Numerical methods for the two-dimensional Vlasov-Poisson equation in the finite Larmor radius approximation regime

Philippe Chartier,* Nicolas Crouseilles, and Xiaofei Zhao‡
(Dated: May 4, 2018)

Abstract

In this paper, we consider the numerical methods for solving the two-dimensional Vlasov-Poisson equation in the finite Larmor radius approximation regime. The model describes the behaviour of charged particles under a strong external magnetic field and the finite Larmor radius approximation. We discretise the equation under Particle-in-Cell method, where the characteristics equations are highly oscillatory system in the limit regime. We apply popular numerical integrators including splitting methods, multi-revolution composition methods, two-scale formulation method and limit solver to integrate the characteristics. Dissuasions are made to highlight the strength and drawback of each method. Numerical experiments are done, and comparisons on the accuracy, efficiency and long-time behaviour of the methods are made, aiming to suggest the method with the best performance for the problem.

* Univ Rennes, INRIA, CNRS, IRMAR-UMR 6625, F-35000 Rennes, France; Philippe.Chartier@inria.fr
† Univ Rennes, INRIA, CNRS, IRMAR-UMR 6625, F-35000 Rennes, France; nicolas.crouseilles@inria.fr
‡ Univ Rennes, INRIA, CNRS, IRMAR-UMR 6625, F-35000 Rennes, France; zhxfnus@gmail.com
I. INTRODUCTION

In tokamak plasma dynamics, the charged particles are confined around the magnetic lines under a strong magnetic field. By assuming that the particle distribution fluctuates at a much larger scale in the parallel direction with respect to the magnetic field than at the Larmor circle length scale along the orthogonal directions, one gets the so-called finite Larmor radius approximation regime [4, 29, 32], which naturally applies in tokamak physics for magnetic confinement fusion [21, 36–38]. In this paper, we consider the two-dimensional Vlasov-Poisson equation in the finite Larmor radius approximation regime (dimensionless form) [29]:

\[
\partial_t f^\varepsilon(t, x, v) + \frac{v}{\varepsilon} \cdot \nabla_x f^\varepsilon(t, x, v) + \left( \mathbf{E}^\varepsilon(t, x) + \frac{1}{\varepsilon} v^\perp \right) \cdot \nabla_v f^\varepsilon(t, x, v) = 0, \quad t > 0, \tag{I.1a}
\]

\[
E^\varepsilon(t, x) = -\nabla_x \phi^\varepsilon(t, x), \quad -\Delta \phi^\varepsilon(t, x) = \rho^\varepsilon(t, x) - n_i, \quad \rho^\varepsilon(t, x) := \int_{\mathbb{R}^2} f^\varepsilon(t, x, v) dv, \tag{I.1b}
\]

\[
f^\varepsilon(0, x, v) = f_0(x, v), \quad x, v \in \mathbb{R}^2, \tag{I.1c}
\]

where the space variable \( x = (x_1, x_2)^T \in \mathbb{R}^2 \), the velocity variable \( v = (v_1, v_2)^T \in \mathbb{R}^2 \), \( 0 < \varepsilon \leq 1 \) is a parameter denotes the ratio between the Larmor radius and the size of the physical domain, and notation \( v^\perp \) is defined hereafter as

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

\( n_i > 0 \) denotes the ion density of the background which ensures global quasi-neutrality.

This model has received widely mathematically addresses [4–6, 29]. When the parameter \( \varepsilon \) is small, the model contains fast oscillations in time which causes analytical and numerical difficulties. As \( \varepsilon \to 0 \), it has been shown in [4, 29] that \( f^\varepsilon(t, x, v) \) (in sense of two-scale [28]) converges to the function \( f(t, x + R(-\tau)v, e^{-J\tau v}) \) where \( f(t, x, v) \) solves

\[
\partial_t f(t, x, v) + \frac{1}{2\pi} \left( \int_0^{2\pi} R(-\tau)E_f(t, \tau, x + R(\tau)v) d\tau \right) \cdot \nabla_x f(t, x, v)
+ \frac{1}{2\pi} \left( \int_0^{2\pi} e^{-J\tau}E_f(t, \tau, x + R(\tau)v) d\tau \right) \cdot \nabla_v f(t, x, v) = 0, \tag{I.2a}
\]

\[
\nabla_x \cdot E_f(t, \tau, x) = \int_{\mathbb{R}^2} f(t, x + R(-\tau)v, e^{-J\tau v}) dv - n_i, \tag{I.2b}
\]

\[
f(0, x, v) = f_0(x, v), \tag{I.2c}
\]
with matrix
\[ R(\tau) := \begin{pmatrix}
\sin(\tau) & 1 - \cos(\tau) \\
\cos(\tau) - 1 & \sin(\tau)
\end{pmatrix}. \]

Our work is to for the first time apply popular numerical integrators under the Particle-In-Cell framework to integrate the highly oscillatory characteristics equations in the finite Larmor radius approximation scaling. The numerical integrators of interests here range from the classical schemes to some recently developed multiscale schemes. Our aims are to understand the limitations of the classical methods for solving the problem, and to explore and compare the efficiency of the multiscale methods. For the classical methods, we include the time splitting schemes and the limit solver on the averaged equation (I.2) (see [23, 31, 40]). The time splitting scheme is popular since it is very efficient in the classical scaling and easy to implement. The limit solver is to put a finite difference integrator on the averaged equation (I.2). For multiscale schemes, we focus on the recently developed multi-revolution composition (MRC) method [10, 12] and the two-scale formulation (TSF) method [11, 15]. TSF is to separate the fast time scale out from the solution and solve an augmented equation. It has been devoted to solve the Vlasov-Poisson equation in strong magnetic field regime [16, 17], where the uniform accuracy of TSF in terms of computational cost for all \(0 < \varepsilon \leq 1\) shows superb efficiency over others [18]. MRC is proposed in spirit of the Heterogeneous multiscale method [2] which consists of a macro solver and a micro solver. It is originally considered for solving oscillatory system [10, 12] under the case \(T/\varepsilon \gg 1\) with \(T > 0\) is the final time. We shall show in this paper that MRC is well defined for all \(T > 0\) and \(0 < \varepsilon \leq 1\), and particularly it transits naturally into splitting schemes when step size is small enough, which gives it an overall uniform accuracy. Both MRC and TSF methods are shown to give uniform accuracy in terms of the oscillation. Through systematical comparisons, their efficiency in solving the Vlasov-Poisson equation (I.1) in the finite Larmor radius regime shall be understood. In particular, it is the first time to perform the MRC method to the kinetic models. There are many other multiscale approaches in the literature, such as the stroboscopic average [7, 8, 13], the exponential time difference [35] and the semi-implicit Runge-Kutta [24, 25] in spirit of asymptotic preserving [43]. For their comparisons with the TSF method, we refer to the work [18].

The rest of the paper is organised as follows. In Section II, we go through the time splitting method, multi-revolution composition method, two-scale-formulation method and
limit solver in a sequence under the Particle-In-Cell discretisation. In Section III, numerical results are presented and detailed comparisons are made. Conclusions are summarised in Section IV.

II. NUMERICAL METHODS

In this section, we are going to present the numerical methods for solving the Vlasov-Poisson equation (I.1) under the framework of the Particle-in-Cell (PIC) discretisation. The PIC discretisation approximates the unknown distribution \( f^\varepsilon(t, x, v) \) of (I.1) by a sum of Dirac masses centred at \((x_k(t), v_k(t))\) with weight \( \omega_k > 0 \) for \( k = 1, \ldots, N_p \) and \( N_p \in \mathbb{N} \) as

\[
 f^\varepsilon_p(t, x, v) = \sum_{k=1}^{N_p} \omega_k \delta(x - x_k(t)) \delta(v - v_k(t)), \quad t \geq 0, \; x, v \in \mathbb{R}^2. \tag{II.1}
\]

The weight \( w_k \) and initial values of the particles \( x_{k,0}, v_{k,0} \) for \( k = 0, \ldots, N_p \) are prescribed according to the given initial distribution \( f_0(x, v) \) in (I.1c). To determine the weight, by integrating (II.1) at \( t = 0 \) in whole space we require

\[
 \sum_{k=1}^{N_p} \omega_k = \int_{\mathbb{R}^2 \times \mathbb{R}^2} f_0(x, v) dxdv.
\]

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

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

The initial data \( x_{k,0}, v_{k,0} \) for \( k = 1, \ldots, N_p \) 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 \( 1/\sqrt{N_p} \) [42]. When \( f_0 \) is of variable separation form, i.e. \( f_0(x, v) = \chi_1(x)\chi_2(v) \), especially for the cold plasma case as widely considered in the literatures where \( \chi_1 \) is a one-dimensional distribution and \( \chi_2 \) 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 [3]. We omit the details here for brevity.

The characteristics equations for particles (II.5) are coupled to the Poisson equation (I.1b) through the electric field \( E^\varepsilon \). Once the positions \( \{x_k(t)\}_{k=1,\ldots,N_p} \) of the particles are obtained
at time $t > 0$, one needs to evaluate the approximated density

$$\rho_p^\varepsilon(t, \mathbf{x}) = \sum_{l=1}^{N_p} \omega_l \delta(\mathbf{x} - \mathbf{x}_l(t)) \approx \rho^\varepsilon(t, \mathbf{x}) \quad \mathbf{x} \in \mathbb{R}^2,$$  \hspace{1cm} (II.2)

then solve the Poisson equation

$$-\Delta_x \phi^\varepsilon_p(t, \mathbf{x}) = \rho^\varepsilon_p(t, \mathbf{x}) - n_i^\varepsilon, \quad \mathbf{x} \in \mathbb{R}^2, \quad (\text{II.3})$$

on a mesh grid of $\mathbf{x}$ in $\mathbb{R}^2$ for $E_p^\varepsilon(t, \mathbf{x}) \approx E^\varepsilon(t, \mathbf{x})$, and finally interpolate for values of $E^\varepsilon(t, \mathbf{x}_k(t))$ at each particle position. In pratice, the Dirac delta function $\delta(\mathbf{x})$ is approximated by some regularized basis $S(x)$. For example in 1D, it is approximated by the B-spline function $S_m(x)$ of order $m \in \mathbb{N}_{[42]}$:

$$S^0(x) := \begin{cases} \frac{1}{\Delta x}, & |x| \leq \frac{\Delta x}{2}, \\ 0, & \text{else}, \end{cases} \quad S^m(x) := \frac{1}{\Delta x} \int_{x - \frac{\Delta x}{2}}^{x + \frac{\Delta x}{2}} S^{m-1}(y) dy, \quad m \geq 1. \quad (\text{II.4})$$

The case in two dimensions is done by tensor product. The B-spline function $S^m(x)$ is defined locally in space which is preferred from numerical point of view, but globally it is only a $C^{m-1}$ function for $m \geq 1$. A smooth but global basis has been considered in [33] to get high order accuracy.

