial_\tau U_\pm(t,\tau) = \mathcal{F}_\pm(\tau, U_\pm(t,\tau)), \quad t > 0, \ \tau \in \mathbb{T}, \tag{3.22}$$

with initial data satisfying

$$U_+(0,0) = \mathbf{x}(0) + \varepsilon J\mathbf{y}(0) =: \mathbf{u}_+^0, \qquad U_-(0,0) = -\varepsilon J\mathbf{y}(0) =: \mathbf{u}_-^0. \tag{3.23}$$

The augmented equation (3.22) recovers the solution of (2.7) via

$$\mathbf{x}(t) = U_+(t, t/\varepsilon^2) + \mathrm{e}^{tJ/\varepsilon^2}U_-(t, t/\varepsilon^2), \qquad \mathbf{y}(t) = \frac{1}{\varepsilon}JU_-(t, t/\varepsilon^2).$$

The initial data of the augmented problem (3.22) is not fully determined. By using the Chapman-Enskog expansion, one can decompose the problem (3.22) into a macro part and a micro part, and a well prepared initial data in order to bound the time derivatives of $U_\pm(t,\tau)$ could then be constructed. To present the initial data, we introduce for some periodic function $u(\tau)$ on $\mathbb{T}$,

$$\Pi u = \frac{1}{2\pi}\int_0^{2\pi} u(s)ds \quad \text{and} \quad L^{-1}u(\tau) = (I-\Pi)\int_0^{\tau} u(s)ds \quad \text{if} \quad \Pi u = 0.$$

Denote the operator $A := L^{-1}(I-\Pi)$. The initial data is constructed in spirit of a two-scale PIC approximation as follows. One starts with

$$X^{0th}(\tau) := \mathbf{u}_+^0 + \mathrm{e}^{\tau J}\mathbf{u}_-^0.$$

Using $X^{0th}(\tau)$ through (2.10), define

$$-\nabla_{\mathbf{x}} \cdot \mathcal{E}^{0th}(\tau,\mathbf{x}) = \sum_{k=1}^{N_p} \omega_k \delta(\mathbf{x} - X_k^{0th}(\tau)) - 1. \tag{3.24}$$

With $\mathcal{E}^{0th}$, one can define the 'well-prepared' initial data $U_\pm^{1st}(\tau)$ as

$$U_\pm^{1st}(\tau) := \mathbf{u}_\pm^0 + \mathbf{h}_\pm^{0th}(\tau) - \mathbf{h}_\pm^{0th}(0), \qquad \mathbf{h}_\pm^{0th}(\tau) := \varepsilon^2 A\mathcal{F}_\pm^{0th}(\tau, \mathbf{u}_\pm^0), \tag{3.25}$$

where

$$\mathcal{F}_+^{0th}(\tau,\mathbf{u}_\pm^0) := J\mathcal{E}^{0th}\left(\tau, \mathbf{u}_+^0 + \mathrm{e}^{J\tau}\mathbf{u}_-^0\right),\ \mathcal{F}_-^{0th}(\tau, U_\pm^{1st}) := -J\mathrm{e}^{-J\tau}\mathcal{E}^{0th}\left(\tau, \mathbf{u}_+^0 + \mathrm{e}^{J\tau}\mathbf{u}_-^0\right).$$

We refer the readers to [11] for the detailed derivation.

For $t \geq 0$, the two-scale electric field is always computed from (2.10). As a property of using $U_\pm^{1st}$ as the initial data for the two-scale problem (3.22), we can have [5, 11]

$$\partial_t^k U_\pm(t,\tau) = O(1), \quad \partial_t^k \mathcal{F}_\pm(t,\tau,U_\pm) = O(1), \qquad k = 1,2,\ 0 < \varepsilon \ll 1.$$

As a consequence, we have

$$\frac{d^k}{dt^k}\mathcal{F}_\pm(t,\tau,U_\pm(t,\tau)) = O(1), \qquad k = 1,2,\ 0 < \varepsilon \ll 1. \tag{3.26}$$

An exponential integrator is then proposed to integrate the two-scale equation (3.22) with the constructed initial data (3.25).

The detailed numerical scheme reads: Denote $U_\pm^n(\tau) \approx U_\pm(t_n,\tau)$ for $n \geq 0$ and let $U_\pm^0 = U(0,\tau)$. We update the $U_\pm^n$ for $n \geq 1$ as

$$\widehat{(U_\pm)}_\ell^1 = \mathrm{e}^{-\frac{il\Delta t}{\varepsilon^2}}\widehat{(U_\pm)}_\ell^0 + p_\ell\widehat{(\mathcal{F}_\pm)}_\ell^0 + q_\ell\frac{1}{\Delta t}\left(\widehat{(\mathcal{F}_\pm)}_\ell^{*,1} - \widehat{(\mathcal{F}_\pm)}_\ell^0\right), \tag{3.27a}$$

$$\widehat{(U_\pm)}_\ell^{n+1} = \mathrm{e}^{-\frac{i\ell\Delta t}{\varepsilon^2}}\widehat{(U_\pm)}_\ell^n + p_\ell\widehat{(\mathcal{F}_\pm)}_\ell^n + q_\ell\frac{1}{\Delta t}\left(\widehat{(\mathcal{F}_\pm)}_\ell^n - \widehat{(\mathcal{F}_\pm)}_\ell^{n-1}\right), \quad n \geq 1, \tag{3.27b}$$

where

$$U_\pm^n(\tau) = \sum_{\ell=-N_\tau/2}^{N_\tau/2-1}\widehat{(U_\pm)}_\ell^n \mathrm{e}^{i\ell\tau}, \quad \mathcal{F}_\pm^n(\tau) = \sum_{\ell=-N_\tau/2}^{N_\tau/2-1}\widehat{(\mathcal{F}_\pm)}_\ell^n \mathrm{e}^{i\ell\tau}, \quad \mathcal{F}_\pm^{*,1}(\tau) = \sum_{\ell=-N_\tau/2}^{N_\tau/2-1}\widehat{(\mathcal{F}_\pm)}_\ell^{*,1} \mathrm{e}^{i\ell\tau}, \quad n \geq 0,$$

and

$$\mathcal{F}_+^n(\tau) = J\mathcal{E}^n\left(\tau, U_+^n(\tau) + \mathrm{e}^{J\tau}U_-^n(\tau)\right),\ \mathcal{F}_-^n(\tau) := -J\mathrm{e}^{-J\tau}\mathcal{E}^n\left(\tau, U_+^n(\tau) + \mathrm{e}^{J\tau}U_-^n(\tau)\right),\ n \geq 0,$$
$$\mathcal{F}_+^{*,1}(\tau) = J\mathcal{E}^{*,1}\left(\tau, U_+^{*,1}(\tau) + \mathrm{e}^{J\tau}U_-^{*,1}(\tau)\right),\ \mathcal{F}_-^{*,1}(\tau) := -J\mathrm{e}^{-J\tau}\mathcal{E}^{1,*}\left(\tau, U_+^{*,1}(\tau) + \mathrm{e}^{J\tau}U_-^{*,1}(\tau)\right),$$

with

$$\widehat{(U_\pm)}_\ell^{*,1} = \mathrm{e}^{-\frac{i\ell\Delta t}{\varepsilon^2}}\widehat{(U_\pm)}_\ell^{0} + p_\ell\widehat{(\mathcal{F}_\pm)}_\ell^{0}, \quad U_\pm^{*,1}(\tau) = \sum_{\ell=-N_\tau/2}^{N_\tau/2-1}\widehat{(U_\pm)}_\ell^{*,1}\mathrm{e}^{i\ell\tau},$$

and

$$p_\ell := \int_0^{\Delta t}\mathrm{e}^{-\frac{i\ell}{\varepsilon^2}(\Delta t-s)}ds = \begin{cases}\dfrac{i\varepsilon^2}{\ell}\left(\mathrm{e}^{-\frac{i\ell\Delta t}{\varepsilon^2}}-1\right), & \ell\neq 0,\\ \Delta t, & \ell=0,\end{cases}$$

$$q_\ell := \int_0^{\Delta t}\mathrm{e}^{-\frac{i\ell}{\varepsilon^2}(\Delta t-s)}sds = \begin{cases}\dfrac{\varepsilon^2}{\ell^2}\left(\varepsilon^2-\varepsilon^2\mathrm{e}^{-\frac{i\ell\Delta t}{\varepsilon^2}}-i\ell\Delta t\right), & \ell\neq 0,\\ \dfrac{\Delta t^2}{2}, & \ell=0.\end{cases}$$

