SIAM J. NUMER. ANAL.
Vol. 53, No. 1, pp. 619–643

AN ITERATION FREE BACKWARD SEMI-LAGRANGIAN SCHEME FOR GUIDING CENTER PROBLEMS*

XIANGFAN PIAO†, SANGDONG KIM‡, PHILSU KIM‡, JAE-MIN KWON§, AND DOKKYUN YI¶

Abstract. In this paper, we develop an iteration free backward semi-Lagrangian method for nonlinear guiding center models. We apply the fourth-order central difference scheme for the Poisson equation and employ the local cubic interpolation for the spatial discretization. A key problem in the time discretization is to find the characteristic curve arriving at each grid point which is the solution of a system of highly nonlinear ODEs with a self-consistency imposed by the Poisson equation. The proposed method is based on the error correction method recently developed by the authors. For the error correction method, we introduce a modified Euler's polygon and solve the induced asymptotically linear differential equation with the midpoint quadrature rule to get the error correction term. We prove that the proposed iteration free method has convergence order at least 3 in space and 2 in time in the sense of the L_2-norm. In particular, it is shown that the proposed method has a good performance in computational cost together with better conservation properties in mass, the total kinetic energy, and the enstrophy compared to the conventional second-order methods. Numerical test results are presented to support the theoretical analysis and discuss the properties of the newly proposed scheme.

Key words. error correction method, backward semi-Lagrangian method, temporal discretization, self-consistency, guiding center problem

AMS subject classifications. 35L04, 35Q83, 65M25, 65M06, 65M12

DOI. 10.1137/130942218

1. Introduction.

The model problem we are concerned with is the guiding center model, which was developed for an efficient description of low-frequency turbulence and resulting transport phenomena in strongly magnetized plasmas. Instead of tracing the fast gyro-motions of charged particles under strong external magnetic fields, the guiding center model follows the evolution of the center of the fast gyro-motions, which allows an efficient description of charged particle dynamics under relatively slow electrostatic fluctuations $\omega_e \ll \omega$. Here, ω denotes the gyro-frequency of a charged particle. If we suppose a uniform external magnetic field and the plane perpendicular to the magnetic field, the density of the guiding centers of charged particles, which are interacting with each other through self-consistent electrostatic potential, satisfies the following form of nonlinear hyperbolic equation in the plane with a proper normalization:

*Received by the editors October 24, 2013; accepted for publication (in revised form) December 5, 2014; published electronically February 24, 2015. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (grant 2011-0029013).
http://www.siam.org/journals/sinum/53-1/94221.html

†Department of Mathematics, Hannam University, Daejeon 306-791, Korea (piaoxf76@hanmail.net).

‡Department of Mathematics, Kyungpook National University, Daegu 702-701, Korea (skim@knu.ac.kr, kimps@knu.ac.kr).

§WCI Center for Fusion Theory, National Fusion Research Institute, Daejeon 305-333, Korea (jmkwon74@nfri.re.kr).

¶Ulsan National Institute of Science and Technology, School of Natural Science, Korea (dkyi2000@gmail.com). The research of this author was supported by the World Class Institute (WCI) Program of the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology of Korea (NRF grant WCI 2009-001).

$$\begin{cases} \dfrac{\partial}{\partial t}\rho(t,\mathbf{x}) + \mathbf{U}(t,\mathbf{x}) \cdot \nabla\rho(t,\mathbf{x}) = 0, \quad (t,\mathbf{x}) \in [0,T] \times \Omega, \\ \rho(0,\mathbf{x}) = \rho_0(\mathbf{x}), \end{cases} \tag{1.1}$$

where $\mathbf{x} := (x_1, x_2)$, $\Omega = [x_{1,min}, x_{1,max}] \times [x_{2,min}, x_{2,max}]$. Here, $\rho(t,\mathbf{x})$ denotes the density of the guiding centers with a given initial function ρ_0. The nonlinear advection field $\mathbf{U}(t,\mathbf{x})$ is defined as

$$\mathbf{U}(t,\mathbf{x}) = \big[U_1(t,\mathbf{x}), U_2(t,\mathbf{x})\big]^T := \Big[-\frac{\partial}{\partial x_2}\varphi(t,\mathbf{x}), \frac{\partial}{\partial x_1}\varphi(t,\mathbf{x})\Big]^T \tag{1.2}$$

for the self-consistent electrostatic potential function $\varphi(t,\mathbf{x})$ satisfying the Poisson equation with periodic boundary conditions in both directions x_1 and x_2:

$$-\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)\varphi(t,\mathbf{x}) = \rho(t,\mathbf{x}). \tag{1.3}$$

Here, ρ, φ, and $\mathbf{U}$ are assumed to be sufficiently smooth for the analysis. The three equations (1.1)–(1.3) are coupled to each other and hence the problem is highly nonlinear with a self-consistency arising from the Poisson equation.

We note that the structure of the nonlinearity is generic for many hyperbolic PDEs describing the dynamics of neutral fluid and plasma. For example, Hasegawa–Mima, Hasegawa–Wakatani, reduced MHD, and gyrokinetic equations for strongly magnetized plasma have such nonlinearity emerging from various drift motions of charged particles. Also, the two-dimensional incompressible Navier–Stokes equation with the stream function φ possesses such nonlinearity, which has a wide application in geophysical problems. Therefore, the model problem merits an effort to develop a numerical scheme for a broad application and also serves as an ideal test problem for newly developed scheme.

There are a lot of numerical methods for hyperbolic PDEs; we mention the papers [2, 7, 8, 10, 11, 12, 15, 16, 18, 20, 22, 23], which are not exhaustive. Numerical methods for the simulation of hyperbolic PDEs range from Lagrangian type such as the particle in cell (PIC) method to an Eulerian type based on a fixed grid system such as the finite volume method, the finite difference method, etc. Regarding the issues of stable long time simulation of PDEs, these methods have their own merits and weaknesses. For example, PIC methods show better conservation of key physical quantities compared to the Eulerian schemes. Also, they are relatively free from the CFL condition, which allows a larger size of time step with good advantages for long time simulation. However, PIC methods suffer from small-scale noise which accumulates and contaminates long time simulation result. Eulerian approaches such as finite difference and finite volume methods resolve these short-scale noise issues with grid-scale smoothing and dissipation. However, their CFL condition dictates a very small size of time step for explicit schemes, which makes long time simulation very costly. A semi-Lagrangian method tries to combine the advantages of both approaches, i.e., control of small-scale noise with better conservation property and larger time step size compared to the conventional Eulerian approaches.

For hyperbolic PDEs possessing time reversal symmetry, the so-called backward semi-Lagrangian method (BSLM) is proved to be very powerful for low-noise robust simulation [7, 24]. For the BSLM, let us review the following relations. Let $\boldsymbol{\pi}(s,\mathbf{x};t) := [\pi_1(t), \pi_2(t)]^T$ be the characteristic curve satisfying the ordinary differential equation (ODE) given by

$$\frac{d\boldsymbol{\pi}(s,\mathbf{x};t)}{dt} = \mathbf{U}(t, \boldsymbol{\pi}(s,\mathbf{x};t)), \quad \boldsymbol{\pi}(s,\mathbf{x};s) = \mathbf{x}, \tag{1.4}$$

where $\mathbf{U}$ is governed by the potential function φ. We introduce the Jacobian $J(s, \mathbf{x}; t)$ between Lagrangian and Eulerian coordinates as

$$J(s, \mathbf{x}; t) = \det\left(\frac{\partial \boldsymbol{\pi}(s, \mathbf{x}; t)}{\partial \mathbf{x}}\right).$$