Under the PIC discretisation (II.1), the characteristic equation of the Vlasov-Poisson model (I.1) reads for $k = 1, \ldots, N_p$

$$\dot{x}_k(t) = \frac{v_k(t)}{\varepsilon}, \quad \text{II.5a}$$

$$\dot{v}_k(t) = \frac{E^\varepsilon(t, x_k(t)) + v_k(t)}{\varepsilon}, \quad t > 0, \quad \text{II.5b}$$

$$x_k(0) = x_{k,0}, \quad v_k(0) = v_{k,0}. \quad \text{II.5c}$$

A standard numerical method to integrate (II.5) would be the time splitting methods. The system (II.5) can split into two subflows:

$$\mathcal{E}^\varepsilon_1(t) : \begin{cases} \dot{x}_k(t) = \frac{v_k(t)}{\varepsilon}, & t > 0, \\ \dot{v}_k(t) = \frac{Jv_k(t)}{\varepsilon}, \end{cases} \quad \text{and} \quad \mathcal{E}^\varepsilon_2(t) : \begin{cases} \dot{x}_k(t) = 0, & t > 0, \\ \dot{v}_k(t) = E^\varepsilon(t, x_k(t)). \end{cases} \quad (\text{II.6})$$

Note that under the PIC discretisation (II.2) and (II.3), the electric field $E^\varepsilon$ is approximated by $E^\varepsilon_p$:

$$-\nabla_x \cdot E^\varepsilon_p(t, \mathbf{x}) = \sum_{l=1}^{N_p} \omega_l \delta(\mathbf{x} - \mathbf{x}_l(t)) - n_i. \quad (\text{II.7})$$
By collecting of all the particles, i.e.

$$x^p(t) := (x_1(t), \ldots, x_{N_p}(t)),$$

we can re-define $E^ε_p(t, x)$ through (II.7) as

$$\Xi[x^p(t)](x) = E^ε_p(t, x), \quad \text{(II.8)}$$

where the time dependance is only through particles. Therefore, both of the subflows $E^ε_1(s)$ and $E^ε_2(s)$ from the splitting can be exactly integrated:

$$E^ε_1(t) : x_k(t) = x_k(0) - J(e^{Jt/ε} - I)v_k(0), \quad v_k(t) = e^{Jt/ε}v_k(0);$$

$$E^ε_2(t) : x_k(t) = x_k(0), \quad v_k(t) = v_k(0) + t\Xi[x^p(0)](x_k(0)).$$

since the electric field in $E^ε_2$ does not change in time. Choose a time step $\Delta t > 0$ and denote $M = T/\Delta t$ as the number of time grids on a time interval $[0, T]$. The standard second order Strang splitting for approximating characteristics flow (II.5) reads

$$\left((E^ε_2(\Delta t/2)E^ε_1(\Delta t))E^ε_2(\Delta t/2)\right)^M, \quad \text{(II.9)}$$

and the fourth order Yoshida splitting [44] reads:

$$\left((E^ε_2(c_1\Delta t/2)E^ε_1(c_1\Delta t))E^ε_2(c_2\Delta t)E^ε_1(c_0\Delta t)E^ε_2(c_2\Delta t)E^ε_1(c_1\Delta t))E^ε_2(c_1\Delta t/2)\right)^M, \quad \text{(II.10)}$$

with

$$c_0 = -\frac{2^{1/3}}{2 - 2^{1/3}}, \quad c_1 = \frac{1}{2 - 2^{1/3}}, \quad c_2 = \frac{c_0 + c_1}{2}. \quad \text{(II.11)}$$

By investigating the commutator, the global error of the $p$-order splitting method is

$$O(\Delta t^p/ε^p). \quad \text{(II.12)}$$

This error bound tells the time step restriction of splitting method as $\Delta t = O(ε)$. 

A. Multi-revolution composition (MRC) method

Among other kinds of multiscale methods such as stroboscopic averaging method [7, 8, 13] and exponential time differencing method [35], the multi-revolution composition (MRC)
method is known for its efficiency and geometric property. As has been proposed in [10, 12], the MRC method applies to oscillatory problems of the form:

\[
\begin{aligned}
\frac{d}{dt} y(t) &= Ay(t) + \varepsilon B(y(t)), \quad 0 < t \leq \frac{T}{\varepsilon}, \\
y(0) &= y_0,
\end{aligned}
\]  

(II.13)

where \( y(t) \) belongs to a Banach space \( X \), and \( A, B \) are two time-independent operators on \( X \). In particular, \( A \) is assumed to be a linear operator on \( X \) whose propagator \( E_A(t) = e^{At} \) is periodic in time \( t \) with a period \( T_0 \), and \( B \) could be a nonlinear operator in general on \( X \). If \( T/\varepsilon = O(1) \), (II.13) is not in the category of highly oscillatory problem, and it can be solved by standard numerical methods with computational costs independent of \( \varepsilon \). For simplicity of illustrations, we shall assume that \( T/\varepsilon \) is a multiple of the period \( T_0 \), otherwise one can split the time as \( T/\varepsilon = \lfloor T/\varepsilon T_0 \rfloor T_0 + T_r \) where \( T_r = O(1) \).

The MRC method begins by choosing an integer \( M_0 \geq 1 \) and denote

\[
H := \varepsilon M_0, \quad M = \frac{T}{HT_0},
\]

where \( H = \Delta t > 0 \) is interpreted as a macro time step and \( M \) is the number of the time grids. Denoting \( y^n \approx y(nM_0 T_0) \) for \( 1 \leq n \leq M \) and choosing \( y^0 = y(0) \), the MRC works as

\[
y^{n+1} = \prod_{j=1}^r (E_{A-\beta_j H B}(-T_0)) E_{A+\alpha_j H B}(T_0) y^n, \quad 0 \leq n \leq M - 1,
\]

where the real-valued coefficients \( (\alpha_j, \beta_j)_j \) are chosen parameters that determine the order of the MRC. The propagator \( E_{A+\alpha_j H B}(T_0) \) for \( j = 1, \ldots, r \) denotes the flow of

\[
\frac{d}{dt} \tilde{y}(t) = A\tilde{y}(t) + \alpha_j H B(\tilde{y}(t)), \quad 0 < t \leq T_0,
\]  

(II.14)

and \( E_{A-\beta_j H B}(-T_0) \) denotes the flow of

\[
\frac{d}{dt} \tilde{y}(t) = A\tilde{y}(t) - \beta_j H B(\tilde{y}(t)), \quad -T_0 \leq t < 0.
\]  

(II.15)

At each macro time level \( n \), MRC needs to evaluate (II.14) and (II.15) as two micro problems for \( r \) times. Since the micro problems (II.14) and (II.15) are imposed on \( O(1) \)-length time intervals, so as said before they can be solved by any standard discretisation. It is proved in [10] that if one applies a \( p \)-th order solver such as splitting method to solve the two micro problems (II.14) and (II.15) with a step size \( h > 0 \), provided that the solution of (II.13) is sufficiently smooth, the approximation error of the \( P \)-th order MRC method reads

\[
\|y^M - y(T/\varepsilon)\|_X \leq C(H^P + h^P).
\]  

(II.16)
When the propagator of the linear part is isometric, some improved convergence results are reported [12]. The second order MRC (MRC2) is typically using
\[ P = 2 : \ r = 1, \ \alpha_1 = \frac{1}{2}(1 + 1/M_0), \ \beta_1 = \frac{1}{2}(1 - 1/M_0), \]
and the fourth order MRC (MRC4) is taking
\[ P = 4 : \ r = 3, \ \alpha_1 = \alpha_3 = \frac{1}{M_0} \left( \frac{c}{12} + \frac{M_0^2}{3c} + \frac{M_0}{3} + \frac{1}{2} \right), \ \alpha_2 = -\frac{c^2 + (M_0 - 3)c + 4M_0^2}{6cM_0}, \]
\[ \beta_1 = \beta_3 = \frac{1}{M_0} \left( \frac{c}{12} + \frac{M_0^2}{3c} + \frac{M_0}{3} - \frac{1}{2} \right), \ \beta_2 = -\frac{c^2 + (M_0 + 3)c + 4M_0^2}{6cM_0}, \]
\[ c = \left( 10M_0^2 - 18M_0 + 6\sqrt{M_0^6 - 10M_0^4 + 9M_0^2} \right)^{1/3}. \]

MRC2 is literally well-defined for \( M_0 \geq 1 \). As for MRC4, we note by the above coefficient formula that \( c \) becomes complex-valued when \( M_0 = 2 \). Therefore, MRC4 is only defined for \( M_0 \geq 3 \).

Based on the observation of the error bound (II.16), we see that the total error of MRC consists of the macro-approximation error and the micro-discretisation error. If one chooses the MRC and the micro-solver of the same order, i.e. \( P = p \), and \( h = O(H) \), then the error from the macro part and the micro part contributes equally, which is \( O(H^P) \). Hence, to get the optimal efficiency of a chosen \( P \)-th order MRC in practical computations, there is no need to call for a high order micro-solver with \( p > P \) or to use a small step size \( 0 < h \ll H \) which will usually be more costly. We shall take this strategy in our study.

Though the MRC method is \( P \)-th order accurate in terms of the macro-step \( H \), as shown by the error bound (II.16). However, due to the existence of the micro-solver at each time level, the total computational cost does not linearly depend on \( 1/H \). We give the following convergence of the error in terms of total computational cost.