The numerical electric field $\mathcal{E}^n$ is given by collecting all the two-scaled characteristics $\{U^n_{\pm,k}\}_{k=1}^{N_p}$ and solving the Poisson equation:

$$\mathcal{E}^n(\tau,\mathbf{x}) = -\nabla_{\mathbf{x}}\Phi^n(\tau,\mathbf{x}),\ -\Delta\Phi^n(\tau,\mathbf{x}) = \sum_{k=1}^{N_p}\omega_k S(\mathbf{x}-X_k^n(\tau))-1,\ X_k^n(\tau) = U^n_{+,k}(\tau)+\mathrm{e}^{\tau J}U^n_{-,k}(\tau),$$

$$\mathcal{E}^{*,1}(\tau,\mathbf{x}) = -\nabla_{\mathbf{x}}\Phi^*(\tau,\mathbf{x}),\ -\Delta\Phi^*(\tau,\mathbf{x}) = \sum_{k=1}^{N_p}\omega_k S(\mathbf{x}-X_k^*(\tau))-1,\ X_k^*(\tau) = U^{*,1}_{+,k}(\tau)+\mathrm{e}^{\tau J}U^{*,1}_{-,k}(\tau), \quad (3.28)$$

where $S(\mathbf{x})$ is the regularised basis.

The proposed TSF scheme (3.27)-(3.28) is self-consistent for solving (3.22). That is to say during the computation, the scheme runs with $U^n_\pm(\tau)$ only. It does not require approximations of the original components $\mathbf{x}(t),\mathbf{y}(t)$ for (2.7) or $X(t,\tau),Y(t,\tau)$. Moreover, the scheme is fully explicit and efficient thanks to fast Fourier transform in τ. It has been shown in [11] that the TSF scheme (3.27)-(3.28) for solving (3.22) with the 'well-prepared' initial data $U_\pm^{1st}$ offers uniformly second order accuracy in the time discretization for all $0\leq\varepsilon\leq 1$. The finite time error bound of the TSF for solving (2.10) formally goes as

$$|\mathbf{u}_\pm(t_n)-U^n_\pm(t_n/\varepsilon^2)|\leq C(\Delta t^2+N_\tau^{-m_0}), \quad 0\leq n\leq\frac{T}{\Delta t},$$

where $m_0>0$ depends on the smoothness of the solution of (3.22). The memory cost is $O(N_\tau)$ and computational cost per time step is $O(N_\tau\log N_\tau)$.

With $\{U^n_{\pm,k}\}_{k=1}^{N_p}$ from the scheme (3.27)-(3.28), one can obtain the approximation to the solution of the original characteristics (2.2) as

$$\mathbf{x}_k(t_n)\approx X_k^n\left(\frac{t_n}{\varepsilon^2}\right), \quad \mathbf{v}_k(t_n)\approx\frac{J}{\varepsilon}\mathrm{e}^{J\frac{t_n}{\varepsilon^2}}U^n_{-,k}\left(\frac{t_n}{\varepsilon^2}\right), \qquad n\geq 1.$$

Then together with the PIC approximation (2.1), the numerical scheme of a TSF method for solving the Vlasov-Poisson equation (1.1) is completed.

We remark that the reason why we choose to solve (3.22) by an exponential integrator rather a finite difference integrator such as the central difference scheme used in [5, 9], is because the error bound of a second order finite difference discretization depends on the norm of the third order time derivative $\partial_t^3U_\pm$, whereas for the exponential integrator, the error depends on the second order time derivative $\partial_t^2U_\pm$ (see [11]). However, the constructed initial data (3.25) is not enough to ensure a uniform bound on the third order time derivative term, and one needs to construct a higher order 'well-prepared' initial data which is more involved.

4. Numerical results

In this section, we shall conduct numerical experiments to test the accuracy, efficiency and long-time behaviour of all the mentioned numerical methods for solving the Vlasov-Poisson equation (1.1) with a wide range of $\varepsilon \in (0, 1]$. Comparisons between different methods shall be made. Throughout this section, we shall use the following numerical example unless specified.

We consider the Vlasov-Poisson equation (1.1) with the initial data of a Kelvin-Helmholtz instability type [12, 23] in space and a double-Maxwellian distribution in velocity space:

$$f_0(\mathbf{x}, \mathbf{v}) = \frac{1}{4\pi}\left(1 + \sin(x_2) + \eta\cos(kx_1)\right)\left(\mathrm{e}^{-\frac{(v_1+2)^2+v_2^2}{2}} + \mathrm{e}^{-\frac{(v_1-2)^2+v_2^2}{2}}\right). \tag{4.1}$$

The computational domain for $\mathbf{x}$ is $\Omega = [0, 2\pi/k] \times [0, 2\pi]$ for some $k, \eta > 0$. We choose $\eta = 0.05, k = 0.5$ and discretise Ω with 64 points in x_1-direction and 32 points in x_2-direction. The Poisson equation (1.2) is always solved on Ω with periodic boundary condition by the fast Fourier transform. We fix the basis function $S(\mathbf{x})$ for the PIC method as the fifth order B-spline, i.e. $m = 5$ in (2.5). The high order spline function is chosen here to reach a high accuracy in spatial approximation so that the temporal error of each method can be distinguished. We take $N_p = 204800$ and generate the initial position and velocity of each particle by means of the rejection sampling. All the numerical methods are programmed in a sequential way in Fortran and are run on an iMac with processor 2.5Ghz Intel Core i5.

4.1. Accuracy test

We first test the short time temporal accuracy of the PIC method coupled with SIRK (3.1), ETD (3.9), SAM2 (3.18), SAM4 (3.19) and TSF (3.27). We solve the Vlasov-Poisson equation (1.1) with (4.1) by the numerical methods with the setup in PIC as described before till $T = \frac{\pi}{2}$. We compute the error of each method regarding the density

$$\rho^\varepsilon(t, \mathbf{x}) = \int_{\mathbb{R}^2} f^\varepsilon(t, \mathbf{x}, \mathbf{v})d\mathbf{v}, \qquad \mathbf{x} \in \Omega,$$

and the error regarding the second moment in $\mathbf{v}$

$$\rho_v^\varepsilon(t, \mathbf{x}) = \int_{\mathbb{R}^2} |\mathbf{v}|^2 f^\varepsilon(t, \mathbf{x}, \mathbf{v})d\mathbf{v}, \qquad \mathbf{x} \in \Omega,$$

where a reference solution is obtained numerically via the TSF with very small time step, e.g. $\Delta t = 10^{-3}$ and $N_\tau = 64$. We measure the error in ρ^ε and ρ_v^ε at $t = T = \frac{\pi}{2}$ under maximum norm in space Ω. For ETD (3.9), SAM2 (3.18) and SAM4 (3.19) methods, we denote by n_0 the number of grid points (in time) for the micro-solver and define it as $n_0 = T/\Delta t$, and hence $\Delta t = H = O(h)$. The errors of SIRK, ETD, SAM2 and SAM4 under different Δt and ε are shown in Figures 1, 3, 4 and 5, respectively. Note that only the error in ρ^ε is shown in Figure (1) for the SIRK method; indeed, the method is designed so as to capture the slow dynamic through the Guiding Center model, then plotting the error in ρ_v^ε is not relevant. The results of TSF under $N_\tau = 32$ are shown in Figure 6.