Then, it can be seen that

$$\frac{\partial}{\partial t} J(s, \mathbf{x}; t) = J(s, \mathbf{x}; t) \nabla \cdot \mathbf{U}(t, \boldsymbol{\pi}(s, \mathbf{x}; t))$$

and the solution of (1.1) satisfies the equation $\rho(t, \boldsymbol{\pi}(s, \mathbf{x}; t)) = \rho(s, \mathbf{x}) J(s, \mathbf{x}; t)$. Since $\nabla \cdot \mathbf{U}(t, \boldsymbol{\pi}(s, \mathbf{x}; t)) \equiv 0$, the Jacobian is constant. Also, the mass conservation property

$$\int_\Omega \rho(s, \mathbf{x}) d\mathbf{x} \equiv \int_\Omega \rho(t, \boldsymbol{\pi}(s, \mathbf{x}; t)) d\boldsymbol{\pi}(s, \mathbf{x}; t) = \int_\Omega \rho(s, \mathbf{x}) J(s, \mathbf{x}; t) d\mathbf{x}$$

gives that $J(s, \mathbf{x}; t) \equiv 1$ and

$$\rho(s, \mathbf{x}) = \rho(t, \boldsymbol{\pi}(s, \mathbf{x}; t)). \tag{1.5}$$

Two essential structures of the BSLM can be explained with (1.4) and (1.5). One is used to evolve the values of the density ρ along the characteristic curve using the property (1.5). The other is used to find the characteristic curve satisfying (1.4) with a given initial value as the fixed grid points. The former problem can be solved with a technique of interpolation for the distribution function as well as a Poisson solver for (1.3). One may use standard interpolation techniques such as cubic spline and Hermite interpolations[3]. Also, there have been efforts to develop nonoscillatory interpolation method such as the piecewise weighted essentially nonoscillatory (WENO) method [1, 21] and the semi-Lagrangian WENO method [19]. Compared to these efforts to improve the techniques of spatial-discretization and interpolation, little attention was given to the temporal discretization and integration in BSLM. We note that for problems with explicitly known advection field $\mathbf{U}$, the implementations of various explicit time integration schemes are relatively straightforward without any complexity from iteration procedures to solve nonlinear equations [17]. However, the existence of a self-consistency and resulting advection field changes the situation significantly.

A self-consistent electrostatic potential φ satisfying the Poisson equation (1.3) determines the advection field for the evolution of the density ρ. For BSLM, the initial conditions for the time integration of the characteristic curves are given as grid points at a future time step, while the density ρ for the evaluation of φ and the advection field is known at the present time step. Therefore, the application of conventional time integration methods results in highly nonlinear self-consistency relations, which require numerical procedures to solve nonlinear self-consistent initial value problems. One popular method to solve the problems is an iterative scheme such as the fixed-point or Newton method. However, these iteration methods are prone to error accumulation and, also without a proper initial guess, the spatial interpolations of the potential and density increase computational costs for every additional iteration step, which can be a serious issue for long time simulations.

This paper aims to develop a BSLM which is iteration free. We apply the fourth-order central finite difference scheme for the Poisson problem (1.3) and use the local cubic interpolation polynomial for the spatial discretization on each time integration step. For an iteration free time discretization, we apply the error correction method

(ECM) recently developed by the authors (see [13, 14]) for solving the initial value problem (1.4). The ECM is based on the Euler's polygon on each time integration step and the induced asymptotically linear ODE of first order. To maintain the essential properties of ECM, we introduce a modified Euler's polygon and solve the asymptotic ODE with the midpoint quadrature rule. This leads to an iteration free method and overcomes the self-consistency. It is proved that the proposed method has the convergence property $\mathcal{O}(h^2 + \Delta x^3 + \frac{\Delta x^4}{h})$ in the sense of the L_2-norm, where h is the uniform time step size and Δx is the maximum spatial grid size. In particular, it is shown that the proposed scheme has not only a good performance in the computational cost but also better conservation properties in mass, the total kinetic energy and the enstrophy compared with the existing second-order methods.

This paper is organized as follows. In section 2, we discuss the local cubic interpolation technique and also a fast two-dimensional Poisson solver for solving (1.3) with periodic boundary condition. Section 3 is devoted to the derivation of the ECM for the time discretization. In section 4, we give a concrete analysis of the convergence for the proposed scheme. Numerical simulations are performed in section 5 to give evidence for the theoretical analysis in section 4. Finally, we provide some comments and conclusions in section 6.

2. Local cubic polynomial and fast Poisson solver. This section aims to give a brief review of the local cubic polynomial interpolation and discuss a fast two-dimensional Poisson solver with fourth-order accuracy. The time and spatial domains are assumed to be uniformly divided as follows:

$$\begin{aligned} &0 = t_0 < t_1 < \cdots < t_M = T, \quad t_m = mh, \\ &x_{k,min} = x_{k,0} < x_{k,1} < \cdots < x_{k,N_k} = x_{k,max}, \quad x_{k,i} = x_{k,0} + i\Delta x_k, \end{aligned} \tag{2.1}$$

where $h := T/M$ is the uniform time step size, and $\Delta x_k := (x_{k,max} - x_{k,min})/N_k$ $(k = 1, 2)$ are the uniform spatial mesh sizes in x_k directions, respectively. Let $(\mathbf{x} - \mathbf{x}_{i,j})^{k,l} := (x_1 - x_{1,i})^k (x_2 - x_{2,j})^l$ for grid points $\mathbf{x}_{i,j} := (x_{1,i}, x_{2,j})$ and $\mathcal{C}_{i,j} := [x_{1,i-1}, x_{1,i}] \times [x_{2,j-1}, x_{2,j}]$ be (i,j)th local cells. Also, let $\Gamma_{1,i,j}$ and $\Gamma_{2,i,j}$ be parts of the boundary of the cells $\mathcal{C}_{i,j}$ defined by $\Gamma_{1,i,j} := \{\mathbf{x} \in \mathcal{C}_{i,j} | x_1 = x_{1,i}, x_{2,j-1} < x_2 < x_{2,j}\}$ and $\Gamma_{2,i,j} := \{\mathbf{x} \in \mathcal{C}_{i,j} | x_2 = x_{2,j}, x_{1,i-1} < x_1 < x_{1,i}\}$, respectively. For a given function $f(\mathbf{x})$ and the grid point $\mathbf{x}_{i,j}$, let $\mathcal{P}_{i,j} f(\mathbf{x})$ be a (i,j)th local bi-cubic interpolation polynomial defined by

$$\mathcal{P}_{i,j} f(\mathbf{x}) = \sum_{k=0}^{3} \sum_{l=0}^{3} c_{k,l} (\mathbf{x} - \mathbf{x}_{i,j})^{k,l}, \quad \mathbf{x} \in \mathcal{C}_{i,j},$$

which solves the interpolation conditions

$$\mathcal{P}_{i,j} f(\mathbf{x}_{k,l}) = f(\mathbf{x}_{k,l}) := f_{k,l}, \quad i - 2 \le k \le i + 1, \quad j - 2 \le l \le j + 1.$$

Then, the explicit form of $\mathcal{P}_{i,j} f(\mathbf{x})$ is given by

$$\mathcal{P}_{i,j} f(\mathbf{x}) = \mathcal{L}(\mu_{1,i}) \mathcal{M}_{i,j}(f) \mathcal{L}(\mu_{2,j})^T, \tag{2.2}$$

where $\mathcal{L}(\mu_{k,l}) := \frac{1}{2} \left[\frac{\mu_{k,l} - \mu_{k,l}^3}{3}, \mu_{k,l}^3 + \mu_{k,l}^2 - 2\mu_{k,l}, 2 + \mu_{k,l} - 2\mu_{k,l}^2 - \mu_{k,l}^3, \frac{\mu_{k,l}^3 + 3\mu_{k,l}^2 + 2\mu_{k,l}}{3} \right]$, $\mu_{k,l} := \frac{x_k - x_{k,l}}{\Delta x_k}$, and

$$\mathcal{M}_{i,j}(f) := \begin{bmatrix} f_{i-2,j-2} & f_{i-2,j-1} & f_{i-2,j} & f_{i-2,j+1} \\ f_{i-1,j-2} & f_{i-1,j-1} & f_{i-1,j} & f_{i-1,j+1} \\ f_{i,j-2} & f_{i,j-1} & f_{i,j} & f_{i,j+1} \\ f_{i+1,j-2} & f_{i+1,j-1} & f_{i+1,j} & f_{i+1,j+1} \end{bmatrix}.$$