**Proposition II.1.** For the \( P \)-th order MRC with a \( p = P \)-th order micro-solver and \( h = O(H) \) to solve the model problem (II.13), if we denote \( N > 0 \) as the total computational cost of the scheme, the error reads:
\[ \|y^M - y(T/\varepsilon)\|_X \leq C \left( \frac{rT}{N} \right)^{P/2}. \]  

**Proof.** By the error bound (II.16) which is proved in [12] and the setup \( p = P, h = O(H) \), we have
\[ \|y^M - y(T/\varepsilon)\|_X \leq CH^P. \]
The computational cost of the micro-solver to the problems (II.14) and (II.15) is \( O(rT_0/h) = O(rT_0/H) \). Since \( M \) is the total number of the macro steps, so the total computational cost of the MRC scheme is

\[
N = O(rT_0M/H) = O(rT/H^2),
\]

by the relation \( M = T/(HT_0) \). Hence we have

\[
H = C \left( \frac{rT}{N} \right)^{1/2},
\]

and the error then can be written as

\[
\|y^M - y(T/\varepsilon)\|_X \leq C \left( \frac{rT}{N} \right)^{P/2}.
\]

\[
\square
\]

The result (II.19) shows that a \( P \)-th order MRC is indeed a \( P/2 \)-th order accurate numerical method in terms of the computational cost.

Now, let us check if the characteristic equation (II.5) fits into the model problem (II.13). Firstly, noting (II.7) and (II.8), we see that the characteristic equations (II.5) for \( k = 1, \ldots, N_p \) as a whole is an autonomous system. By a time scaling

\[
\tilde{x}_k(t) = x_k(\varepsilon t), \quad \tilde{v}_k(t) = v_k(\varepsilon t), \quad \tilde{x}^p(t) = x^p(\varepsilon t),
\]

to solve the Vlasov-Poisson equation (I.1) till a fixed \( T > 0 \), we turn to solve

\[
\begin{align*}
\dot{x}_k(t) &= v_k(t), \\
\dot{v}_k(t) &= Jv_k(t) + \varepsilon \Xi[x^p(t)](x_k(t)), & 0 < t \leq \frac{T}{\varepsilon}, \\
x_k(0) &= x_{k,0}, \quad v_k(0) = v_{k,0}, \quad k = 1, \ldots, N_p,
\end{align*}
\]

where we removed all the \( \sim \) for simplicity. Secondly, we check the linear part of (II.20):

\[
\begin{align*}
\dot{x}(t) &= v(t), \\
\dot{v}(t) &= Jv(t).
\end{align*}
\]

The flow is exactly given by

\[
\begin{align*}
x(t) &= x(0) - J(e^{It} - I)v(0), \\
v(t) &= e^{It}v(0),
\end{align*}
\]

with \( I \) the two by two identity matrix and

\[
\begin{pmatrix}
\cos(t) & \sin(t) \\
-\sin(t) & \cos(t)
\end{pmatrix}.
\]
Clearly, this propagator of the linear part is periodic in $t$ with period $T_0 = 2\pi$ (but not isometric), and therefore we are ready to apply the MRC method on formulation (II.20).

Suppose that we are using a $P$-th order MRC with coefficients $(\alpha_j, \beta_j)_{j=1}^r$ and we denote $x^n_k \approx x_k(nM_0T_0)$, $v^n_k \approx v_k(nM_0T_0)$ for $n = 0, \ldots, M$, with the chosen integer $M_0 \geq 1$ and $M = T/(HT_0)$, $H = \varepsilon M_0$. The scheme proceeds with $x^0_k = x_{k,0}$, $v^0_k = v_{k,0}$ and

$$
\begin{pmatrix}
  x_{k+1}^n \\
  v_{k+1}^n
\end{pmatrix} = \prod_{j=1}^r \left( \mathcal{E}_{A-\beta_jHB}(-T_0)\mathcal{E}_{A+\alpha_jHB}(T_0) \right) \begin{pmatrix}
  x^n_k \\
  v^n_k
\end{pmatrix}, \quad 0 \leq n \leq M - 1,
$$

(II.21)

where the propagator $\mathcal{E}_{A+\alpha_jHB}(T_0)$ for $j = 1, \ldots, r$ corresponds to the flow of

$$
\dot{x}_k(t) = v_k(t),
$$

(II.22a)

$$
\dot{v}_k(t) = Jv_k(t) + \alpha_jH\Xi[x^p(t)](x_k(t)), \quad 0 < t \leq T_0,
$$

(II.22b)

and $\mathcal{E}_{A-\beta_jHB}(-T_0)$ corresponds to the flow of

$$
\dot{x}_k(t) = v_k(t),
$$

(II.23a)

$$
\dot{v}_k(t) = Jv_k(t) - \beta_jH\Xi[x^p(t)](x_k(t)), \quad -T_0 \leq t < 0.
$$

(II.23b)

For the discretisation of the two subproblems (II.22) and (II.23), we can use the splitting methods same as directly on (II.5). For example, we split the system (II.22) as

$$
\mathcal{E}_1(t) : \begin{cases}
\dot{x}_k(t) = v_k(t), \\
\dot{v}_k(t) = Jv_k(t),
\end{cases}
$$

and

$$
\mathcal{E}_2(t) : \begin{cases}
\dot{x}_k(t) = 0, \\
\dot{v}_k(t) = \alpha_jH\Xi[x^p(t)](x_k(t)),
\end{cases}
$$

(II.24)

and $\mathcal{E}_1(t)$ and $\mathcal{E}_2(t)$ can be exactly integrated:

$$
\begin{align*}
\mathcal{E}_1(t) : x_k(t) &= x_k(0) - Je^Jt v_k(0), \quad v_k(t) = e^{Jt}v_k(0); \\
\mathcal{E}_2(t) : x_k(t) &= x_k(0), \quad v_k(t) = v_k(0) + t\alpha_jH\Xi[x^p(0)](x_k(0)).
\end{align*}
$$

Choose a micro time step $h > 0$ and denote $N_m = T_0/h$ as the number of grids for the subproblem (II.22) or (II.23). Then for the 2nd order MRC, i.e. $P = 2$, we approximate the propagator $\mathcal{E}_{A+\alpha_jHB}(T_0)$ by the second order Strang splitting

$$
\mathcal{E}_{A+\alpha_jHB}(T_0) \approx (\mathcal{E}_2(h/2)\mathcal{E}_1(h)\mathcal{E}_2(h/2))^{N_m}.
$$

The same method applies for (II.23). As for a 4th order MRC, i.e. $P = 4$, we turn to use the fourth order Yoshida splitting to approximate the propagator $\mathcal{E}_{A+\alpha_jHB}(T_0)$:

$$
\mathcal{E}_{A+\alpha_jHB}(T_0) \approx (\mathcal{E}_2(c_1h/2)\mathcal{E}_1(c_1h)\mathcal{E}_2(c_2h)\mathcal{E}_1(c_0h)\mathcal{E}_2(c_2h)\mathcal{E}_1(c_1h)\mathcal{E}_2(c_1h/2))^{N_m}.
$$
with \(c_0, c_1, c_2\) given in (II.11).

Based on the accuracy and efficiency discussion, we assume \(h = O(H)\) for the MRC method. To do so, we consider a constant ratio \(c_R > 0\) between the macro steps \(M\) and the micro steps \(N_m\),

\[
c_R := \frac{N_m}{M}.
\]

For fixed \(T\) and \(\varepsilon\), a MRC scheme is then determined by the computational parameter \(M > 0\). As mentioned, the MRC2 scheme under a chosen \(M\), is originally defined only for \(M_0\) being a positive integer, i.e.

\[
M_0 = \frac{T}{\varepsilon T_0 M} \geq 1.
\]

By increasing \(M\) such that \(M_0\) achieves the smallest integer value, i.e. \(M_0 = 1\), the error of the MRC2 would be stuck by the macro-part error in (II.16) as

\[
\text{error} = O(\varepsilon^p + h^p).
\]

Simply refining \(h\) will not give convergence for a fixed \(\varepsilon\). To make the MRC2 scheme well-defined for arbitrarily large \(M\) or saying to make the scheme converge as \(M \to \infty\), we assume that MRC2 becomes the splitting scheme when \(\frac{T}{\varepsilon T_0 M} < 1\). In fact, for the critical \(M_d = \frac{T}{\varepsilon T_0}\) such that \(M_0 = \frac{T}{\varepsilon T_0 M_d} = 1\), we have in MRC2 the coefficients \(\alpha_1 = 1, \beta_1 = 0\). Since \(E_A(T_0)\) is identity, so MRC2 under \(M = M_d\) is equivalent to apply the micro-solver which as we choose is the Strang splitting scheme, directly on

\[
\dot{x}_k(t) = v_k(t), \quad \text{(II.25a)}
\]

\[
\dot{v}_k(t) = Jv_k(t) + \varepsilon \Xi[x^p(t)](x_k(t)), \quad 0 < t \leq \frac{T}{\varepsilon}, \quad \text{(II.25b)}
\]

with step size \(\Delta t = h = T_0/N_m\). This motivated us to consider the extended 2nd order MRC method globally defined for arbitrary parameter \(M \geq 1\) chosen by users for solving the Vlasov-Poisson equation (I.1):

\[
\text{MRC2} := \begin{cases} 
\text{MRC scheme (II.21) with (II.17)}, & \text{when } M \leq M_d, \\
\text{Strang splitting (II.9) with } \Delta t = \frac{T}{c_R M_d M}, & \text{when } M > M_d,
\end{cases}
\]

with \(M_d := \frac{T}{\varepsilon T_0}\).

Similarly for MRC4, if \(M > 0\) is chosen such that \(\frac{T}{\varepsilon T_0 M} < 3\), we assume the method turns to the Yoshida splitting scheme (II.10) on (II.5). The extended 4th order MRC method with
arbitrary given parameter $M \geq 1$ reads:

$$
\text{MRC4} := \begin{cases} 
\text{MRC scheme (II.21) with (II.18),} & \text{when } M \leq M_d, \\
\text{Yoshida splitting (II.10) with } \Delta t = \frac{T}{c_R M_d M}, & \text{when } M > M_d,
\end{cases}
$$

with $M_d := \frac{T}{3c_T}$.