In order to understand the asymptotic error bound of the SIRK scheme in the limit regime and then to have an estimate of the q exponent in (3.2), we investigate the convergence rate of the Vlasov-Poisson equation (1.1) to (1.3) as $\varepsilon \to 0$. To do so, we solve the Vlasov-Poisson equation (1.1) accurately by TSF, and solve the limit model (1.3) accurately by classical PIC with fourth order Runge-Kutta scheme. The convergence results in the two quantities ρ^ε and ρ_v^ε as $\varepsilon \to 0$ are shown in Figure 2 which indicate the rate q is one. Next, in order to the test convergence of the TSF method in τ-direction, we show the error of the TSF method under different N_τ in Figure 7.

Based on the convergence results in Figures 1-7, we can draw the following conclusions:

Concerning SIRK method, the second order accuracy is well observed in the classical regime $\varepsilon = O(1)$, while in the intermediate regime of ε this scheme loses convergence. The reasons are from two sides. One is due to the temporal oscillations, and the other is due to the formulation (2.2) where the approximation

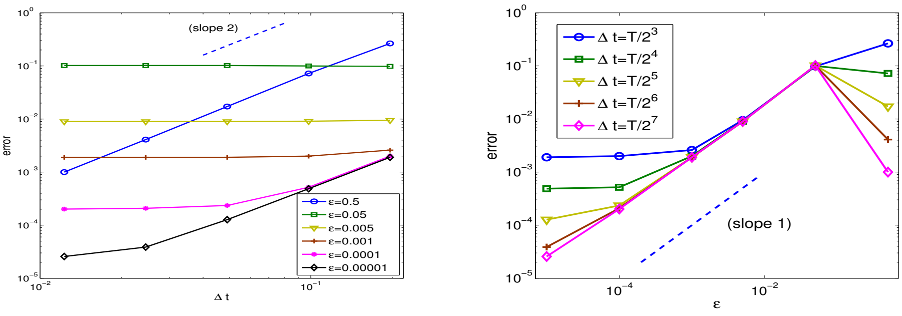

Figure 1: Temporal error of SIRK-PIC at $T = \pi/2$ with respect to $\Delta t = H = h$ and ε: maximum error in the density ρ^ε.

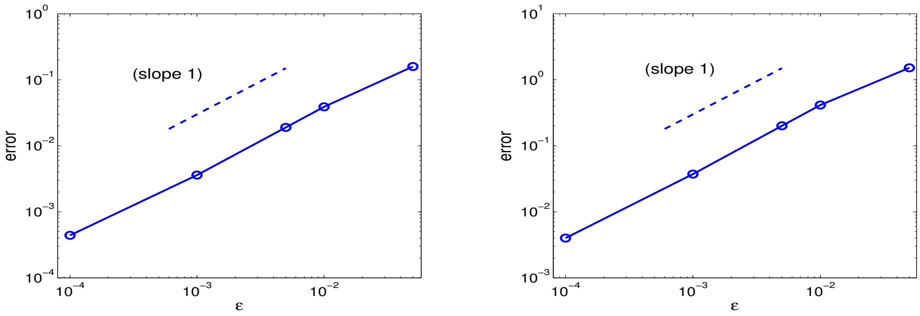

Figure 2: Convergence from the Vlasov-Poisson equation (1.1) to the limit model (1.3): maximum error with respect to ε at $T = \pi/2$ in the density ρ^ε (left) and in the kinetic energy density ρ_v^ε (right).

error of $\mathbf{E}^\varepsilon$ (computed numerically through Poisson equation) is amplified by the term $1/\varepsilon$. As ε becomes very small, the SIRK scheme recovers some convergence rate in the density function ρ^ε but the error always scales as $O(\varepsilon)$ due to the convergence rate of the model as explained before thanks to Figure 2. Indeed, we see the line in the left figure of Figure 2 matches with the slope in the right figure of Figure 1.

Concerning ETD, second order accuracy is also observed for $\varepsilon = O(1)$, but order reduction occurs (first order) in the limit regime $0 < \varepsilon \ll 1$, which is consistent with the observation in [18]. In the intermediate regime of ε, the method has strong convergence order reduction. There is a heavy bump in the error when the ETD reduces to classical integrator, but it is accurate in both ρ^ε and ρ_v^ε if the time step is small enough.

Concerning SAM methods, the errors of SAM2 and SAM4 stuck at $O(\varepsilon^4)$ and $O(\varepsilon^8)$ respectively. This is consistent with the fact that these methods only solve the averaged model which is valid when ε is small. Thus, they do not work for $\varepsilon = O(1)$, but in the limit regime when $O(\varepsilon^4)$ and $O(\varepsilon^8)$ are negligible, the SAM methods can have second order convergence rate in time and offer very accurate approximation. In the intermediate regime of ε, SAM4 is better than SAM2 to reach a higher accuracy. In this numerical test, SAM4 produces accurate results and works well for $0 < \varepsilon \leq 0.1$.

Concerning TSF method, a uniform second order accurate in time for all $0 < \varepsilon \leq 1$ is obtained for both ρ^ε and ρ_v^ε. The discretization error in τ-direction converges spectrally. In particular, the smaller the ε is, the less grid points are needed in τ. However, when $\varepsilon = O(1)$, a substantial number of points in the τ direction is required to make the error in this direction small enough so as the uniform accuracy is recovered.

To compare the accuracy and efficiency of each numerical method in this test, we list of the errors in ρ^ε together with computational time of all the numerical methods under some Δt in Tables 1, 2 and 3, respectively representing the classical regime, the intermediate regime and the limit regime of ε.

From the comparisons in Tables 1-3, the following comments can be done.

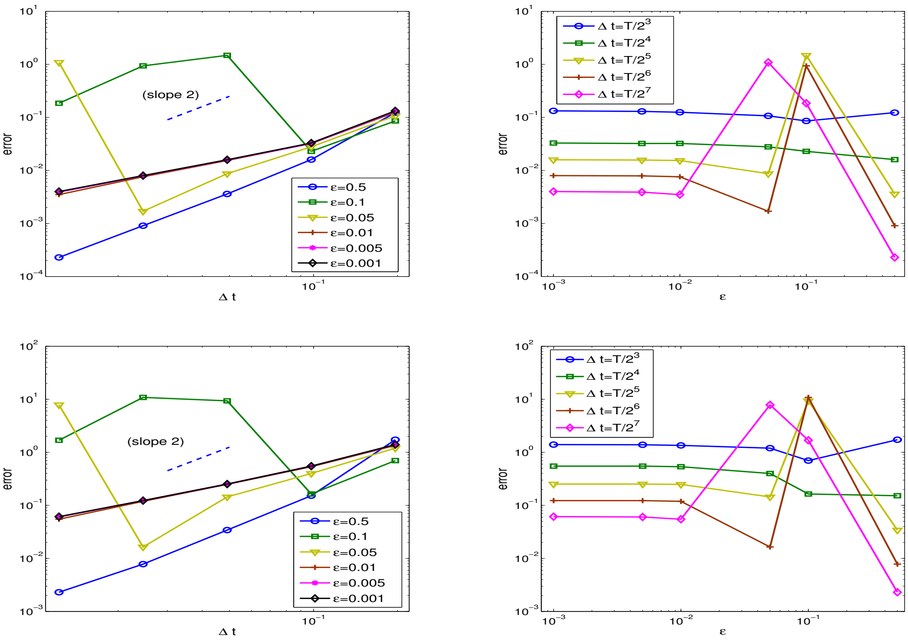

Figure 3: Temporal error of EDT-PIC at $T = \pi/2$ with respect to $\Delta t = H = h$ and ε: maximum error in the density ρ^ε (first row) and in the kinetic energy density ρ_v^ε (second row).