Using $\mathcal{P}_{i,j}f(\mathbf{x})$, the function $f(\mathbf{x})$ defined on Ω can be approximated by

$$f(\mathbf{x}) \approx \mathcal{P}\mathbf{f}(\mathbf{x}) := \sum_{i=1,j=1}^{N_1,N_2} \mathbf{1}_{\mathcal{C}_{ij}}(\mathbf{x})\mathcal{P}_{i,j}f(\mathbf{x}), \quad \mathbf{f} := [f(\mathbf{x}_{0,0}), f(\mathbf{x}_{1,0}), \dots, f(\mathbf{x}_{N_1,N_2})]^T,$$

whose truncation error has the magnitude $\mathcal{O}(\Delta x^4)$, where $\Delta x := \max(\Delta x_1, \Delta x_2)$ and $\mathbf{1}_{\mathcal{C}_{ij}}$ denotes the indicator function. Note that the introduced local cubic polynomial $\mathcal{P}\mathbf{f}(\mathbf{x})$ is continuous in Ω but not differentiable on the edge of the cells $\mathcal{C}_{i,j}$. As the approximation of the derivatives of f, we use the following approximations to avoid nondifferentiability of $\mathcal{P}\mathbf{f}(\mathbf{x})$ on edges:

(2.3)
$$D^{\boldsymbol{\beta}}f(\mathbf{x}) \approx \widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\mathbf{x}) := \begin{cases} \frac{1}{2}\sum_{l=0}^{1} D^{\boldsymbol{\beta}}\mathcal{P}_{i+l,j}f(\mathbf{x}) & \text{if } \beta_1 \neq 0,\ \mathbf{x} \in \Gamma_{1,i,j}, \\ \frac{1}{2}\sum_{l=0}^{1} D^{\boldsymbol{\beta}}\mathcal{P}_{i,j+l}f(\mathbf{x}) & \text{if } \beta_2 \neq 0,\ \mathbf{x} \in \Gamma_{2,i,j}, \\ \frac{1}{4}\sum_{l=0}^{1}\sum_{k=0}^{1} D^{\boldsymbol{\beta}}\mathcal{P}_{i+l,j+k}f(\mathbf{x}) & \text{if } \beta_1 = \beta_2 = 1,\ \mathbf{x} = \mathbf{x}_{i,j}, \\ \frac{1}{2}\sum_{l=0}^{1} D^{\boldsymbol{\beta}}\mathcal{P}_{i+l,j}f(\mathbf{x}) & \text{if } \beta_1 \neq 0,\ \beta_2 = 0,\ \mathbf{x} = \mathbf{x}_{i,j} \\ \frac{1}{2}\sum_{l=0}^{1} D^{\boldsymbol{\beta}}\mathcal{P}_{i,j+l}f(\mathbf{x}) & \text{if } \beta_1 = 0,\ \beta_2 \neq 0,\ \mathbf{x} = \mathbf{x}_{i,j}, \\ D^{\boldsymbol{\beta}}\mathcal{P}_{i,j}f(\mathbf{x}) & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2)$, $\beta_i =$ integer ≥ 0, with $|\boldsymbol{\beta}| = \beta_1 + \beta_2 \leq 2$, is a multi-index and $D^{\boldsymbol{\beta}}u := \frac{\partial^{|\boldsymbol{\beta}|}u}{\partial x_1^{\beta_1}\partial x_2^{\beta_2}}$. Here, we regard $D^{\boldsymbol{\beta}}\mathcal{P}_{i,j}f(\mathbf{x})$ on the edge of the cell $\mathcal{C}_{i,j}$ defined on (2.3) as either the left derivative or the right derivative.

We now discuss a fast solver for the Poisson equation (1.3) with periodic boundary conditions. In particular, for given approximations $\rho_{i,j}^m$ of the analytic solutions of the density ρ at time t_m and at the grid point $\mathbf{x}_{i,j}$, we are going to describe a fourth-order central difference method to solve (1.3) at time t_m. Also, we will describe an approximation scheme of the advection field $\mathbf{U}(t, \mathbf{x})$ based on the interpolation scheme discussed above.

For convenience, let us denote $\varphi_{i,j}(t) := \varphi(t, \mathbf{x}_{i,j})$ and $\rho_{i,j}(t) := \rho(t, \mathbf{x}_{i,j})$ for each grid point $\mathbf{x}_{i,j}$ $(0 \leq i \leq N_1 - 1,\ 0 \leq j \leq N_2 - 1)$. Then, employing the fourth-order central finite difference for the second derivatives (for example, see [5]) given by

$$\frac{\partial^2}{\partial x_1^2}\varphi_{i,j}(t) = \frac{-\varphi_{i-2,j}(t) + 16\varphi_{i-1,j}(t) - 30\varphi_{i,j}(t) + 16\varphi_{i+1,j}(t) - \varphi_{i+2,j}(t)}{12\Delta x_1^2} + \mathcal{O}\Big(\Delta x_1^4\Big),$$
$$\frac{\partial^2}{\partial x_2^2}\varphi_{i,j}(t) = \frac{-\varphi_{i,j-2}(t) + 16\varphi_{i,j-1}(t) - 30\varphi_{i,j}(t) + 16\varphi_{i,j+1}(t) - \varphi_{i,j+2}(t)}{12\Delta x_2^2} + \mathcal{O}\Big(\Delta x_2^4\Big),$$

one may discretize the Poisson equation (1.3) as follows:

(2.4)
$$\begin{aligned} &\frac{\varphi_{i-2,j}(t_m) - 16\varphi_{i-1,j}(t_m) + 30\varphi_{i,j}(t_m) - 16\varphi_{i+1,j}(t_m) + \varphi_{i+2,j}(t_m)}{12\Delta x_1^2} \\ &\quad + \frac{\varphi_{i,j-2}(t_m) - 16\varphi_{i,j-1}(t_m) + 30\varphi_{i,j}(t_m) - 16\varphi_{i,j+1}(t_m) + \varphi_{i,j+2}(t_m)}{12\Delta x_2^2} \\ &= \rho_{i,j}(t_m) + \mathcal{O}(\Delta x^4). \end{aligned}$$

For convenience, let us define vectors $\widehat{\boldsymbol{\phi}}^m$ and $\widehat{\boldsymbol{\rho}}^m$ by

$$\begin{aligned}\widehat{\boldsymbol{\phi}}^m &:= \Big[\varphi_{0,0}(t_m), \varphi_{1,0}(t_m), \ldots, \varphi_{N_1-1,0}(t_m), \varphi_{0,1}(t_m), \ldots, \varphi_{N_1-1,N_2-1}(t_m)\Big]^T,\\ \widehat{\boldsymbol{\rho}}^m &:= \Big[\rho_{0,0}(t_m), \rho_{1,0}(t_m), \ldots, \rho_{N_1-1,0}(t_m), \rho_{0,1}(t_m), \ldots, \rho_{N_1-1,N_2-1}(t_m)\Big]^T\end{aligned} \tag{2.5}$$

and a matrix $\mathcal{S}$ by

$$\mathcal{S} := \mathcal{I}_2 \otimes \mathcal{R}_1 + \mathcal{R}_2 \otimes \mathcal{I}_1, \tag{2.6}$$

where $\mathcal{I}_k$ is the $N_k \times N_k$ identity matrix and $\mathcal{R}_k$ is a sparse matrix whose (i,j) entries are given by

(2.7)

$$\mathcal{R}_k := \Big(\mathcal{R}_k(i,j)\Big)_{N_k\times N_k},\ \ \mathcal{R}_k(i,j) = \frac{1}{12\Delta x_k^2}\begin{cases}30 & \text{if } \ i=j,\\ -16 & \text{if } \ |i-j| = N_k-1 \text{ or } |i-j|=1,\\ 1 & \text{if } \ |i-j| = N_k-2 \text{ or } |i-j|=2,\\ 0 & \text{otherwise.}\end{cases}$$