**Proposition II.2.** Denote error$(M)$ as the error function of the extended MRC scheme (MRC2 or MRC4) with respect to ambiguous $M \in \mathbb{R}^+$. With some ratio $c_R > 0$, then

$$\lim_{M \searrow M_d} \text{error}(M) = O \left( \lim_{M \nearrow M_d} \text{error}(M) \right). \quad \text{(II.26)}$$

**Proof.** For $M \leq M_d$, the error of the scheme is given by (II.16) where $H = \varepsilon M_0 = \frac{T}{T_0 M}$ and $h = \frac{T_0}{N_m} = \frac{T_0}{C_R M}$. Hence under $p = P$ ($= 2$ or 4), if one chooses $C_R = C T_0^2 / T$ for some $C > 0$, then

$$\lim_{M \searrow M_d} \text{error}(M) = O(h^p) = O \left( \frac{T_0}{C_R M_d} \right)^p.$$

For $M > M_d$, the extend MRC scheme is the splitting scheme on (II.5) with time step

$$\Delta t = \frac{T}{c_R M_d M} = O \left( \frac{\varepsilon T_0}{c_R M} \right),$$

where we used the definition of $M_d$. By the error of the splitting method (II.12), we have

$$\lim_{M \nearrow M_d} \text{error}(M) = O(\Delta t^p / \varepsilon^p) = O \left( \frac{T_0}{C_R M_d} \right)^p.$$

The above result means that two extended MRC methods have comparable error at the transition point of the scheme. In particular, MRC2 degenerates automatically to the Strang splitting method. After degeneration, the total computational cost of the MRC method is $O(M_d M)$. Hence, in terms of total cost, the extended MRC schemes actually have an order leap after the degeneration. The extended MRC method is still geometric which guarantees reliable long-time behaviour of the scheme.

**B. Two-scale formulation (TSF) method**

The general framework and mathematical foundations of the two-scale formulation method were given in [11]. It has already been devoted to solving the Vlasov-Poisson
equation with a strong magnetic field \[14, 17\], and its efficiency has been identified in \[18\] through comparisons other schemes. We are going to briefly describe this approach and show how it applies to the Vlasov-Poisson equation in the finite Larmor radius regime.

**Reformulation and two-scale model**

Introduce

\[ u_{k,+}(t) := x_k(t) + Jv_k(t), \quad u_{k,-}(t) := -Je^{-tJ/\varepsilon}v_k(t), \quad t \geq 0, \quad (\text{II.27}) \]

which indicates

\[ x_k(t) = u_{k,+}(t) + e^{tJ/\varepsilon}u_{k,-}(t), \quad v_k(t) = Je^{tJ/\varepsilon}u_{k,-}. \]

Then from (II.5) we get

\[
\begin{align*}
\dot{u}_{k,+}(t) &= JE^\varepsilon(t, u_{k,+}(t) + e^{tJ/\varepsilon}u_{k,-}(t)), \\
\dot{u}_{k,-}(t) &= -Je^{-tJ/\varepsilon}E^\varepsilon(t, u_{k,+}(t) + e^{tJ/\varepsilon}u_{k,-}(t)), \quad t > 0.
\end{align*}
\]

(II.28)

As we illustrated before, the electric field \( E^\varepsilon(t, x) \approx \Xi[x^p](x) \) is now given as:

\[
-\nabla \cdot \Xi[u_+(t) + e^{tJ/\varepsilon}u_-(t)](x) = \sum_{k=1}^{N_p} \omega_k \delta(x - (u_{k,+}(t) + e^{tJ/\varepsilon}u_{k,-}(t))) - n_i, \quad (\text{II.29})
\]

where \( u_+ := (u_{1,+}, \ldots, u_{N_p,+}) \in \mathbb{R}^{2 \times N_p} \). Hence, (II.28) as a system of size \( 2 \times N_p \) can be written as

\[
\dot{u}_\pm(t) = F_\pm(t/\varepsilon, u_+(t), u_-(t)),
\]

where \( F_\pm := (F_{1,\pm}, \ldots, F_{N_p,\pm}) \in \mathbb{R}^{2 \times N_p} \) and

\[
\begin{align*}
F_{k,+}(s, u_+, u_-) &= J\Xi[u_+ + e^{sJ}u_-](u_{k,+} + e^{sJ}u_{k,-}), \\
F_{k,-}(s, u_+, u_-) &= -Je^{-sJ}\Xi[u_+ + e^{sJ}u_-](u_{k,+} + e^{sJ}u_{k,-}).
\end{align*}
\]

(II.30)

Denoting the initial data as \( x_0 = (x_{1,0}, \ldots, x_{N_p,0}) \in \mathbb{R}^{2 \times N_p} \) and \( y_0 = (v_{1,0}, \ldots, v_{N_p,0}) \in \mathbb{R}^{2 \times N_p} \), we summarise the reformulated characteristics as

\[
\begin{align*}
\dot{u}_\pm(t) &= F_\pm(t/\varepsilon, u_+(t), u_-(t)), \quad 0 < t \leq T, \\
u_+(0) &= x_0 + Jv_0, \quad u_-(0) = -Jv_0.
\end{align*}
\]

(II.31)

We consider the following new two-scale formulation for (II.31) as

\[
\partial_t U_\pm(t, \tau) + \frac{1}{\varepsilon} \partial_\tau U_\pm(t, \tau) = F_\pm(\tau, U_+(t, \tau), U_-(t, \tau)), \quad t > 0, \quad \tau \in \mathbb{T},
\]

(II.32)
with initial data satisfying
\[ U_+(0, 0) = x(0) + Jv(0) =: u_+^0, \quad U_-(0, 0) = -Jv(0) =: u_-^0. \]

The augmented equation (II.32) recovers the solution of (II.5) via
\[ x(t) = U_+(t, t/\varepsilon) + e^{tJ/\varepsilon}U_-(t, t/\varepsilon), \quad v(t) = Je^{tJ/\varepsilon}U_-(t, t/\varepsilon). \]

Now following the general strategy in [11], we carry out the Chapman-Enskog expansion to get the well-prepared initial data to bound the time derivatives of \( U_\pm \). Denote \( Lu(\tau) = \partial_\tau u(\tau) \) for some periodic function \( u(\tau) \) on \( T \) and introduce the average operator \( \Pi \) as
\[ \Pi u = \frac{1}{2\pi} \int_0^{2\pi} u(s)ds. \]

Then \( L \) is invertible on the set of periodic functions with zero average, i.e. for \( u(\tau) \) with \( \Pi u = 0 \),
\[ L^{-1}u(\tau) = (I - \Pi) \int_0^\tau u(s)ds. \]

We denote the decomposition
\[ U_\pm(t, \tau) = \underline{U}_\pm(t) + h_\pm(t, \tau), \quad \text{with} \quad \underline{U}_\pm(t) = \Pi U_\pm(t, \tau), \]
then we have the equation for the macro part of the solution as
\[ \partial_t \underline{U}_\pm(t) = \Pi F_\pm(\tau, \underline{U}_+(t) + h_+(t, \tau), \underline{U}_-(t) + h_-(t, \tau)), \quad \text{(II.33)} \]
and the equation for the micro part as
\[ \partial_t h_\pm(t, \tau) + \frac{1}{\varepsilon} L h_\pm(t, \tau) = (I - \Pi) F_\pm(\tau, \underline{U}_+(t) + h_+(t, \tau), \underline{U}_-(t) + h_-(t, \tau)). \]

By inverting \( L \) on the micro part equation, we get
\[ h_\pm(t, \tau) = \varepsilon A F_\pm(\tau, \underline{U}_+(t) + h_+(t, \tau), \underline{U}_-(t) + h_-(t, \tau)) - \varepsilon L^{-1} \partial_t h_\pm(t, \tau), \]
where \( A := L^{-1}(I - \Pi) \). Formally, we see \( h_\pm = O(\varepsilon) \). As \( \varepsilon \to 0 \), the Chapman-Enskog expansion indicates the limit model to the leading order on the characteristics level as
\[ \begin{cases} \frac{d}{dt} \underline{U}_\pm(t) = \Pi F_\pm(\tau, \underline{U}_+(t), \underline{U}_-(t)), & \quad t > 0, \\ \underline{U}_\pm(0) = u_\pm^0. \end{cases} \quad \text{(II.34)} \]
Proposition II.3. The model (II.34) given by the Chapman-Enskog expansion on the characteristics, is equivalent to the limit model (I.2).