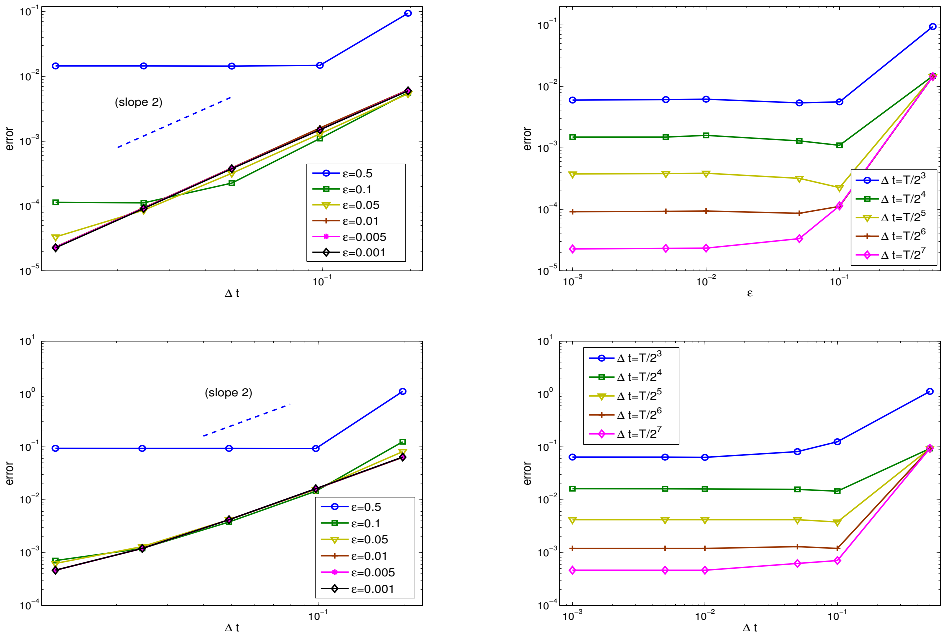

Figure 4: Temporal error of SAM2-PIC at $T = \pi/2$ with respect to $\Delta t = H = h$ and ε: maximum error in the density ρ^ε (first row) and in the kinetic energy density ρ_v^ε (second row).

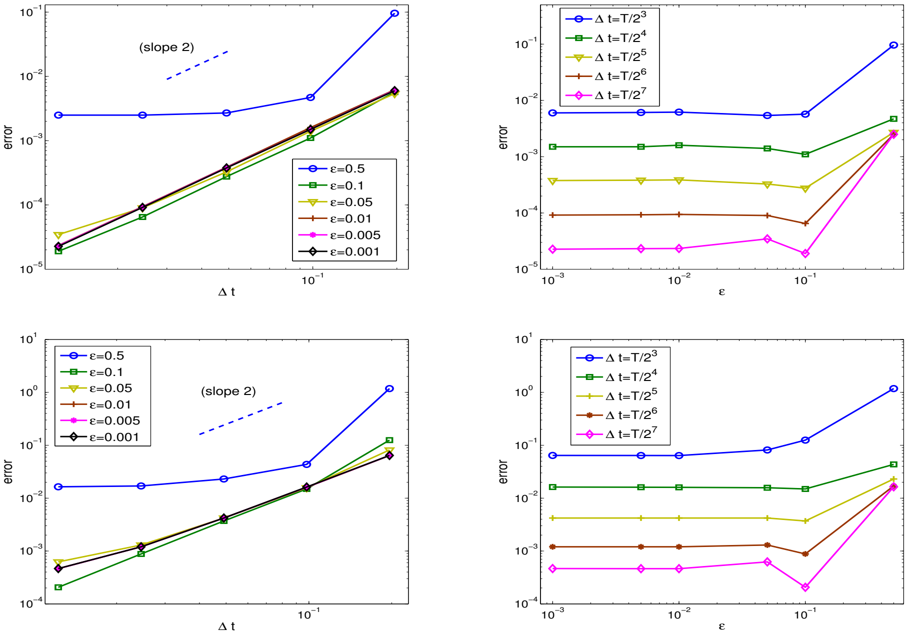

Figure 5: Temporal error of SAM4-PIC at $T = \pi/2$ with respect to $\Delta t = H = h$ and ε: maximum error in the density ρ^ε (first row) and in the kinetic energy density ρ_v^ε (second row).

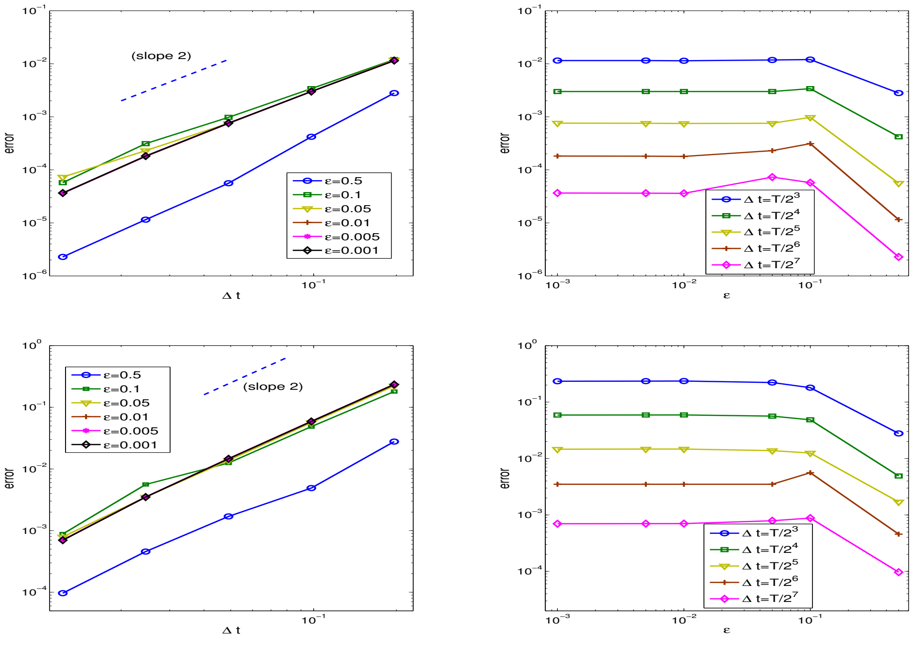

Figure 6: Temporal error of UA-PIC at $T = \pi/2$ with respect to Δt and ε: maximum error in the density ρ^ε (first row) and in the kinetic energy density ρ_v^ε (second row).