The notation $\otimes$ denotes the tensor product. Then, one may get a system of equations for (2.4) given by

$$\mathcal{S}\widehat{\boldsymbol{\phi}}^m = \widehat{\boldsymbol{\rho}}^m + \mathcal{O}(\Delta x^4). \tag{2.8}$$

Now, we define vectors $\boldsymbol{\phi}^m$ and $\boldsymbol{\rho}^m$ by

$$\begin{aligned}\boldsymbol{\phi}^m &:= \Big[\varphi^m_{0,0}, \varphi^m_{1,0}, \ldots, \varphi^m_{N_1-1,0}, \varphi^m_{0,1}, \ldots, \varphi^m_{N_1-1,N_2-1}\Big]^T,\\ \boldsymbol{\rho}^m &:= \Big[\rho^m_{0,0}, \rho^m_{1,0}, \ldots, \rho^m_{N_1-1,0}, \rho^m_{0,1}, \ldots, \rho^m_{N_1-1,N_2-1}\Big]^T.\end{aligned} \tag{2.9}$$

Then, instead of solving (2.8) for $\varphi_{i,j}(t_m)$, we will approximate the solution $\varphi_{i,j}(t_m)$ by $\varphi^m_{i,j}$, which solves the system

$$\mathcal{S}\boldsymbol{\phi}^m = \boldsymbol{\rho}^m. \tag{2.10}$$

Notice that the matrix $\mathcal{R}_k$ is a symmetric circulant matrix and its eigenvalue decomposition (see [4]) is given by

$$\mathcal{R}_k = \mathcal{Q}_k \Lambda_k \mathcal{Q}_k^T, \tag{2.11}$$

where Λ_k and $\mathcal{Q}_k$ $(k=1,2)$ are defined by

(2.12)

$$\Lambda_k = \operatorname{diag}(\lambda_{k,1}, \ldots, \lambda_{k,N_k}), \quad \lambda_{k,j} = \frac{1}{6\Delta x_k^2}\Big(\cos\Big(\frac{4j\pi}{N_k}\Big) - 16\cos\Big(\frac{2j\pi}{N_k}\Big) + 15\Big),$$

$$\mathcal{Q}_k = \Big(\mathbf{q}_k^{(1)}, \ldots, \mathbf{q}_k^{(N_k)}\Big), \quad \mathbf{q}^{(j)}_{k,l} := \frac{1}{\sqrt{N_k}}\begin{cases}\cos(\frac{2\pi jl}{N_k}), & 2j < N_k,\\ \sqrt{2}\cos(\frac{2\pi jl}{N_k}), & 2j = N_k \text{ or } j = N_k,\\ \sin(\frac{2\pi(N_k-j)l}{N_k}) & \text{otherwise.}\end{cases}$$

Recalling the property of tensor operator [9] and combining (2.6), (2.11), and (2.12), we get

$$\Lambda\widetilde{\boldsymbol{\phi}} := \Big(\mathcal{I}_2 \otimes \Lambda_1 + \Lambda_2 \otimes \mathcal{I}_1\Big)\widetilde{\boldsymbol{\phi}} = \widetilde{\boldsymbol{\rho}}, \tag{2.13}$$

where

$$\widetilde{\boldsymbol{\Phi}} = \Big(\mathcal{Q}_2^T \otimes \mathcal{Q}_1^T\Big)\boldsymbol{\Phi}^m, \quad \widetilde{\boldsymbol{\rho}} = \Big(\mathcal{Q}_2^T \otimes \mathcal{Q}_1^T\Big)\boldsymbol{\rho}^m. \tag{2.14}$$

To speed up the computational time in solving the system (2.13) (or (2.10)), we use the following property of the tensor product:

$$\mathcal{A} \otimes \mathcal{B}\mathbf{u} = \mathcal{B}\mathcal{U}\mathcal{A}^T,$$

where $\mathcal{A}$ and $\mathcal{B}$ are any square matrices and $\mathcal{U}$ is a matrix constructed by reshaping the vector $\mathbf{u}$. For a detailed property and the computational efficiency of it, we refer to [9].

LEMMA 2.1. *All eigenvalues $\lambda_{k,i}$ defined in* (2.12) *are nonnegative and the zero eigenvalue occurs only when $i = N_k$. Furthermore, the positive eigenvalues can be estimated by*

$$\frac{16}{3\Delta x_k^2} \geq \lambda_{k,j} \geq \Big(\frac{2\pi}{L_k}\Big)^2 - \frac{1}{90}\Big(\frac{2\pi}{L_k}\Big)^6 \Delta x^4, \tag{2.15}$$

where $L_k = x_{k,max} - x_{k,min}$ and $k = 1, 2$.

Proof. The eigenvalue $\lambda_{k,j}$ in (2.12) can be factorized by

$$\lambda_{k,j} = \frac{1}{3\Delta x_k^2}\Big(\cos(j\alpha) - 1\Big)\Big(\cos(j\alpha) - 7\Big) \geq 0, \quad \alpha = \frac{2\pi}{N_k}. \tag{2.16}$$

It implies that $\lambda_{k,j} = 0$ if and only if $j = N_k$. Further, when $\cos(j\alpha) = -1$, $\lambda_{k,j}$ has the maximum value $\frac{16}{3\Delta x_k^2}$. To get a lower bound of $\lambda_{k,j}$, we first note that $\lambda_{k,j}$, as a function of $j\alpha$, is increasing in $(0, \pi)$ and decreasing in $(\pi, 2\pi)$, which can be easily seen from the derivative of the function. Also, $\lambda_{k,j} = \lambda_{k,N_k-j}$ due to $\cos(j\alpha) = \cos((N_k - j)\alpha)$. Therefore, the minimum of positive eigenvalues are $\lambda_{k,1}$ and λ_{k,N_k-1}. Thus, $\lambda_{k,1}$ can be estimated with the Taylor series with N_k large enough as follows:

$$\begin{aligned}\lambda_{k,j} \geq \lambda_{k,1} &= \frac{\alpha^2}{\Delta x_k^2} - \frac{1}{90}\frac{\alpha^6}{\Delta x_k^2} + \frac{1}{1008}\frac{\alpha^8}{\Delta x_k^2} + \mathcal{O}(\Delta x_k^8) \\ &> \frac{\alpha^2}{\Delta x_k^2} - \frac{1}{90}\frac{\alpha^6}{\Delta x_k^2} = \Big(\frac{2\pi}{L_k}\Big)^2 - \frac{1}{90}\Big(\frac{2\pi}{L_k}\Big)^6 \Delta x^4.\end{aligned}$$

This completes the proof. ∎

Remark 2.2. All diagonal elements of the matrix Λ defined in (2.12) are of the form $\lambda_{1,i} + \lambda_{2,j}$. Hence, Lemma 2.1 shows that only one of the elements of the diagonal matrix Λ is zero, which implies the Poisson equation and the system (2.10) has a unique solution up to a constant. To solve (2.13), we simply take the value 0 for the solution and $\widetilde{\boldsymbol{\rho}}$ of the corresponding position where the eigenvalue of Λ is zero and replace the corresponding eigenvalue with nonzero data (for example, we take 1); then the invertibility of matrix Λ will give a unique solution. The matrix obtained by modifying the zero eigenvalue of $\mathcal{S}$ in the above sense is invertible, and let $\mathcal{S}^\dagger$ be the inverse of the matrix. For simplicity, we call $\mathcal{S}^\dagger$ a pseudoinverse of $\mathcal{S}$.