Proof. Defining

\[ g(t, U_+, U_-) := f^\varepsilon(t, x, v), \quad \text{under} \quad x = U_+ + e^{tJ/\varepsilon}U_-, \quad v = Je^{tJ/\varepsilon}U_-, \quad U_\pm \in \mathbb{R}^2, \]

then from (I.1a) we find for \( g = g(t, U_+, U_-) \),

\[ \partial_t g + JE_g(t, t/\varepsilon, U_+ + e^{tJ/\varepsilon}U_-) \cdot \nabla_+ g - Je^{-tJ/\varepsilon}E_g(t, t/\varepsilon, U_+ + e^{tJ/\varepsilon}U_-) \cdot \nabla_- g = 0, \quad t > 0, \]

(II.35a)

\[ \nabla \cdot E_g(t, t/\varepsilon, x) = \int_{\mathbb{R}^2} g(t, x + Jv, -Je^{-tJ/\varepsilon}v)dv - n_i, \quad x \in \mathbb{R}^2, \]

(II.35b)

\[ g(0, U_+, U_-) = f_0(U_+ + U_-, JU_-), \quad U_\pm \in \mathbb{R}^2, \]

(II.35c)

where \( \nabla \pm g \) denotes \( \nabla \nabla \pm g \). Noticing the definition of \( F_\pm \) in (II.30), it is clear to see that (II.34) is the average of the characteristic equation of (II.35) in \( \tau = t/\varepsilon \). Therefore, as formally shown by the Chapman-Enskog expansion, the average of (II.35) is the limit of (I.1) as \( \varepsilon \to 0 \). Introducing another change of variable on (II.35) as

\[ \tilde{g}(t, \tilde{x}, \tilde{v}) := g(t, U_+, U_-), \quad \text{under} \quad \tilde{x} = U_+ + U_- \quad \tilde{v} = JU_-, \quad \tilde{x}, \tilde{v} \in \mathbb{R}^2, \]

and noting the matrix \( R(\tau) = J - Je^{t\tau} \) for \( \tau \in \mathbb{R} \), it is directly to check that (II.35) becomes

\[ \partial_{\tilde{t}} \tilde{g} + R(-t/\varepsilon)E_g(t, t/\varepsilon, \tilde{x} + R(tJ/\varepsilon)\tilde{v}) \cdot \nabla \tilde{x} \tilde{g} + e^{-tJ/\varepsilon}E_g(t, t/\varepsilon, \tilde{x} + R(tJ/\varepsilon)\tilde{v}) \cdot \nabla \tilde{v} \tilde{g} = 0, \]

\[ \nabla \cdot E_g(t, t/\varepsilon, \tilde{x}) = \int_{\mathbb{R}^2} \tilde{g}(t, \tilde{x} + R(-t/\varepsilon)v, e^{-tJ/\varepsilon}v)dv - n_i, \]

\[ \tilde{g}(0, \tilde{x}, \tilde{v}) = g(0, \tilde{x} + J\tilde{v}, -J\tilde{v}) = f_0(\tilde{x}, \tilde{v}), \]

which after averaging in \( t/\varepsilon \) gives (I.2). \( \square \)

Remark II.1. A direct transform on (I.1) would be \( \tilde{g}(t, \tilde{x}, \tilde{v}) = f^\varepsilon(t, x, v) \) with \( \tilde{x} = x + R(-tJ/\varepsilon)v, \tilde{v} = e^{-tJ/\varepsilon}v \), and the Chapman-Enskog expansion also indicates the formal convergence from \( f^\varepsilon(t, x, v) \) to \( \Pi \tilde{g}(t, x + R(-\tau)v, e^{-\tau}v) \) as \( \varepsilon \to 0 \) (or in the sense of two-scale convergence [1, 28]).

Initial conditions
Let us come back and focus on the numerical approximation of (II.32). To construct a full initial data for (II.32) that could bound the time derivatives of $U_\pm$ uniformly for $0 < \varepsilon \leq 1$, one starts with

$$X^{0th}(\tau) := u^0_\pm + e^{\tau J} u^0_\mp.$$

Using $X^{0th}(\tau)$ through (II.29), define

$$-\nabla \cdot \Xi[X^{0th}(\tau)](x) = \sum_{l=1}^{N_\tau} \omega_l \delta(x - X^{0th}_l(\tau)) - n_i. \quad (II.36)$$

With $\Xi[X^{0th}(\tau)]$, one can define the ‘well-prepared’ initial data $U^{1st}_\pm(\tau)$ as

$$U^{1st}_\pm(\tau) := u^0_\pm + h^{0th}_\pm(\tau) - h^{0th}_\pm(0), \quad h^{0th}_\pm(\tau) := \varepsilon A F^{0th}_\pm(\tau, u^0_+, u^0_-), \quad (II.37)$$

where

$$F^{0th}_+(\tau, u^0_+, u^0_-) := J \Xi[X^{0th}(\tau)] (u^0_+ + e^{J \tau} u^0_-),$$

$$F^{0th}_-(\tau, U^{1st}_+, U^{1st}_-) := -J e^{-J \tau} \Xi[X^{0th}(\tau)] (u^0_+ + e^{J \tau} u^0_-).$$

For $t \geq 0$, the two-scale electric field is always computed from (II.29). As a property of using $U^{1st}_\pm(\tau)$ as the initial data for the two-scale problem (II.32), we have [11, 17]

$$\partial_t^k U_\pm(t, \tau) = O(1), \quad \partial_t^k F_\pm(\tau, U_+, U_-) = O(1), \quad k = 1, 2, \; 0 < \varepsilon \ll 1.$$

As a consequence, we have

$$\frac{d^k}{dt^k} F_\pm(\tau, U_+(t, \tau), U_-(t, \tau)) = O(1), \quad k = 1, 2, \; 0 < \varepsilon \ll 1. \quad (II.38)$$

This procedure to construct initial data can be extended to uniformly bound the higher order time derivatives. We refer the readers to [11, 17] for the theoretical supports of the procedure.

**Numerical discretisation**

Now we give the numerical discretisation of the characteristics. We shall integrate the two-scale equation (II.32) by an exponential integrator. Let $\Delta t > 0$ be the time step and denote $t_n = n \Delta t$ for $n \geq 0$. Discretizing the $\tau$-direction as $\tau_j = j \Delta \tau$, $j = 0, 1, \ldots, N_\tau$, with $\Delta \tau = 2\pi/N_\tau$ and $N_\tau$ some positive even integer, applying the Fourier transform to (II.32) on the torus $\mathbb{T}$, we get for $l = -N_\tau/2, \ldots, N_\tau/2 - 1$,

$$\frac{d}{dt} \widehat{(U_\pm)_l(t)} + \frac{il}{\varepsilon} \widehat{(U_\pm)_l(t)} = \widehat{(F_\pm)_l(t)}, \quad t > 0, \quad (II.39)$$
Integrating (II.39) from $t_n$ to $t_{n+1}$ ($n \geq 0$), we get

$$
\overline{U}_\pm(t_{n+1}) = e^{-\frac{\Delta t}{\tau}}(\overline{U}_\pm)(t_n) + \int_0^{\Delta t} e^{-\frac{\Delta t}{\tau}(s)}(\overline{F}_\pm)(t_n + s)\,ds.
$$

We approximate the above integration by the following quadrature for $n \geq 1$,

$$
\overline{U}_\pm(t_{n+1}) \approx e^{-\frac{\Delta t}{\tau}}(\overline{U}_\pm)(t_n) + \int_0^{\Delta t} e^{-\frac{\Delta t}{\tau}(s)}\left((\overline{F}_\pm)(t_n) + s\frac{d}{dt}(\overline{F}_\pm)(t_n)\right)\,ds
$$

$$
\approx e^{-\frac{\Delta t}{\tau}}(\overline{U}_\pm)(t_n) + p_l(\overline{F}_\pm)(t_n) + q_l \frac{1}{\Delta t} \left((\overline{F}_\pm)(t_n) - (\overline{F}_\pm)(t_{n-1})\right),
$$

where

\[
p_l := \int_0^{\Delta t} e^{-\frac{\Delta t}{\tau}(s)}\,ds = \begin{cases} 
\frac{i\varepsilon}{l} \left(e^{-\frac{i\Delta t}{\tau}} - 1\right), & l \neq 0, \\
\Delta t, & l = 0,
\end{cases}
\]

\[
q_l := \int_0^{\Delta t} e^{-\frac{\Delta t}{\tau}(s)}\,ds = \begin{cases} 
\frac{\varepsilon}{l^2} \left(e^{-\frac{i\Delta t}{\tau}} - i\varepsilon e^{-\frac{i\Delta t}{\tau}} - il\Delta t\right), & l \neq 0, \\
\frac{\Delta t^2}{2}, & l = 0.
\end{cases}
\]

As for the starting value $(\overline{U}_\pm)_l(t_1)$, we start with a first order approximation as

$$
(\overline{U}_\pm)_l(t_1) := e^{-\frac{i\Delta t}{\tau}}(\overline{U}_\pm)_l(0) + p_l(\overline{F}_\pm)_l(0),
$$

which gives a prediction of the nonlinearity at $t_1$ as

$$
\overline{F}_\pm(\tau, U_+^*(t_1, \tau), U_-^*(t_1, \tau)) = \sum_{l=-N/2}^{N/2-1} (\overline{F}_\pm)_l(t_1)e^{it\tau},
$$

then we do a correction as

$$
(\overline{U}_\pm)_l(t_1) \approx e^{-\frac{i\Delta t}{\tau}}(\overline{U}_\pm)_l(0) + p_l(\overline{F}_\pm)_l(0) + q_l \frac{1}{\Delta t} \left((\overline{F}_\pm)_l(t_1) - (\overline{F}_\pm)_l(0)\right).
$$

At each time level $t_n$ ($n \geq 0$), when the approximated $U_\pm(t_n, \tau)$ is obtained as above, we get

$$
X(t_n, \tau) = U_+(t_n, \tau) + e^{\tau J}U_-(t_n, \tau), \quad t \geq 0, \quad \tau \in \mathbb{T}.
$$

Then we can compute the two-scale electric field as

$$
\nabla_x \cdot \Xi[X(t_n, \tau)](x) = \sum_{l=1}^{N_p} \omega_l \delta(x - X_l(t_n, \tau)) - n^e_i,
$$

\[
\]
and the evaluation of nonlinearity \( F_{\pm} \) follows by definition.