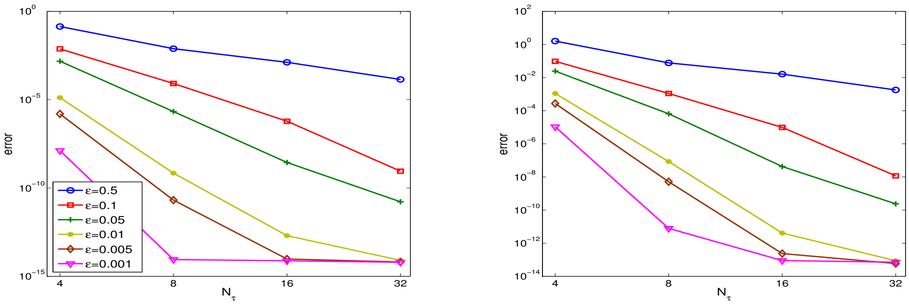

Figure 7: Discretization error of UA-PIC in τ-direction with respect to N_τ at $T = \pi/2$: maximum error in the density ρ^ε (left) and in the kinetic energy density ρ_v^ε (right).

First, in the classical regime of ε, TSF, SIRK and ETD have similar accuracy, but TSF is much more expensive due to additional variable τ (which has to be sampled with $N_\tau = 64$). SIRK and ETD are then the most efficient method in this case. As ε diminishes (see Table 1), we see that the error of SIRK and ETD for a fixed Δt increase. As expected, SAM methods do not work in $\varepsilon = O(1)$ regime.

Second, in the intermediate regime of ε, the two SAM methods and TSF are more accurate than other methods. SAM4 is more accurate than SAM2 as Δt is small, but SAM4 is twice more expensive than SAM2. When Δt becomes small, TSF is more efficient than the SAM methods as we see in Table 2. This is due to the total computational cost of SAM methods which scales as $O(n_0^2)$ where $n_0 = T/\Delta t$, due to micro step at each macro time level. For TSF, the total computational cost is $O(n_0 N_\tau)$ where N_τ is fixed (small enough from the results in Figure 7) as Δt reduces. Hence, the computation time of SAM increases quadratically as Δt decreases, while that of TSF increases linearly. ETD shows second order accuracy before and after the regime $\Delta t \approx 2\pi\varepsilon^2$ but with a significant change in error and computational time. Indeed, ETD has total computational cost $O(n_0^2)$ similarly as SAM2 when $\Delta t > 2\pi\varepsilon^2$, but for $\Delta t < 2\pi\varepsilon^2$ as the scheme degenerates to classical finite difference integrator the total cost is $O(n_0)$. This comes from the fact that in (3.4), N vanishes when $\Delta t < 2\pi\varepsilon^2$; then, the error can become large since the size of the macro time step Δt is similar to the period, which is not enough to get accurate results.

Lastly, in the limit regime of ε, ETD converges linearly and it is as costly as SAM2 for $\Delta t > 2\pi\varepsilon^2$. SIRK could produce an accurate approximation until the error is stuck at $O(\varepsilon)$. However, it is much more efficient than SAM methods. If one wants error further less than $O(\varepsilon)$, TSF becomes the most efficient scheme since the workload due to the additional variable τ is much reduced.

4.2. Long-time behavior

In this part, we focus on more quantitative results and consider the long time behavior of (1.1) with the initial condition (4.1). We look at the time evolution of the total energy defined as

$$H(t) := \frac{1}{2}\int_{\mathbb{R}^2\times\mathbb{R}^2} |\mathbf{v}|^2 f^\varepsilon(t,\mathbf{x},\mathbf{v})d\mathbf{x}d\mathbf{v} + \frac{1}{2}\int_{\mathbb{R}^2} |\mathbf{E}^\varepsilon(t,\mathbf{x})|^2 d\mathbf{x}, \quad t \geq 0. \tag{4.2}$$

The total energy $H(t)$ of the Vlasov-Poisson equation (1.1) is conserved for all ε at the continuous level and is a good diagnostic to compare the numerical methods.

We solve the Vlasov-Poisson equation (1.1) numerically till $T = 32\pi$ with time step $\Delta t = \pi/2^7$ ($H = h = \Delta t$) and the same numerical parameters as previously and obtain the approximated energy $H(t)$ at different time. We compute the relative energy error

$$|H(t) - H(0)|/|H(0)|, \qquad 0 < t = \frac{n\pi}{2} \leq T,$$

for the different numerical methods under different regimes of ε.

Table 1: Comparison in classical regime of ε: temporal errors in ρ^ε and computational time (seconds) of different PIC methods for the Vlasov-Poisson (1.1) with $\Delta t = H = h$ and $N_\tau = 64$.

$\varepsilon = 1$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	2.72E-1	1.90E-1	1.94E-1	1.95E-1	1.95E-1
time (cpu)	11.3	45.6	182	740	2930
SAM4	2.79E-1	1.8E-1	1.84E-1	1.85E-1	1.85E-1
time (cpu)	23.2	93.6	376	1470	5840
TSF	2.20E-3	5.82E-4	1.48E-4	3.56E-5	8.00E-6
time (cpu)	129	264	516	962	1918
SIRK	1.10E-2	2.9E-3	7.57E-4	1.85E-4	4.74E-5
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	1.70E-2	4.20E-3	1.10E-3	2.80E-4	7.50E-5
time (cpu)	0.8	1.2	3.3	6.1	11.8
$\varepsilon = 0.5$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	9.43E-2	1.48E-2	1.44E-2	1.45E-2	1.45E-2
time (cpu)	11.3	45.6	182	740	2930
SAM4	9.61E-2	4.70E-3	2.70E-3	2.50E-3	2.50E-3
time (cpu)	23.2	93.6	376	1470	5840
TSF	2.80E-3	4.20E-4	5.60E-5	1.15E-5	2.28E-6
time (cpu)	129	264	516	962	1918
SIRK	2.66E-1	7.22E-2	1.72E-2	4.10E-3	1.00E-3
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	1.23E-1	1.60E-2	3.60E-3	9.08E-4	2.29E-4
time (cpu)	0.8	1.2	3.3	6.1	11.8

On the one side, the SAM methods (3.18) and (3.19) are not valid in classical regime $\varepsilon = O(1)$ and on the other side, the SIRK method (3.1) is not designed to capture the oscillations when ε is small. Thus, we show the time history of the relative energy error (in semi-log-scale) obtained with SIRK, ETD and TSF for $\varepsilon = 1$ and show the results obtained with ETD, SAM2, SAM4 and TSF for $\varepsilon = 0.1$, $\varepsilon = 0.01$ and $\varepsilon = 0.001$ in Figure 8.

In the case $\varepsilon = 1$, as SIRK is a L-stable method, it then enjoys a good long time behavior. It can be observed than both ETD and TSF preserved in a good way the total energy. Note that for $\varepsilon = 1$ and $\varepsilon = 0.1$, ETD reduces to the classical central difference integrator (since we use $\Delta t = \pi/2^7$). When ε becomes smaller ($\varepsilon = 0.01$ and $\varepsilon = 0.001$), ETD shows a slight increasing in the energy error.