In this paper, we will study the convergence for the scheme we develop in the sense of the L_2-norm. To do this, we introduce the following convectional L_2 and discrete l_2-norms as

$$\|f\|_2 := \sqrt{\int_\Omega f(\mathbf{x})^2 d\mathbf{x}}, \quad \|f\|_{l_2,\Delta x} := \Delta x \sqrt{\sum_{i=0}^{N_1}\sum_{j=0}^{N_2} f_{i,j}^2} = \Delta x \|\mathbf{f}\|_2, \tag{2.17}$$

where $\|\mathbf{f}\|_2$ is given by

$$\|\mathbf{f}\|_2 := \sqrt{\sum_{i=0}^{N_1}\sum_{j=0}^{N_2} f_{i,j}^2} \tag{2.18}$$

with $\mathbf{f} := [f_{0,0}, f_{1,0}, \ldots, f_{N_1,0}, f_{0,1}, \ldots, f_{N_1,N_2}]^T$. Using the fact $\|f\|_{l_2,\Delta x} = \|\mathcal{P}\mathbf{f}\|_{l_2,\Delta x}$ and the same technique used in Lemma 2.1 of [6], one can get the following equivalent relation between the L_2-norm $\|\cdot\|_2$ and the discrete l_2-norm $\|\cdot\|_{l_2,\Delta x}$:

$$C_1\|f\|_{l_2,\Delta x} \le \|\mathcal{P}\mathbf{f}\|_2 \le C_2\|f\|_{l_2,\Delta x} \tag{2.19}$$

for positive constants C_1 and C_2.

The following lemma gives approximation properties for the derivatives $\widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}f(\mathbf{x})$ defined by (2.3).

LEMMA 2.3. *For a given sufficiently smooth function $f(\mathbf{x})$, the approximate polynomial $\widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\mathbf{x})$ defined by* (2.3) *has the approximation properties*

$$\|D^{\boldsymbol{\beta}}f(\cdot) - \widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\cdot)\|_2 = \mathcal{O}\big(\Delta x^{4-|\boldsymbol{\beta}|}\big),$$

where $0 \le |\boldsymbol{\beta}| \le 2$.

Proof. It is easy to see that the approximations $\widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\mathbf{x})$ defined by (2.3) are piecewise polynomials and bounded in whole cells of the computational domain. Hence, $\widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\mathbf{x})$ belong to the L^2 space. Using (2.2) and $\frac{d}{dx_k}\mu_{k,l} = \frac{1}{\Delta x_k}$, it can be easily shown that the relation

$$D^{\boldsymbol{\beta}}f(\mathbf{x}) = \widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\mathbf{f}(\mathbf{x}) + \mathcal{O}\big(\Delta x^{4-|\boldsymbol{\beta}|}\big) \tag{2.20}$$

is valid for the cases $\mathbf{x}$ in the interior of $\mathcal{C}_{i,j}$ or $\mathbf{x} \in \Gamma_{1,i,j}, \beta_1 = 0$ or $\mathbf{x} \in \Gamma_{2,i,j}, \beta_2 = 0$. For the case $\mathbf{x} \in \Gamma_{1,i,j}$, the Taylor series expansion gives

$$\begin{aligned}
\widetilde{D}^{(1,0)}\mathcal{P}\mathbf{f}(\mathbf{x}) &= \frac{1}{12\Delta x_1}\big([1,-6,3,2]\mathcal{M}_{i,j} + [-2,-3,6,-1]\mathcal{M}_{i+1,j}\big)\mathcal{L}(\mu_{2,j})^T \\
&= \frac{1}{12\Delta x_1}\sum_{l=-2}^{2} w_{1,l}\left[f_{i-l,j-2}, f_{i-l,j-1}, f_{i-l,j}, f_{i-l,j+1}\right]\mathcal{L}(\mu_{2,j})^T \\
&= \big[D^{(1,0)}f(\mathbf{x}_{i,j-2}), D^{(1,0)}f(\mathbf{x}_{i,j-1}), D^{(1,0)}f(\mathbf{x}_{i,j}), \\
&\quad D^{(1,0)}f(\mathbf{x}_{i,j+1})\big]\mathcal{L}(\mu_{2,j})^T + \mathcal{O}(\Delta x_1^4) \\
&= D^{(1,0)}f(\mathbf{x}) + \mathcal{O}(\Delta x_1^4)
\end{aligned} \tag{2.21}$$

and

$$
\begin{aligned}
\widetilde{D}^{(2,0)}\mathcal{P}\mathbf{f}(\mathbf{x}) &= \frac{1}{4\Delta x_1^2}\big([0,2,-4,2]\mathcal{M}_{i,j} + [2,-4,2,0]\mathcal{M}_{i+1,j}\big)\mathcal{L}(\mu_{2,j})^T \\
&= \frac{1}{\Delta x_1^2}\sum_{l=-1}^{1} w_{2,l}\left[f_{i-l,j-1}, f_{i-l,j}, f_{i-l,j+1}\right]\mathcal{L}(\mu_{2,j})^T \\
&= \big[D^{(2,0)}f(\mathbf{x}_{i,j-2}), D^{(2,0)}f(\mathbf{x}_{i,j-1}), D^{(2,0)}f(\mathbf{x}_{i,j}), \\
&\quad D^{(2,0)}f(\mathbf{x}_{i,j+1})\big]\mathcal{L}(\mu_{2,j})^T + \mathcal{O}(\Delta x_1^2) \\
&= D^{(2,0)}f(\mathbf{x}) + \mathcal{O}(\Delta x_1^2),
\end{aligned} \tag{2.22}
$$

where $w_{1,l}$ and $w_{2,l}$ are defined by

$$w_{1,-2}=1,\ w_{1,-1}=-8,\ w_{1,0}=0,\ w_{1,1}=8,\ w_{1,2}=-1,\ w_{2,-1}=1,\ w_{2,0}=-2,\ w_{2,1}=1.$$

In a similar way, we can show that

$$
\begin{aligned}
&\widetilde{D}^{(1,1)}\mathcal{P}\mathbf{f}(\mathbf{x}) = D^{(1,1)}f(\mathbf{x}) + \mathcal{O}(\Delta x^3) \quad \text{if } \mathbf{x}=\mathbf{x}_{i,j} \text{ or } \mathbf{x}\in\Gamma_{1,i,j} \text{ or } \mathbf{x}\in\Gamma_{2,i,j}, \\
&\widetilde{D}^{(0,1)}\mathcal{P}\mathbf{f}(\mathbf{x}) = D^{(0,1)}f(\mathbf{x}) + \mathcal{O}(\Delta x_2^4) \quad \text{if } \mathbf{x}\in\Gamma_{2,i,j}, \\
&\widetilde{D}^{(0,2)}\mathcal{P}\mathbf{f}(\mathbf{x}) = D^{(0,2)}f(\mathbf{x}) + \mathcal{O}(\Delta x_2^2) \quad \text{if } \mathbf{x}\in\Gamma_{2,i,j}.
\end{aligned} \tag{2.23}
$$

Summarizing (2.20), (2.21), (2.22), and (2.23), one can complete the proof. ∎

From Lemma 2.1, it can be shown that the pseudoinverse $\mathcal{S}^\dagger$ of $\mathcal{S}$ is uniformly bounded in the L_2-norm as follows.

LEMMA 2.4. *The upper bound of* $\|\mathcal{S}^\dagger\|_2$ *is estimated by*

$$\|\mathcal{S}^\dagger\|_2 \le C_1 := \max_k\Big(\Big(\frac{2\pi}{L_k}\Big)^2 - \frac{1}{90}\Big(\frac{2\pi}{L_k}\Big)^6\Delta x^4\Big)^{-1}. \tag{2.24}$$

Proof. Since $\mathcal{S}$ is symmetric, $\mathcal{S}^\dagger$ is symmetric. Hence, by the well-known fact $\|A\|_2 = \sqrt{\lambda_{max}(A^*A)}$ and Lemma 2.1, the matrix norm $\|\mathcal{S}^\dagger\|_2$ can be estimated by

$$\|\mathcal{S}^\dagger\|_2 = \max_k\Big(\frac{1}{\lambda_{k,1}}\Big) \le \max\Big(\Big(\frac{2\pi}{L_k}\Big)^2 - \frac{1}{90}\Big(\frac{2\pi}{L_k}\Big)^6\Delta x^4\Big)^{-1}$$

which completes the proof. ∎

For the approximate solutions $\boldsymbol{\phi}^m$ and $\boldsymbol{\rho}^m$, let us assume that the cubic interpolation polynomials $\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})$ and $\mathcal{P}\boldsymbol{\rho}^m(\mathbf{x})$ which solve

$$\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x}_{i,j}) = \varphi_{i,j}^m, \quad \mathcal{P}\boldsymbol{\rho}^m(\mathbf{x}_{i,j}) = \rho_{i,j}^m, \quad 0\le i\le N_1-1, \quad 0\le j\le N_2-1, \tag{2.25}$$

are constructed. Then, we define the errors $\mathbf{e}^m$ and $\mathbf{E}^m$ for potential $\varphi(t,\mathbf{x})$ and density $\rho(t,\mathbf{x})$ at time $t=t_m$, respectively, by

$$\mathbf{e}^m := \|\varphi(t_m,\cdot) - \mathcal{P}\boldsymbol{\phi}^m\|_2, \quad \mathbf{E}^m := \|\rho(t_m,\cdot) - \mathcal{P}\boldsymbol{\rho}^m\|_2. \tag{2.26}$$

Then, we have the following lemmas.