In all, the detailed numerical scheme for integrating (II.32) 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

\[
\left( U_{\pm}^{-1} \right)_t = e^{-\frac{\Delta t}{\varepsilon} (U_{\pm}^-)_t} + p_t(F_{\pm})_t + q_t \frac{1}{\Delta t} \left( (F_{\pm})_{-1} - (F_{\pm})_0 \right), \tag{II.40a}
\]

\[
\left( U_{\pm}^{-n+1} \right)_t = e^{-\frac{\Delta t}{\varepsilon} (U_{\pm}^-)_t} + p_t(F_{\pm})_t + q_t \frac{1}{\Delta t} \left( (F_{\pm})_{n-1} - (F_{\pm})^-_n \right), \quad n \geq 1, \tag{II.40b}
\]

where for \( n \geq 0 \),

\[
U_{\pm}^n(\tau) = \sum_{l=-N/2}^{N/2} (\tilde{U}_{\pm}^l)^n e^{i\tau r}, \quad F_{\pm}^n(\tau) = \sum_{l=-N/2}^{N/2} (\tilde{F}_{\pm}^l)^n e^{i\tau r}, \quad F_{\pm}^{*1}(\tau) = \sum_{l=-N/2}^{N/2} (\tilde{F}_{\pm}^l)^{*1} e^{i\tau r},
\]

and

\[
F_{\pm}^n(\tau) = J\Xi[X^n(\tau)] \left( U_{\pm}^n(\tau) + e^{J\tau}U_{-\pm}^n(\tau) \right),
\]

\[
F_{\pm}^n(\tau) := -Je^{-J\tau}\Xi[X^n(\tau)] \left( U_{\pm}^n(\tau) + e^{J\tau}U_{-\pm}^n(\tau) \right), \quad n \geq 0,
\]

\[
F_{\pm}^{*1}(\tau) = J\Xi[X^{*1}(\tau)] \left( U_{\pm}^{*1}(\tau) + e^{J\tau}U_{-\pm}^{*1}(\tau) \right),
\]

\[
F_{\pm}^{*1}(\tau) := -Je^{-J\tau}\Xi[X^{*1}(\tau)] \left( U_{\pm}^{*1}(\tau) + e^{J\tau}U_{-\pm}^{*1}(\tau) \right),
\]

with

\[
\left( U_{\pm}^{*1} \right)_t = e^{-\frac{\Delta t}{\varepsilon} (U_{\pm}^-)_t} + p_t(F_{\pm})_t, \quad U_{\pm}^{*1}(\tau) = \sum_{l=-N/2}^{N/2} (\tilde{U}_{\pm}^l)^{*1} e^{i\tau r},
\]

and

\[
\nabla_\mathbf{x} \cdot \Xi[X^n(\tau)](\mathbf{x}) = \sum_{l=1}^{N} \omega_l S(\mathbf{x} - X^n_l(\tau)) - n_i, \quad X^n_i(\tau) = U_{\pm,l}^n(\tau) + e^{J\tau}U_{-\pm,l}^n(\tau),
\]

\[
\nabla_\mathbf{x} \cdot \Xi[X^{*1}(\tau)](\mathbf{x}) = \sum_{l=1}^{N} \omega_l S(\mathbf{x} - X^{*1}_l(\tau)) - n_i, \quad X^{*1}_i(\tau) = U_{\pm,l}^{*1}(\tau) + e^{J\tau}U_{-\pm,l}^{*1}(\tau),
\]

where \( S(\mathbf{x}) \) is the regularised basis as defined in (II.4).

With \( \{U_{\pm, k}^n\}_{k=1}^{N_p} \) from the exponential integrator (II.40), one can obtain the approximation to the solution of the original characteristics (II.5) as

\[
x_k(t_n) \approx U_{+, k}^n \left( \frac{t_n}{\varepsilon} \right) + e^{t_n, J/k} U_{-, k}^n \left( \frac{t_n}{\varepsilon} \right), \quad \mathbf{v}_k(t_n) \approx J e^{J/k} U_{+, k}^n \left( \frac{t_n}{\varepsilon} \right), \quad n \geq 1.
\]

Then together with the PIC approximation (II.1), we complete the numerical scheme of a two-scale formulation (TSF) PIC method for solving the Vlasov-Poisson equation (I.1). As has been proved in [15], the exponential integrator (II.40) for solving the two-scale formulation (II.31) enjoys the second order uniformly accurate property.
Proposition II.4. Let \( U_\pm(t, \tau) \) be the solution of (II.32) on \([0, T] \times [0, 2\pi]\) subject to the initial condition (II.37). Let \((U_\pm)^n(\cdot), n \geq 0\) be the numerical solution defined by the scheme (II.40) with the initial data (II.37). Then, the following estimate holds

\[
sup_{\varepsilon \in [0, 1]} \| U_\pm(t_n, \cdot) - U_\pm^n(\cdot) \|_{L^\infty} \leq C \Delta t^2,
\]

with \( C > 0 \) independent of \( \varepsilon \), for all \( n = 0, \ldots, N \) with \( N \Delta t = T \) is the final time.

Then it directly follows that the original particles’ positions and velocities are also approximated uniformly in \( \varepsilon \) with second accuracy.

Corollary II.1. Let \((U_\pm)^n(\cdot), n \geq 0\) defined by the numerical scheme (II.40) with the initial data (II.37). Then the numerical particles:

\[
\begin{align*}
  x_n^k := U_{+,k}^n \left( \frac{t_n}{\varepsilon} \right) + e^{t_n J/\varepsilon} U_{-,k}^n \left( \frac{t_n}{\varepsilon} \right), \\
  v_n^k := J e^{t_n J/\varepsilon} U_{-,k}^n \left( \frac{t_n}{\varepsilon} \right),
\end{align*}
\]

satisfy the following estimate

\[
\sup_{\varepsilon \in [0, 1]} |x_k(t_n) - x_k^n| + |v_k(t_n) - v_k^n| \leq C \Delta t^2, \quad k = 1, \ldots, N_p,
\]

with \( C > 0 \) independent of \( \varepsilon \), for all \( n = 0, \ldots, M \) with \( M \Delta t = T \) is the final time.

Hence, the exponential integrator (II.40) based on the two-scale formulation (II.32) with (II.37) is a second order uniformly accurate (UA) scheme for integrating the Vlasov-Poisson equation (I.1) for all \( \varepsilon \in [0, 1] \). In practice, the \( \tau \)-direction needs a discretisation. With an even integer \( N_\tau > 0 \), we can consider a uniform grid \( \tau_j = \frac{2\pi j}{N_\tau} \) \((j = 0, \ldots, N_\tau)\) on \([0, 2\pi]\) and use fast Fourier transform for the Fourier modes in the scheme (II.40). The solution is smooth and periodic in the \( \tau \)-direction, and therefore the discretisation error in \( \tau \) would be independent of \( \varepsilon \) and spectrally accurate under the optimal regularity. The computational cost of the two-scale method at each time level is \( O(N_\tau \log N_\tau) \).

C. Limit solver (LS)

Solving the limit equation instead of the original equation is a possible choice for to approximate the solution of the kinetic models when \( 0 < \varepsilon \ll 1 \). This approach is popular when the original equation lacks of efficient solvers, while the limit equation comparatively
is easy to discretise [27, 40]. The limit model (I.2) of the Vlasov-Poisson equation in the
finite Larmor radius limit could also be discretised using the PIC method. Let
\[ f(t, x, v) = \sum_{k=1}^{N_p} \omega_k \delta(x - x_k(t)) \delta(v - v_k(t)). \]
The characteristic equations read
\[
\dot{x}_k(t) = \frac{1}{2\pi} \left( \int_0^{2\pi} R(-\tau) E_f(t, \tau, x_k(t) + R(\tau)v_k(t)) d\tau \right), \tag{II.41a}
\]
\[
\dot{v}_k(t) = \frac{1}{2\pi} \left( \int_0^{2\pi} e^{-J\tau} E_f(t, \tau, x_k(t) + R(\tau)v_k(t)) d\tau \right), \quad t > 0, \tag{II.41b}
\]
\[ x_k(0) = x_{k,0}, \quad v_k(0) = v_{k,0}. \tag{II.41c} \]
The Poisson equation reads
\[
\nabla_x \cdot E_f(t, \tau, x) = \sum_{l=1}^{N_p} \int_{\mathbb{R}^2} \omega_l \delta(x + R(-\tau)v - x_l(t)) \delta(e^{-J\tau}v - v_l(t)) dv - n_i
\]
\[ = \sum_{l=1}^{N_p} \omega_l \delta(x + R(-\tau)e^{J\tau}v_l(t) - x_l(t)) - n_i. \]
The characteristic equations (II.41) contains no high frequencies and could be integrated
by a standard numerical integrator such as the leap-frog method:
\[
\begin{align*}
\dot{x}_k^1 &= x_k^0 + \Delta t F_1^0, & \dot{v}_k^1 &= v_k^0 + \Delta t F_2^0, \tag{II.42a} \\
\dot{x}_k^{n+1} &= x_k^{n+1} + 2\Delta t F_1^n, & \dot{v}_k^{n+1} &= v_k^{n+1} + 2\Delta t F_2^n, & n \geq 1, \tag{II.42b}
\end{align*}
\]
with
\[
F_1^n = \frac{1}{2\pi} \left( \int_0^{2\pi} R(-\tau) E_f^n(\tau, x_k^n + R(\tau)v_k^n) d\tau \right), \quad F_2^n = \frac{1}{2\pi} \left( \int_0^{2\pi} e^{-J\tau} E_f^n(\tau, x_k^n + R(\tau)v_k^n) d\tau \right),
\]
\[
\nabla_x \cdot E_f^n(\tau, x) = \sum_{l=1}^{N_p} \omega_l \delta(x + R(-\tau)e^{J\tau}v_l^n - x_l^n) - n_i, \quad n \geq 0.
\]
In practice, the \( \tau \) direction also needs a discretisation similarly as in the two-scale method.
Denoting still \( N_{\tau} \) as the number of the uniform grids on \([0, 2\pi]\], the averages in \( \tau \) can be
computed by a composite trapezoidal rule with \( N_{\tau} \) points of spectral accuracy in case of
optimal regularity. At each time level, the computational cost of the limit solver is \( O(N_{\tau}) \).

Scheme (II.42) is a second order accurate integrator for the limit equations (II.41). Meanwhile, it also contributes as a limit solver (LS) for the Vlasov-Poisson equation (I.1) in the
finite Larmor radius regime, which will offer accurate approximations to the solution as $\varepsilon \ll 1$. Formally, the error of the limit integrator for solving (I.1) is

$$\Delta t^2 + \Delta \tau^m + \varepsilon^p,$$

(II.43)

where $m > 0$ depends on the smoothness and $p > 0$ describes the convergence rate from the Vlasov-Poisson equation (I.1) to the limit model (I.2) in the finite Larmor radius regime. Theoretically, only the weak convergence in the limit has been established, but the optimal rate $p$ remains unknown in the literature. In the next section, we shall investigate it numerically where we observe $p = 1$.