Concerning the SAM methods, we observe that the error increases dramatically after some time. This is because the stroboscopic approximation (3.18) or (3.19) breaks (2.13). The perturbed problem (3.16) introduces non-pure imaginary eigenvalues λ, while the central difference integrator for

$$\dot{u}(t) = \lambda u(t), \quad t > 0,$$

is not stable when λ is outside the imaginary axis. One solution to improve the long time performance of SAM methods is to use schemes with larger stability region as the macro integrator for (3.16). For example, one can use the second order Runge-Kutta method (RK2) with time step small enough, but the two-stage method would obviously double the computational time of the whole SAM scheme. The energy error of the SAM2 scheme (3.18) with RK2 as the macro-solver (same $\Delta t = \pi/2^7 = H = h$) is shown in Figure 9 under different ε. From the results, we see the long-time behavior of the SAM method is significantly improved although there is still a small drift in the energy error. As being applied to study the Schrödinger equation in [7], the macro solver with the best long-time behavior is suggested as the implicit midpoint rule which is A-stable (which is symplectic). However, the implicit midpoint rule will make the integrator fully implicit,

Table 2: Comparison in intermediate regime of ε: temporal errors in ρ^ε and computational time (seconds) of different PIC methods for the Vlasov-Poisson (1.1) with $\Delta t = H = h$ and $N_\tau = 16$.

$\varepsilon = 0.1$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	5.60E-3	1.10E-3	2.26E-4	1.12E-4	1.14E-4
time (cpu)	11.3	45.6	182	740	2930
SAM4	5.70E-3	1.10E-3	2.76E-4	6.50E-5	1.90E-5
time (cpu)	23.2	93.6	376	1470	5840
TSF	1.20E-2	3.40E-3	9.77E-4	3.12E-4	5.73E-5
time (cpu)	39.8	72.4	142	266	532
SIRK	1.97E-1	1.95E-1	1.94E-1	1.92E-1	2.15E-1
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	8.58E-2	2.29E-2	1.48	9.37E-1	1.85E-1
time (cpu)	12	47.6	3.3	6.1	11.7
$\varepsilon = 0.05$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	5.40E-3	1.30E-3	3.20E-4	8.63E-5	3.34E-5
time (cpu)	11.3	45.6	182	740	2930
SAM4	5.40E-3	1.40E-3	3.27E-4	9.01E-5	3.46E-5
time (cpu)	23.2	93.6	376	1470	5840
TSF	1.18E-2	3.00E-3	7.55E-4	2.30E-4	7.31E-5
time (cpu)	39.8	72.4	142	266	532
SIRK	9.83E-2	1.00E-1	1.02E-1	1.02E-1	1.02E-1
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	1.07E-1	2.79E-2	8.70E-3	1.70E-3	1.09
time (cpu)	12	47.6	188	727	11.8

which would lead to the expensive inversion of a nonlinear system of size N_p in our context.

Finally, the TSF method (with $N_\tau = 32$) behaves well in the long time run except for the case $\varepsilon = 0.1$. In the intermediate regime of ε, the TSF method has a significant increase in the energy which has been reported in [11]. This is not due to the stability of the integrator and could be overcome by a refresh procedure. We can choose a $t_0 > 0$ to stop the algorithm and obtain the numerical solution of (2.2) at t_0. Since (2.2) is autonomous, we then treat the approximated solution at t_0 as the initial data for (1.1) and restart the TSF method for solving (1.1) from $t = 0$. That is to say we reformulate (2.2) into (2.13), construct of the initial data (3.25) for the two-scale problem (3.22) and run (3.27) from $t = 0$ till $t = t_0$ to repeat the procedure. We plot the energy error without this refresh procedure in Figure 10. From the result in Figure 10, we observe that the energy error of TSF when $\varepsilon = 0.1$ starts to increase at around $t = 6$, so we choose the refresh time $t_0 = 6$. The relative energy error of the TSF method with a refresh every $t_0 = 6$ is shown in Figure 8. Clearly from the results, the TSF with refresh totally overcome the increase of the error and significantly improves its long-time behavior in the problematic intermediate regime.

At last, we compute the dynamics of the following initial data (which is a semi-Gaussian beam in $\mathbf{v}$, as in [10]) in the Vlasov-Poisson equation (1.1):

$$f_0(\mathbf{x}, \mathbf{v}) = \frac{1}{2r_0\sqrt{2\pi}\sigma}\left(1 + \sin(x_2) + \eta\cos(kx_1)\right) \mathrm{e}^{-\frac{(v_1)^2}{2\sigma^2}} \mathbb{1}_{[-r_0,r_0]}(v_2), \tag{4.3}$$

where the function $\mathbb{1}_{[-r_0,r_0]}(v_2) = 1$, if $v_2 \in [-r_0, r_0]$, otherwise $\mathbb{1}_{[-r_0,r_0]}(v_2) = 0$. We choose $\eta = 0.05, kx_1 = 0.5, \sigma = 0.2, r_0 = 1.4$ in (4.3) and we used the same numerical parameters as previously. We then solve the equation by the TSF method (with $N_\tau = 32$) for $\varepsilon = 0.01$ and look at the time history of $\rho^\varepsilon(t, \mathbf{x}) ==$

Table 3: Comparison in limit regime of ε: temporal errors in ρ^ε and computational time (seconds) of different PIC methods for the Vlasov-Poisson (1.1) with $\Delta t = H = h$ and $N_\tau = 4$.

$\varepsilon = 0.001$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	6.00E-3	1.50E-3	3.77E-4	9.16E-5	2.27E-5
time (cpu)	11.3	45.6	182	740	2930
SAM4	6.00E-3	1.50E-3	3.77E-4	9.16E-5	2.27E-5
time (cpu)	23.2	93.6	376	1470	5840
TSF	1.15E-2	3.00E-3	7.60E-4	1.82E-4	3.66E-5
time (cpu)	12.4	22.6	41.7	81.6	162
SIRK	2.60E-3	2.00E-3	1.90E-3	1.90E-3	1.90E-3
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	1.33E-1	3.29E-2	1.59E-2	8.00E-3	4.00E-3
time (cpu)	13.7	48.5	193	738	2910
$\varepsilon = 0.0001$	$\Delta t = T/2^3$	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/16$
SAM2	5.00E-3	1.50E-3	3.76E-4	9.14E-5	2.25E-5
time (cpu)	11.3	45.6	182	740	2930
SAM4	6.00E-3	1.50E-3	3.76E-4	9.14E-5	2.25E-5
time (cpu)	23.2	93.6	376	1470	5840
TSF	1.16E-2	3.00E-3	7.61E-4	1.82E-4	3.66E-5
time (cpu)	12.4	22.6	41.7	81.6	162
SIRK	2.00E-3	5.18E-4	2.35E-4	2.08E-4	2.01E-4
time (cpu)	1.4	2.8	5.6	11.2	22.4
ETD	1.33E-1	3.18E-2	1.58E-2	8.30E-3	4.10E-3
time (cpu)	13.7	48.5	193	738	2910

$\int_{\mathbb{R}^2} f^\varepsilon(t,\mathbf{x},\mathbf{v})d\mathbf{v}$ and

$$\chi^\varepsilon(t,\mathbf{v}) := \int_{\mathbb{R}^2} f^\varepsilon(t,\mathbf{x},\mathbf{v})d\mathbf{x},$$

The results are shown in Figures 11 and 12. First, we remark that the dynamics for $\rho^e ps$ is very close to the one obtained with an isotropic initial distribution (see [? 11]). However, the dynamics of χ^ε is strongly related to the velocity distribution of f_0. In this case, the anisotropic character of f_0 induces some effects in the velocity direction which an be captured by the sole Guiding Center model. Note that in this regime, ETD and SAM methods produce very similar results.