LEMMA 2.5. *The errors* $\mathbf{e}^m$ *and* $\mathbf{E}^m$ *defined by* (2.26) *satisfies the inequality*

$$\mathbf{e}^m \le C\Big(\mathbf{E}^m + \Delta x^4\Big) \tag{2.27}$$

for some constant C *only depending on* L_k.

Proof. Using (2.17) and (2.19), one may have

$$\mathbf{e}^m \le \|\varphi(t_m,\cdot)-\mathcal{P}\widehat{\boldsymbol{\phi}}^m\|_2+\|\mathcal{P}(\widehat{\boldsymbol{\phi}}^m-\boldsymbol{\phi}^m)\|_2 \le C\Big(\Delta x^4+\Delta x\|\widehat{\boldsymbol{\phi}}^m-\boldsymbol{\phi}^m\|_2\Big). \tag{2.28}$$

Combining (2.8), (2.10), and Lemma 2.4, one can prove that

$$\|\widehat{\boldsymbol{\phi}}^m-\boldsymbol{\phi}^m\|_2=\|\mathcal{S}^\dagger(\widehat{\boldsymbol{\rho}}^m-\boldsymbol{\rho}^m)\|_2+\mathcal{O}(\Delta x^4)\le C\|\widehat{\boldsymbol{\rho}}^m-\boldsymbol{\rho}^m\|_2+\mathcal{O}(\Delta x^4). \tag{2.29}$$

From (2.17) and (2.19), substituting (2.29) into (2.28) gives the desired inequality. ∎

LEMMA 2.6. *For the solution $\varphi(t,\mathbf{x})$ of (1.3), the cubic interpolation polynomial $\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})$ satisfies*

$$\|D^{\boldsymbol{\beta}}\varphi(t_m,\cdot)-\widetilde{D}^{\boldsymbol{\beta}}\mathcal{P}\boldsymbol{\phi}^m(\cdot)\|_2=\mathcal{O}\Big(\frac{\mathbf{E}^m}{\Delta x^{|\boldsymbol{\beta}|}}+\Delta x^{4-|\boldsymbol{\beta}|}\Big),$$

where $\boldsymbol{\beta}=(\beta_1,\beta_2)$, $\beta_i=$ integer ≥ 0, with $|\boldsymbol{\beta}|=\beta_1+\beta_2\le 2$, is a multi-index, and $D^{\boldsymbol{\beta}}u:=\frac{\partial^{|\boldsymbol{\beta}|}u}{\partial x_1^{\beta_1}\partial x_2^{\beta_2}}$.

Proof. From (2.2), $\frac{d\mu_{k,l}}{dx_k}=\frac{1}{\Delta x_k}$. Hence, combining Lemmas 2.3 and 2.5 directly leads to the required result. ∎

If we define

$$\mathbf{U}^m(\mathbf{x}):=\Big[-\widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x}),\widetilde{D}^{(1,0)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\Big]^T, \tag{2.30}$$

then Lemma 2.6 and (1.2) give

$$\|\mathbf{U}(t_m,\cdot)-\mathbf{U}^m\|_2=\mathcal{O}\left(\frac{\mathbf{E}^m}{\Delta x}+\Delta x^3\right). \tag{2.31}$$

Also, if we define

$$\mathbf{U}_{\boldsymbol{\pi}}(t_m,\mathbf{x}):=\begin{pmatrix}-\frac{\partial^2}{\partial x_1\partial x_2} & -\frac{\partial^2}{\partial x_2^2}\\ \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1\partial x_2}\end{pmatrix}\varphi(t_m,\mathbf{x}),\ \mathcal{J}^m(\mathbf{x}):=\begin{pmatrix}-\widetilde{D}^{(1,1)} & -\widetilde{D}^{(0,2)}\\ \widetilde{D}^{(2,0)} & \widetilde{D}^{(1,1)}\end{pmatrix}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x}), \tag{2.32}$$

then one can see that

$$\|\mathbf{U}_{\boldsymbol{\pi}}(t_m,\cdot)-\mathcal{J}^m\|_2=\mathcal{O}\left(\frac{\mathbf{E}^m}{\Delta x^2}+\Delta x^2\right). \tag{2.33}$$

Further, the following relation between $\mathbf{U}^m$ and $\mathcal{J}^m$ is valid.

LEMMA 2.7. *For fixed indices i and j, we assume that $\mathbf{y}$ is in a neighborhood of $\mathbf{x}\in\mathcal{C}_{i,j}$, in particular $\mathbf{y}\in\bigcup_{\substack{k=0,1\\ l=0,1}}\mathcal{C}_{i+k,j+l}$. Then, it is valid that*

$$\mathbf{U}^m(\mathbf{y})-\mathbf{U}^m(\mathbf{x})=\mathcal{J}^m(\mathbf{x})(\mathbf{y}-\mathbf{x})+\mathcal{O}\Big(\Delta x^3+\big(\mathbf{y}-\mathbf{x}\big)^2\Big). \tag{2.34}$$

Proof. It is enough to prove the assertion for three cases: (i) $\mathbf{y}\in\mathcal{C}_{i,j}$, (ii) $\mathbf{y}\in\mathcal{C}_{i,j+1}$, and (iii) $\mathbf{y}\in\Gamma_{2,i,j}$. A similar method can be applied to prove the assertion

for the other cases. For the first case $\mathbf{y} \in \mathcal{C}_{i,j}$, it is easy to see that the definition of the piecewise cubic interpolation and the derivative $\tilde{D}^{\boldsymbol{\beta}}$ yield

$$\begin{aligned}(2.35)\qquad &\widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{y}) - \widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\\ &\quad = \Big[\widetilde{D}^{(1,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x}), \widetilde{D}^{(0,2)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\Big](\mathbf{y}-\mathbf{x})^T + \mathcal{O}((\mathbf{y}-\mathbf{x})^2).\end{aligned}$$

For the second case $\mathbf{y} \in \mathcal{C}_{i,j+1}$, one can also get that

$$\begin{aligned}(2.36)\qquad &\widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{y}) - \widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x}) = D^{(0,1)}\mathcal{P}_{i,j+1}\boldsymbol{\phi}^m(\mathbf{y}) - D^{(0,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\\ &= \Big(D^{(0,1)}\mathcal{P}_{i,j+1}\boldsymbol{\phi}^m(\mathbf{y}) - D^{(0,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{y})\Big)\\ &\quad + \Big(D^{(0,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{y}) - D^{(0,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\Big)\\ &= \Big[\widetilde{D}^{(1,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x}), \widetilde{D}^{(0,2)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\Big](\mathbf{y}-\mathbf{x})^T + \mathcal{O}\Big(\Delta x^3 + (\mathbf{y}-\mathbf{x})^2\Big).\end{aligned}$$