### III. NUMERICAL RESULT

We present the numerical results of applying the two MRC methods and the two-scale formulation (TSF) method to solve the Vlasov-Poisson equation (I.1) for a wide range of $\varepsilon \in [0, 1]$. We shall firstly study the convergence of the methods and then make some comparisons in the computational efficiencies.

We take for the Vlasov-Poisson equation (I.1) a bi-Maxwellian initial data periodically perturbed in space:

$$f_0(x_1, x_2, v_1, v_2) = \frac{1}{4\pi} (1 + \sin(x_2) + \eta \cos(kx_1)) \left( e^{-\frac{(v_{1+1.5})^2+v_2^2}{2}} + e^{-\frac{(v_{1-1.5})^2+v_2^2}{2}} \right).$$

(III.1)

The computational domain for $x$ is $\Omega = [0, 2\pi/k] \times [0, 2\pi]$ for some $k, \eta > 0$. We choose $\eta = 0.05, k = 0.5$ and discretize $\Omega$ with 64 points in $x_1$-direction and 32 points in $x_2$-direction. The background constant $n_i = 1$ in this case. The Poisson equation (I.1b) is always solved on $\Omega$ with periodic boundary condition by the fast Fourier transform. We fix the basis function $S(x)$ for the PIC method as the fifth order B-spline, i.e. $m = 5$ in (II.4).

The initial position and velocity of each particle are generated by means of the rejection sampling. All the numerical methods are programmed in a sequential way in Fortran and are run on an Interl(R) Xeon(R) CPU 2.4GHz server. As a diagnostic, we consider the convergence regarding the density:

$$\rho_\varepsilon(t, x) = \int_{\mathbb{R}^2} f_\varepsilon(t, x, v) dv, \quad x \in \Omega,$$

and the kinetic energy density:

$$\rho_\varepsilon'(t, x) = \int_{\mathbb{R}^2} |v|^2 f_\varepsilon(t, x, v) dv, \quad x \in \Omega.$$
TABLE I: Error of Strang splitting and Yoshida splitting with respect to $M$ at $T = \pi/2(= M\Delta t)$: maximum norm of error in the kinetic energy density $\rho^\varepsilon_v$.

<table>
<thead>
<tr>
<th></th>
<th>$M = 2^2$</th>
<th>$M = 2^3$</th>
<th>$M = 2^4$</th>
<th>$M = 2^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strang</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon = 1/2^1$</td>
<td>3.52E-2</td>
<td>9.20E-3</td>
<td>2.00E-3</td>
<td>5.11E-4</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^2$</td>
<td>1.19E-1</td>
<td>7.70E-3</td>
<td>2.10E-3</td>
<td>4.25E-4</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^3$</td>
<td>1.88E-1</td>
<td>1.12E-3</td>
<td>6.50E-3</td>
<td>1.60E-3</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^4$</td>
<td>6.67E-1</td>
<td>1.83E-1</td>
<td>1.19E-1</td>
<td>6.10E-3</td>
</tr>
<tr>
<td>Yoshida</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon = 1/2^4$</td>
<td>5.10E-4</td>
<td>1.56E-5</td>
<td>8.96E-7</td>
<td>5.30E-8</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^2$</td>
<td>2.60E-3</td>
<td>2.51E-4</td>
<td>8.02E-6</td>
<td>4.37E-7</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^3$</td>
<td>9.90E-3</td>
<td>2.20E-3</td>
<td>2.50E-4</td>
<td>5.02E-6</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^4$</td>
<td>1.99E-1</td>
<td>9.10E-3</td>
<td>2.30E-3</td>
<td>1.94E-4</td>
</tr>
</tbody>
</table>

Firstly, we study the error of the time splitting method, aiming to understand the dependence of error on $\varepsilon$ and the meshing strategy. For the study, we take the number of particles $N_p = 102400$ and solve the Vlasov-Poisson equation (I.1) till $T = \pi/2$ by the Strang splitting (II.9) or Yoshida splitting (II.10) method for different $M (= T/\Delta t)$ and $\varepsilon$. The reference solution is obtained by using the 4th order Yoshida splitting method with time step $\Delta t \approx 2.99 \times 10^{-6}$. Table I shows the relative error of the Strang splitting and Yoshida splitting in $\rho^\varepsilon_v(t, x)$ under maximum norm. From the results, we see that the splitting method only converges when $\Delta t < \varepsilon$ (the underlined place in the table corresponds to $\Delta t = \varepsilon$). Outside the regime $\Delta t < \varepsilon$, there is no convergence. The error under a fixed $M$ increases as $\varepsilon$ decreases. The error remains at the same order under meshing strategy $\Delta t = O(\varepsilon)$. One remark is that here we did not observe the super-convergence of the splitting methods as on the nonlinear Schrödinger equation [9].

Secondly, we study the convergence of the extended MRC2 and MRC4 under the same particle setup. We compute the relative error in $\rho_v(t, x)$ and $\rho^\varepsilon_v(t, x)$ at $T = \pi/2$ and measure error in maximum norm in space. The error of MRC2 and MRC4 with respect to $M_0$ (number of periods) is tabulated in Table II which is a traditional way [12] to view the convergence the MRC methods as $M_0$ decreases (consequently $M$ increases). The MRC method converges
TABLE II: Error of MRC2 ($N_m = M$) and MRC4 ($N_m = 8M$) with respect to $M_0$ (before degeneration) at $T = \pi/2$: maximum error in the kinetic energy density $\rho_\varepsilon^\prime$.

<table>
<thead>
<tr>
<th></th>
<th>$M_0 = 8$</th>
<th>$M_0 = 4$</th>
<th>$M_0 = 2$</th>
<th>$M_0 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon = 1/2^7$</td>
<td>1.25E-1</td>
<td>5.70E-3</td>
<td>1.50E-3</td>
<td>1.07E-4</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^8$</td>
<td>5.70E-3</td>
<td>1.50E-3</td>
<td>1.10E-4</td>
<td>1.49E-6</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^9$</td>
<td>1.50E-3</td>
<td>1.12E-4</td>
<td>1.29E-6</td>
<td>2.04E-7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$M_0 = 32$</th>
<th>$M_0 = 16$</th>
<th>$M_0 = 8$</th>
<th>$M_0 = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon = 1/2^8$</td>
<td>2.30E-3</td>
<td>1.76E-4</td>
<td>1.25E-6</td>
<td>5.72E-9</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^9$</td>
<td>1.76E-4</td>
<td>1.22E-6</td>
<td>4.52E-9</td>
<td>2.51E-10</td>
</tr>
<tr>
<td>$\varepsilon = 1/2^{10}$</td>
<td>1.21E-6</td>
<td>4.27E-9</td>
<td>2.35E-10</td>
<td>2.19E-11</td>
</tr>
</tbody>
</table>

well from the results, and the MRC is consistent till $M_0$ reaches the smallest possible value before it degenerates. The convergence results indicate the successful application of the MRC methods on the Vlasov-Poisson model (I.1). To view the convergence of the MRC methods in terms of step size $\Delta t = H$, we plot the errors versus the number of macro time steps $M = T/(HT_0)$. Figure 1 shows the temporal error of the MRC2-PIC method with respect to $M = N_m$ (same number of micro grids) under different $\varepsilon$. The results illustrate the second order accuracy of the extended MRC2 and the convergence in terms of $M$ is uniform with respect to $0 < \varepsilon \leq 1$. Figure 2 shows the temporal error of the MRC4-PIC method with respect to $M = N_m/8$ (i.e. the ratio $C_R = 8$) under different $\varepsilon$, and Figure 3 shows the case with less micro grids, e.g. $N_m = M$ (i.e. $C_R = 1$). We see that the MRC4 method converges uniformly, but it fails to reach the optimal fourth order convergence rate if the ratio between the number of micro-grid points and macro-grid points is not chosen large enough. Under this practical guide, MRC4 is much more expensive than MRC2 in computational time.

Thirdly, we test the convergence of the TSF method. Figure 4 shows the temporal error of the TSF-PIC method with respect to $M = (T/\Delta t)$ under different $\varepsilon$. Figure 5 shows the error of the TSF-PIC method with respect to $N_\tau$. The numerical results illustrate the uniform convergence rate of the TSF method for all $\varepsilon \in [0, 1]$. In $\tau$-discretisation, we did not reach spectral rate since the $\delta$-function in the PIC method is approximated by quintic
splines which are $C^4$ functions. One interesting fact is that here we did not observe the super-convergence in the error as $\varepsilon \to 0$, i.e. the error with a fixed $\tau$ converges to zero as $\varepsilon \to 0$, which is unlike the application of the two-scale strategy in the strong-magnetic field regime of the Vlasov-Poisson equation case $[14, 17, 18]$. This is due to the $\tau$-variable appears in the macro part of the two-scale equation (II.33), which consequently also appears in the leading order limit equation (II.34). On the contrast, the leading order limit model of Vlasov-Poisson equation in the strong-magnetic field limit does not dependent on $\tau$ $[17, 30]$ and therefore the discretisation error in $\tau$ comes from the high order terms which will vanish as $\varepsilon \to 0$.

Now, we make some comparisons on the computational efficiency of MRC2, MRC4, TSF method and including the limit solver (LS). We list the error in $\rho^{\varepsilon}(t, x)$ and the corresponding computational time of each method under the same macro time step. The results are shown in Table III for $\varepsilon \in [0, 1/4^3]$. Figure 6 depicts the efficiency of each by showing the error versus computational time of the three convergent methods: MRC2, MRC4 and TSF method. From the results, we see that: i) LS works accurately and efficiently in the limit regime, but the error easily gets stuck in the intermediate regime (c.f. Table III) due to the gap between the two models. ii) MRC4 and TSF are very accurate, but they are costly. MRC4 is much more expensive that MRC2. The error of both MRC2 and MRC4 under a fixed $M$ is uniform in $\varepsilon$ (c.f. Table III). iii) To reach the same accuracy, MRC2 is the most efficient method (c.f. Figure 6).