5. Conclusion

We considered the long-time Vlasov-Poisson equation with a strong homogeneous magnetic field. The equation contains a small parameter ε which endows fast oscillations in time to the solution. We studied numerically several multiscale Particle-In-Cell (PIC) methods for solving this problem. The methods we considered include asymptotic preserving Runge-Kutta schemes, exponential time discretization schemes, stroboscopic averaging methods and uniformly accurate two-scale formulation method. Systematical comparisons in accuracy, efficiency and long time behavior between the methods were performed for a wide range of $0 < \varepsilon \leq 1$ aiming to identify the advantages and drawbacks of the different methods.

Acknowledgements

This work was partly supported by the French ANR project MOONRISE ANR-14-CE23-0007-01. N. Crouseilles and S. Hirstoaga are supported by the Enabling Research EUROFusion project CfP-WP14-ER-01/IPP-03. X. Zhao is supported by the IPL FRATRES.

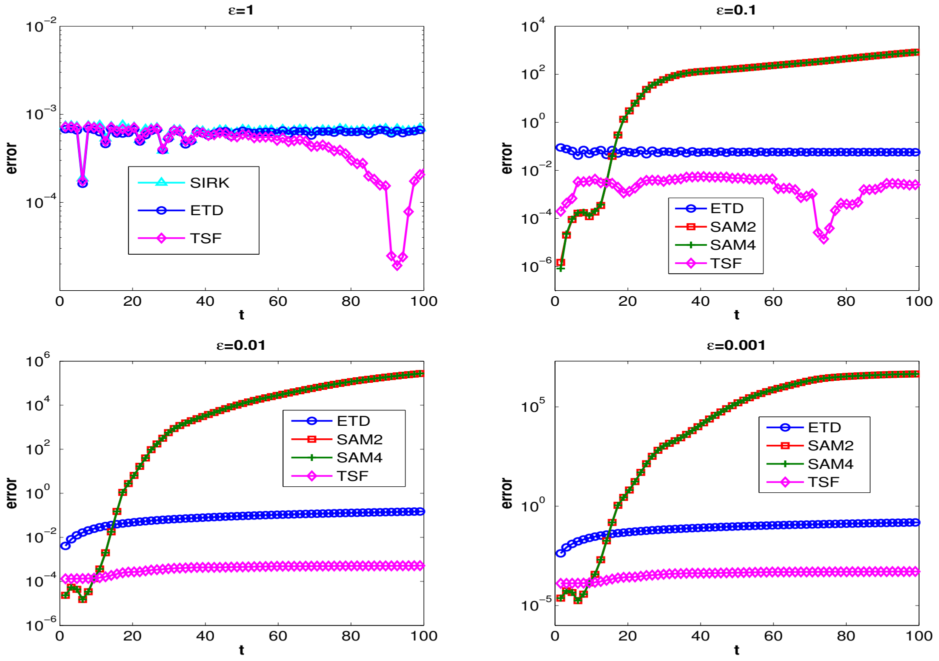

Figure 8: Relative energy error of the numerical methods under different ε.

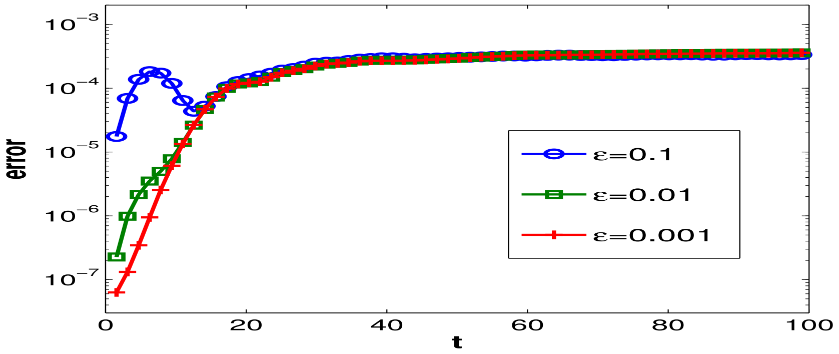

Figure 9: Relative energy error of SAM2 with RK2 as macro-integrator under different ε.

[1] A. Abdulle, W. E, B. Engquist, E. Vanden-Eijnden, The heterogeneous multiscale method, Acta Numer. 21 (2012) pp. 1-87.

[2] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, Adam Hilger, 1991.

[3] M.P. Calvo, P. Chartier, A. Murua, J.M. Sanz-Serna, A stroboscopic numerical method for highly oscillatory problems, Lect. Notes Comput. Sci. Eng. 82, Springer 2011, pp. 73-87.

[4] F. Castella, P. Chartier, F. Méhats, A. Murua, Stroboscopic averaging for the nonlinear Schrödinger equation, Found. Comput. Math. 15 (2015) pp. 519-559.

[5] Ph. Chartier, N. Crouseilles, M. Lemou, F. Méhats, Uniformly accurate numerical schemes for highly oscillatory Klein-Gordon and nonlinear Schrödinger equations, Numer. Math. 129 (2015) pp. 211-250.

[6] P. Chartier, J. Makazaga, A. Murua, G. Vilmart, Multi-revolution composition methods for highly oscillatory differential equations, Numer. Math. 128 (2014) pp.167-192.

[7] Ph. Chartier, N. Mauser, F. Méhats, Y. Zhang, Solving highly-oscillatory NLS with SAM: Numerical efficiency and long-time behavior, Discret. Contin. Dyn. Syst. Ser. S 5 (2016) pp. 1327-1349.

[8] P. Chartier, F. Méhats, M. Thalhammer, Y. Zhang, Convergence analysis of multi-revolution composition time-splitting pseudo-spectral methods for highly oscillatory differential equations of Schrdinger equations, preprint (2017).

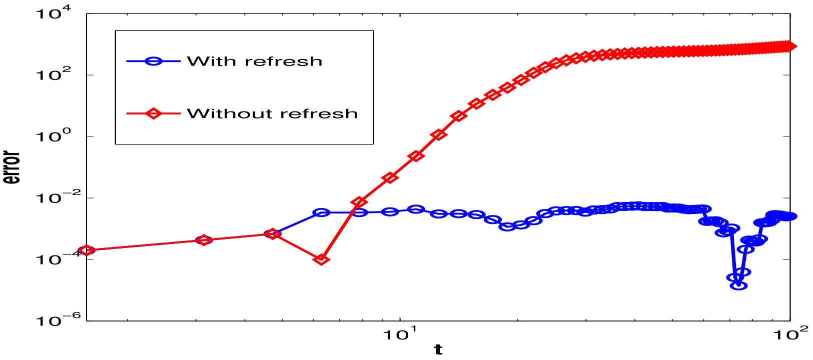

Figure 10: Relative energy error of TSF with refresh under $\varepsilon = 0.1$.