Finally, for the case $\mathbf{y} \in \Gamma_{2,i,j}$, combining (2.35), (2.36), and (2.3) leads to

$$\begin{aligned}(2.37)\qquad &\widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{y}) - \widetilde{D}^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\\ &= \frac{1}{2}\Big(\lim_{\substack{\mathbf{z}\to\mathbf{y}\\ \mathbf{z}\in\mathcal{C}_{i,j}}} D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{z}) + \lim_{\substack{\mathbf{z}\to\mathbf{y}\\ \mathbf{z}\in\mathcal{C}_{i,j+1}}} D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{z})\Big) - D^{(0,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\\ &= \lim_{\substack{\mathbf{z}\to\mathbf{y}\\ \mathbf{z}\in\mathcal{C}_{i,j}}} \frac{1}{2}\Big(D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{z}) - D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\Big)\\ &\quad + \lim_{\substack{\mathbf{z}\to\mathbf{y}\\ \mathbf{z}\in\mathcal{C}_{i,j+1}}} \frac{1}{2}\Big(D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{z}) - D^{(0,1)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\Big)\\ &= \Big[\widetilde{D}^{(1,1)}P_{i,j}\boldsymbol{\phi}^m(\mathbf{x}), \widetilde{D}^{(0,2)}P_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\Big](\mathbf{y}-\mathbf{x})^T + \mathcal{O}\Big(\Delta x^3 + (\mathbf{y}-\mathbf{x})^2\Big).\end{aligned}$$

In a similar way, one can show that

$$\begin{aligned}(2.38)\qquad &\widetilde{D}^{(1,0)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{y}) - \widetilde{D}^{(1,0)}\mathcal{P}\boldsymbol{\phi}^m(\mathbf{x})\\ &\quad = \Big[\widetilde{D}^{(2,0)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x}), \widetilde{D}^{(1,1)}\mathcal{P}_{i,j}\boldsymbol{\phi}^m(\mathbf{x})\Big](\mathbf{y}-\mathbf{x})^T + \mathcal{O}\Big(\Delta x^3 + (\mathbf{y}-\mathbf{x})^2\Big).\end{aligned}$$

Combining (2.30), (2.32), and (2.35)–(2.38), one can get the identity (2.34). ∎

3. Error correction method. The target of this section is to solve the nonlinear self-consistent initial value problem described by

$$(3.1)\qquad \begin{cases}\dfrac{d\boldsymbol{\pi}(t_{m+1},\mathbf{x};t)}{dt} = \mathbf{U}(t,\boldsymbol{\pi}(t_{m+1},\mathbf{x};t)), \quad t < t_{m+1},\\ \boldsymbol{\pi}(t_{m+1},\mathbf{x};t_{m+1}) = \mathbf{x},\end{cases}$$

where $\boldsymbol{\pi}(t_{m+1},\mathbf{x};t) := [\pi_1(t),\pi_2(t)]^T$ is the characteristic curve and $\mathbf{U}$ is the advection field constrained by (1.2) and (1.3). In particular, we focus on finding the starting position $\boldsymbol{\pi}(t_{m+1},\mathbf{x};t_{m-1})$ at time t_{m-1} for the BSLM. The section consists of two parts. The first part reviews the existing second-order iteration methods. Then we introduce our method in the second part.

3.1. Second-order iteration method. The second-order iteration method for solving (3.1) first integrates (3.1) over $[t_{m-1}, t_{m+1}]$ and then applies the midpoint rule. Then, one may get

$$\mathbf{x} - \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}) = 2h\mathbf{U}(t_m, \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_m)) + \mathcal{O}(h^3). \tag{3.2}$$

Replacing the unknown value $\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_m)$ in (3.2) with the mean value of $\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t)$ at time t_{m-1} and t_{m+1} yields

$$\mathbf{x} - \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}) = 2h\mathbf{U}\left(t_m, \frac{1}{2}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}) + \mathbf{x})\right) + \mathcal{O}(h^3), \tag{3.3}$$

which is a nonlinear equation for $\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})$. To simplify the nonlinear equation, let

$$\boldsymbol{\alpha} := \frac{1}{2}\Big(\mathbf{x} - \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})\Big). \tag{3.4}$$

Then, (3.3) can be written by

$$\boldsymbol{\alpha} = h\mathbf{U}(t_m, \mathbf{x} - \boldsymbol{\alpha}) + \mathcal{O}(h^3). \tag{3.5}$$

Once one finds the solution $\boldsymbol{\alpha}$ for (3.5), one may get a seconds order discretization scheme to get $\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})$ with (3.4). As a numerical scheme for the nonlinear equation (3.5), one may apply an iteration method such as the fixed-point method or the Newton's method.

Remark 3.1. If one applies the fixed-point iteration method to find the solution $\boldsymbol{\alpha}$ of (3.5), then the evaluation of the advection field defined in (2.30) is required at each iteration procedure. On the other hand, the Newton's method for (3.5) requires the evaluations of (2.30) and (2.32) at each iteration scheme.

3.2. Iteration free second-order method. We begin this subsection with the discussion of an Euler's polygon to modify the error correction strategy recently developed by the authors [13, 14]. First, we consider the nonlinear ODE (3.1) with $\mathbf{x} = \mathbf{x}_{i,j}$. For simplicity, let us define $\boldsymbol{\pi}_{i,j}(t) := \boldsymbol{\pi}(t_{m+1}, \mathbf{x}_{i,j}; t)$. Applying the Taylor's expansion for $\boldsymbol{\pi}_{i,j}(t)$ about t_{m+1} and for $\mathbf{U}(t_{m+1}, \boldsymbol{\pi}_{i,j}(t_{m+1}))$ about t_m leads to

$$\begin{aligned} \boldsymbol{\pi}_{i,j}(t) &= \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{U}(t_{m+1}, \boldsymbol{\pi}_{i,j}(t_{m+1})) + \mathcal{O}(h^2) \\ &= \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{U}(t_m, \boldsymbol{\pi}_{i,j}(t_m)) + \mathcal{O}(h^2). \end{aligned} \tag{3.6}$$

Since $\boldsymbol{\pi}_{i,j}(t_m)$ is unknown, some modifications are required to get a known Euler's polygon. Let us assume that $\mathbf{x}_{i,j}^m$ is an approximation of $\boldsymbol{\pi}_{i,j}(t_m)$ satisfying

$$\boldsymbol{\pi}_{i,j}(t_m) - \mathbf{x}_{i,j}^m = \mathcal{O}(h^2). \tag{3.7}$$

Then, from (3.6) and (3.7), one may define the modified Euler's polygon $\mathbf{y}_{i,j}(t)$ by

$$\mathbf{y}_{i,j}(t) := \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{U}(t_m, \mathbf{x}_{i,j}^m). \tag{3.8}$$

Combining (3.6), (3.7), and (3.8) yields

$$\boldsymbol{\psi}_{i,j}(t) := \boldsymbol{\pi}_{i,j}(t) - \mathbf{y}_{i,j}(t) = \mathcal{O}(h^2). \tag{3.9}$$

By differentiating both sides of (3.9) and applying the Taylor's expansion into the result together with (3.1), one gets an asymptotically first-order ODE given by

$$\begin{aligned}\boldsymbol{\psi}'_{i,j}(t) &= \mathbf{U}(t, \boldsymbol{\psi}_{i,j}(t) + \mathbf{y}_{i,j}(t)) - \mathbf{y}'_{i,j}(t) \\ &= \mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}_{i,j}(t_m))\boldsymbol{\psi}_{i,j}(t) + \mathbf{U}(t, \mathbf{y}_{i,j}(t)) - \mathbf{U}(t_m, \mathbf{x}^m_{i,j}) + \mathcal{O}(h^3), \quad t \le t_{m+1},\end{aligned} \tag{3.10}$$

where $\mathbf{U}_{\boldsymbol{\pi}}(t, \mathbf{y}_{i,j}(t))$ denotes the Jacobian matrix defined in (2.32). For detailed derivation, we refer to the papers [13, 14].