Then, we test and compare the long-time performance of MRC2, MRC4 and TSF method. As a diagnostic, we consider the energy or Hamiltonian of the Vlasov-Poisson equation (I.1), which is a conserved quantity for all time, i.e.

$$H(t) := \frac{1}{2} \int_{\mathbb{R}^2} \int_{\Omega} |v|^2 f^{\varepsilon}(t, x, v) dv \, dx + \frac{\varepsilon}{2} \int_{\Omega} |E^{\varepsilon}(t, x)|^2 dx \equiv H(0), \quad t \geq 0. \quad (III.2)$$

We solve the Vlasov-Poisson equation (I.1) till $T = 32\pi$ with MRC2, MRC4 and TSF method under the same $M = 2^{7}(\Delta t \approx 0.78)$. We compute the numerical value of the energy $H(t)$ on the time grids. The relative numerical energy error $|H(t) - H(0)|/H(0)$ is shown in Figure 12 for several $\varepsilon$. By the results, we see that the TSF method has an increasing drift in the energy error which will eventually leads to large error in long time approximation. This illustrates that the approach is not geometric, though one could consider the restart strategy in $[17, 18]$ for improvement. On the contrast, the energy error of MRC method
remains at the same order in the computation and the error at long time is much smaller than that of TSF.

Based on overall observations from our numerical experiments, we suggest the MRC2 method for solving the Vlasov-Poisson equation (I.1) in the finite Larmor radius limit regime from the efficiency point of view, particularly for long-time dynamics.

At last but not least, we apply the schemes to do some numerical studies and simulations. Firstly, we want to identify the rate of convergence of \( f^\varepsilon \) to \( f \) as \( \varepsilon \) goes to zero, i.e. the \( p > 0 \) in (II.43). We accurately solve the limit model (I.2) by the LS method (II.42) and solve the Vlasov-Poisson equation (I.1) with the TSF method under small step size, e.g. \( \Delta t \approx 6.13 \times 10^{-3} \) and \( N_x = 64 \). We compute the relative difference between the quantity \( \rho^\varepsilon(t, x) \) and \( \rho^\varepsilon_\mu(t, x) \) given by the two models. Figure 8 shows the convergence of the difference (maximum norm) as \( \varepsilon \to 0 \). As the results shown, the convergence rate between the solution of the two models is one. Then we consider the time evolution of the first Fourier mode of the electric potential \( \phi^\varepsilon \) in semi-log scale, so we compute quantity:

\[
\xi(t) := \ln(|\hat{\phi}_{1,1}^\varepsilon|(t)), \quad t \geq 0.
\]  

(III.3)

We take \( \eta = 0.05 \) or \( \eta = 0.01 \) in (III.1) and solve the Vlasov-Poisson equation (I.1) by the MRC2 method with \( M = N_m = 32 \) (\( \Delta t \approx 0.196 \)). Figure 9 plots \( \xi(t) \) till \( T = 32\pi \). For the Vlasov-Poisson equation (I.1) with the given initial data (III.1) under \( \eta = 0.05 \), we show the long-time simulation of the solution given by the MRC2 method under \( M = 32 \). For a better visualisation quality, here we enlarge the number of particles \( N_p = 409600 \). Figure 10 and Figure 11 present respectively the 2D plots of \( \rho^\varepsilon(t, x) - 1 \) and \( \chi^\varepsilon(t, v) \):

\[
\chi^\varepsilon(t, v) := \int_\Omega f^\varepsilon(t, x, v)dx, \quad t \geq 0, \; v \in \mathbb{R}^2,
\]

for the dynamics of the solution in physical space and velocity space under \( \varepsilon = 1/4^3 \). The time evolution of the relative numerical fluctuation of the energy (III.2) during the computing is given in Figure 12.

IV. CONCLUSION

We applied numerical schemes under Particle-In-Cell discretisation to the Vlasov-Poisson equation in the finite Larmor radius regime. The methods include classical methods such
FIG. 1: Temporal error of MRC2-PIC ($N_m = M$) at $T = \pi/2$ with respect to number of macro steps $M$ and $\varepsilon$: maximum error in the density $\rho^\varepsilon$ (first row) and in the kinetic energy density $\rho_v^\varepsilon$ (second row).

as the time splitting scheme and limit solver, and the multiscale methods such as multi-revolution composition (MRC) method and the two-scale formulation (TSF) method. Limitations of the classical methods are reviewed and summarised. Comparisons between the two popular multiscale methods TSF and MRC, which are both claimed to be uniformly accurate in the literature, were carried out through numerical experiments. We found the TSF is not as efficient as it is for the strong magnetic field application, and MRC turns out to be more efficient and has a good long-time behaviour.

ACKNOWLEDGEMENTS

This work was supported by the French ANR project MOONRISE ANR-14-CE23-0007-01. N. Crouseilles is supported by the Enabling Research EUROFusion project CfP-WP14-ER-01/IPP-03. X. Zhao is partially supported by the IPL FRATRES and partially by the Deutsche Forschungsgemeinschaft (DFG) through CRC 1173. We thank Dr. Yong Zhang
FIG. 2: Temporal error of MRC4-PIC \((N_m = 8M)\) at \(T = \pi/2\) with respect to number of macro steps \(M\) and \(\varepsilon\): maximum error in the density \(\rho^\varepsilon\) (first row) and in the kinetic energy density \(\rho_v^\varepsilon\) (second row).

FIG. 3: Order reduction of MRC4-PIC with less micro grids \((N_m = M)\): temporal error at \(T = \pi/2\) with respect to number of macro steps \(M\) and \(\varepsilon\): maximum error in the density \(\rho^\varepsilon\) (first row) and in the kinetic energy density \(\rho_v^\varepsilon\) (second row).
FIG. 4: Temporal error of TSF-PIC ($N_\tau = 64$) at $T = \pi/2$ with respect to number of $M$ and $\varepsilon$: maximum error in the density $\rho^\varepsilon$ (first row) and in the kinetic energy density $\rho_v^\varepsilon$ (second row).

FIG. 5: Error of TSF-PIC in $\tau$ at $T = \pi/2$ with respect to number of grids $N_\tau$ and $\varepsilon$: maximum error in the density $\rho^\varepsilon$ (left) and in the kinetic energy density $\rho_v^\varepsilon$ (right).

for the help on the programming of multi-revolution composition method.

TABLE III: Temporal error of MRC4 ($N_m = 8M$), MRC2 ($N_m = M$), TSF method ($N_r = 64$) and limit solver (LS) ($N_r = 64$) in $\rho^\epsilon$ at $T = 2\pi$ with computational time (seconds) under several $M$ for $\epsilon = 1/4^3$ or $\epsilon = 1/4^6$.

<table>
<thead>
<tr>
<th>$\epsilon = 1/4^3$</th>
<th>$M = 2^2$</th>
<th>$M = 2^3$</th>
<th>$M = 2^4$</th>
<th>$\epsilon = 1/4^6$</th>
<th>$M = 2^2$</th>
<th>$M = 2^3$</th>
<th>$M = 2^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRC4</td>
<td>2.80E-3</td>
<td>1.62E-4</td>
<td>1.01E-5</td>
<td>MRC4</td>
<td>2.80E-3</td>
<td>1.67E-4</td>
<td>8.45E-6</td>
</tr>
<tr>
<td>cputime</td>
<td>99.1s</td>
<td>395s</td>
<td>1580s</td>
<td>cputime</td>
<td>101s</td>
<td>401s</td>
<td>1583s</td>
</tr>
<tr>
<td>MRC2</td>
<td>7.28E-1</td>
<td>4.86E-2</td>
<td>4.40E-3</td>
<td>MRC2</td>
<td>7.24E-1</td>
<td>4.65E-2</td>
<td>4.60E-3</td>
</tr>
<tr>
<td>cputime</td>
<td>1.35s</td>
<td>6.2s</td>
<td>25.6s</td>
<td>cputime</td>
<td>1.9s</td>
<td>6.7s</td>
<td>25.6s</td>
</tr>
<tr>
<td>TSF</td>
<td>2.62E-1</td>
<td>7.44E-2</td>
<td>1.86E-2</td>
<td>TSF</td>
<td>2.67E-1</td>
<td>7.71E-2</td>
<td>1.88E-2</td>
</tr>
<tr>
<td>cputime</td>
<td>73.7s</td>
<td>126s</td>
<td>231s</td>
<td>cputime</td>
<td>73.7s</td>
<td>126s</td>
<td>231s</td>
</tr>
<tr>
<td>LS</td>
<td>1.74E-1</td>
<td>5.02E-2</td>
<td>2.26E-2</td>
<td>LS</td>
<td>1.91E-1</td>
<td>5.40E-2</td>
<td>1.42E-2</td>
</tr>
<tr>
<td>cputime</td>
<td>19s</td>
<td>37.4s</td>
<td>75.8s</td>
<td>cputime</td>
<td>19s</td>
<td>37.4s</td>
<td>75.8s</td>
</tr>
</tbody>
</table>

FIG. 6: Efficiency plot of MRC2, MRC4 and TSF: relative error in $\rho^\epsilon$ at $T = 2\pi$ under $\epsilon = 1/4^3$ versus computation time.


FIG. 7: Long time test of TSF, MRC2 and MRC4: relative error in energy under $M = 2^7$ till $T = 32\pi$ ($\Delta t \approx 0.78$) for different $\varepsilon$.

FIG. 8: Convergence of the Vlasov-Poisson equation (I.1) to the limit model (I.2) in the asymptotic regime: difference in the density $\rho^\varepsilon$ (left) and in the kinetic energy density $\rho^\varepsilon_k$ (right).

FIG. 9: Time evolution of quantity $\xi(t)$ (III.3) under $\varepsilon = 1/4^3$ (first row) and $\varepsilon = 1/4^6$ (second row).


FIG. 10: Contour plot on $\rho^\varepsilon(t, x) - 1$ for different time with $\varepsilon = 1/4^3$.

FIG. 11: Contour plot of quantity $\chi^\varepsilon(t, v)$ in example at different $t$ with $\varepsilon = 1/4^3$.

FIG. 12: Relative numerical energy fluctuation $|H(t) - H(0)|/H(0)$ during the simulation of the Vlasov-Poisson equation (I.1) till $t = 32\pi$ under $\varepsilon = 1/4^3$ by the MRC2 method with different $M$.