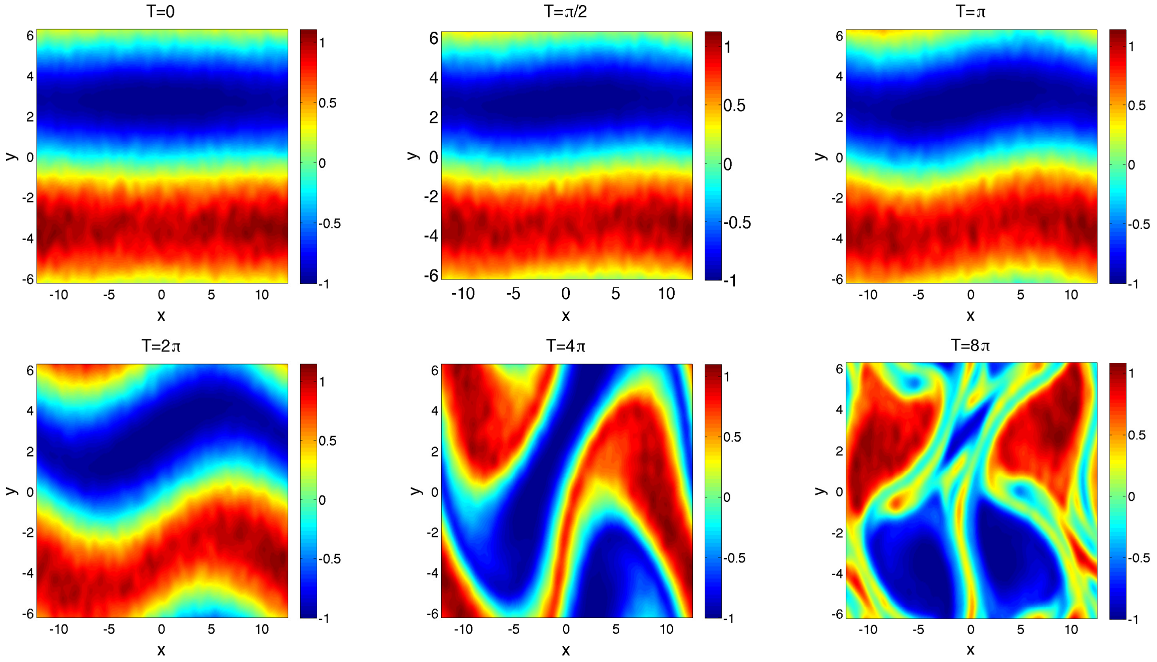

Figure 11: Contour plot of quantity $\rho^\varepsilon(t, \mathbf{x}) - 1$ in example (4.3) at different t with $\varepsilon = 0.01$.

[9] N. Crouseilles, M. Lemou, F. Méhats, Asymptotic preserving schemes for highly oscillatory Vlasov-Poisson equations, J. Comput. Phys. 248 (2013) pp. 287-308.

[10] N. Crouseilles, M. Lemou, F. Méhats, X. Zhao, Uniformly accurate forward semi-Lagrangian methods for highly oscillatory Vlasov-Poisson equations, to appear in SIAM Multiscale Model. Simul. (2017) hal-01286947.

[11] N. Crouseilles, M. Lemou, F. Méhats, X. Zhao, Uniformly accurate Particle-in-Cell method for the long time two-dimensional Vlasov-Poisson equation with strong magnetic field, preprint, hal-01418976 (2017).

[12] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, Conservative semi-Lagrangian schemes for Vlasov equations, J. Comput. Phys. 229 (2010) pp. 1927-1953.

[13] P. Degond, P.A. Raviart, The paraxial approximation of the Vlasov-Maxwell equations, Math. Models Methods Appl. Sci. 3 (1993) pp. 513-562.

[14] W. E, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, Heterogeneous multiscale methods: a review, Commun. Comput. Phys. 2(2007) pp. 367-450.

[15] F. Filbet, L.M. Rodrigues, Asymptotically stable particle-in-cell methods for the Vlasov-Poisson system with a strong external magnetic field, SIAM J. Numer. Anal. 54 (2016) pp. 1120-1146.

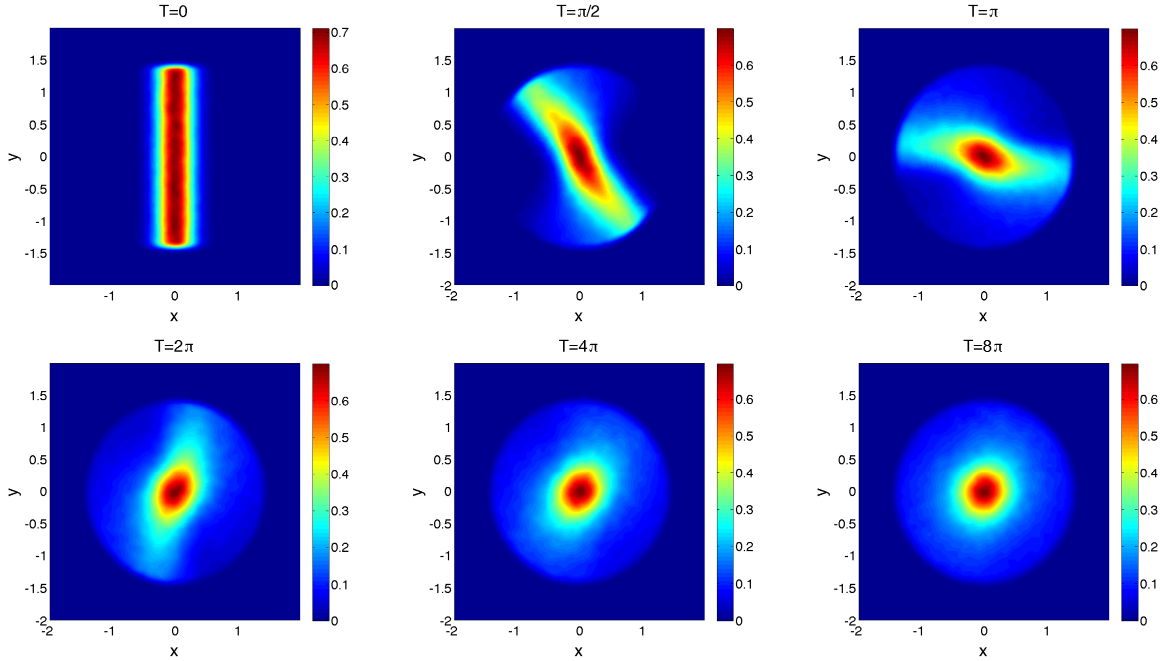

Figure 12: Contour plot of quantity $\chi^\varepsilon(t,\mathbf{x})$ in example (4.3) at different t with $\varepsilon = 0.01$.

[16] F. Filbet, E. Sonnendrücker, Modeling and numerical simulation of space charge dominated beams in the paraxial approximation, Math. Models Methods Appl. Sci. 16 (2005) pp. 763-791.

[17] 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 (2009) pp. 175-197.

[18] E. Frénod, S.A. Hirstoaga, M. Lutz, E. Sonnendrücker, Long time behavior of an exponential integrator for a Vlasov-Poisson system with strong magnetic field, Commun. in Comput. Phys. 18 (2015) pp. 263-296.

[19] E. Frénod, E. Sonnendrücker, Long time behavior of the two-dimensional Vlasov equation with a strong external magnetic field, Math. Models Methods Appl. Sci. 10 (2000) pp. 539-553.

[20] G.B. Jacobs, J.S. Hesthaven, Implicit-Explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning, Comput. Phys. Comm. 180 (2009) pp. 1760-1767.

[25] S. Jin, Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations, SIAM J. Sci. Comput. 21 (1999) pp. 441-454.

[21] M. Lemou, F. Méhats, X. Zhao, Uniformly accurate numerical schemes for the nonlinear Dirac equation in the nonrelativistic limit regime, preprint, to appear in Commun. Math. Sci. (2017) hal-01313976.

[22] A. Mouton, Two-scale semi-Lagrangian simulation of a charged particles beam in a periodic focusing channel, Kinet. Relat. Models 2 (2009) pp. 251-274.

[23] M. Shoucri, A two-level implicit scheme for the numerical solution of the linearized vorticity equation, Internat. J. Numer. Methods Engrg. 17 (1981) pp. 1525-1538.

[24] E. Sonnendrücker, Numerical Methods for Vlasov-Maxwell Equations, Springer, 2016.