Notice that the first equation of (3.10) is known as the defect differential equation and one may use this equation to get the error correction term together with (3.9). But, it may give highly nonlinear difficulty from the self-consistent constraint for the advection field. Whereas, the reduced asymptotic equation (3.10) is a linear one together with the unknown function $\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}_{i,j}(t_m))$. Since the solution of (3.10) is corresponding to the error of the Euler's method, an approximation for $\boldsymbol{\pi}_{i,j}(t)$ obtained by (3.9) can be regarded as an error correction. In this sense, we would like to call the proposed method an ECM rather than the defect correction method. Note that if the problem (3.1) is stiff, then so is the problem (3.10). Hence, as a numerical scheme of (3.10), we apply the midpoint rule, which is known as an A-stable method.

The fact $\mathbf{y}_{i,j}(t_{m+1}) = \boldsymbol{\pi}_{i,j}(t_{m+1})$ implies $\boldsymbol{\psi}_{i,j}(t_{m+1}) = 0$. Thus, by integrating (3.10) over the interval $[t_{m-1}, t_{m+1}]$ and then applying the midpoint integration rule, one may have an asymptotic formula such that

$$\begin{aligned}-\boldsymbol{\psi}_{i,j}(t_{m-1}) &= 2h\Big(\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}_{i,j}(t_m))\boldsymbol{\psi}_{i,j}(t_m) \\ &\qquad + \mathbf{U}(t_m, \mathbf{y}_{i,j}(t_m)) - \mathbf{U}(t_m, \mathbf{x}^m_{i,j})\Big) + \mathcal{O}(h^3) \\ &= h\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}_{i,j}(t_m))\boldsymbol{\psi}_{i,j}(t_{m-1}) \\ &\qquad + 2h\Big(\mathbf{U}(t_m, \mathbf{y}_{i,j}(t_m)) - \mathbf{U}(t_m, \mathbf{x}^m_{i,j})\Big) + \mathcal{O}(h^3).\end{aligned} \tag{3.11}$$

From (2.30) and (3.8), we approximate $\mathbf{y}_{ij}(t_m)$ with $\mathbf{y}^m_{i,j} := \mathbf{x}_{i,j} - h\mathbf{U}^m(\mathbf{x}^m_{i,j})$ and substitute it into (3.11). Then, (3.11) can be rewritten by

$$\Big(\mathcal{I} + h\mathcal{J}^m(\mathbf{y}^m_{i,j})\Big)\boldsymbol{\psi}_{i,j}(t_{m-1}) = 2h\Big(\mathbf{U}^m(\mathbf{x}^m_{i,j}) - \mathbf{U}^m(\mathbf{y}^m_{i,j})\Big) + \epsilon_{i,j}, \tag{3.12}$$

where $\mathbf{U}^m(\mathbf{y}^m_{i,j})$ and $\mathcal{J}^m(\mathbf{y}^m_{i,j})$ are defined by (2.30) and (2.32), respectively, and the truncation term $\epsilon_{i,j}$ is given by

$$\begin{aligned}\epsilon_{i,j} &= h\Big(\mathcal{J}^m(\mathbf{y}^m_{i,j}) - \mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}_{i,j}(t_m))\Big)\boldsymbol{\psi}_{i,j}(t_{m-1}) \\ &\qquad + 2h\Big(\mathbf{U}^m(\mathbf{y}^m_{i,j}) - \mathbf{U}(t_m, \mathbf{y}_{i,j}(t_m)) - \mathbf{U}^m(\mathbf{x}^m_{i,j}) + \mathbf{U}(t_m, \mathbf{x}^m_{i,j})\Big) + \mathcal{O}(h^3).\end{aligned} \tag{3.13}$$

Thus, by combining (3.12) with (3.8) and (3.9), one may get an asymptotic formula for $\boldsymbol{\pi}_{i,j}(t_{m-1})$ given by

$$\begin{aligned}\boldsymbol{\pi}_{i,j}(t_{m-1}) &= \boldsymbol{\pi}^{m-1}_{i,j} + \mathcal{O}(\epsilon_{i,j}), \\ \boldsymbol{\pi}^{m-1}_{i,j} &:= \mathbf{x}_{i,j} - 2h\mathbf{U}^m(\mathbf{x}^m_{i,j}) + 2h\Big(\mathcal{I} + h\mathcal{J}^m(\mathbf{y}^m_{i,j})\Big)^{-1}\Big(\mathbf{U}^m(\mathbf{x}^m_{i,j}) - \mathbf{U}^m(\mathbf{y}^m_{i,j})\Big).\end{aligned} \tag{3.14}$$

Remark 3.2. For the calculation of $\boldsymbol{\pi}^{m-1}_{i,j}$ defined by (3.14), we remark that the associated two approximated advection fields and one Jacobian matrix for two cubic

interpolations at different positions defined by (2.30) and (2.32) are required. It is comparable to the second-order iteration method discussed in the previous subsection in the sense of computational cost.

Combining the approximations $\boldsymbol{\pi}_{i,j}^{m-1}$ defined by (3.14) with the formula (1.5) yields an asymptotic formula for $\rho(t_{m+1}, \mathbf{x}_{i,j})$, which is given by

$$\begin{aligned}\rho(t_{m+1}, \mathbf{x}_{i,j}) &= \rho(t_{m-1}, \boldsymbol{\pi}_{i,j}(t_{m-1})) = \rho(t_{m-1}, \boldsymbol{\pi}_{i,j}^{m-1} + \mathcal{O}(\epsilon_{i,j})) \\ &= \rho_{i,j}^{m+1} + \rho(t_{m-1}, \boldsymbol{\pi}_{i,j}^{m-1}) - \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}_{i,j}^{m-1}) + \mathcal{O}(\epsilon_{i,j}),\end{aligned} \tag{3.15}$$

where $\rho_{i,j}^{m+1}$ is defined by

$$\rho_{i,j}^{m+1} := \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}_{i,j}^{m-1}). \tag{3.16}$$

We close the section by summarizing and presenting the pseudocode for the proposed BSLM based on ECM as follows:

ALGORITHM ECM-BSL $(\rho_0(x_1, x_2), t_0, T, x_{1,min}, x_{1,max}, x_{2,min}, x_{2,max}, h, \Delta x_1, \Delta x_2)$. The algorithm is capable of solving nonlinear advection equation (1.1)–(1.3). The approximate values are saved at each time level.

1. Discretize the time and spatial domain as (2.1) and evaluate the initial solution $\rho_{i,j}^0$ at time t_0 on the grid point $\mathbf{x}_{i,j} = (x_{1,min} + i\Delta x_1, x_{2,min} + j\Delta x_2)$.
2. Let $t_1 := t_0 + h$.
3. The approximate solution $\rho_{i,j}^1$ at time t_1 on the grid point $\mathbf{x}_{i,j}$ is computed with the backward Euler scheme instead of the midpoint rule.
4. Let $t_2 := t_1 + h$.
5. If $t_2 > T$, then exit.
6. Solve the linear system (2.10) to approximate the potential for the given values $\rho_{i,j}^1$ of $\rho(t_1, \mathbf{x}_{i,j})$ and then approximate the advection field $\mathbf{U}(t_1, \mathbf{x}_{i,j})$ at time $t = t_1$ at grid point $\mathbf{x}_{i,j}$ with the formula (2.30).
7. Calculate $\boldsymbol{\pi}_{i,j}^0$ from the formula (3.14) with the advection $\mathbf{U}(t_1, \mathbf{x}_{i,j})$.
8. Calculate the approximations $\rho_{i,j}^2$ with the formula (3.16) for $\rho(t_2, \mathbf{x}_{i,j})$ and save these values.
9. Set $t_1 := t_2$, $\rho_{i,j}^0 := \rho_{i,j}^1$, $\rho_{i,j}^1 := \rho_{i,j}^2$ and go to step 4.

4. Error analysis. This section aims to analyze the convergence for the proposed approximation scheme. To do this, consider the nonlinear ODE (3.1) with an arbitrary arriving point $\mathbf{x} \in \Omega$ (therefore, we do not specify (i, j) indices hereafter) and let us define $\boldsymbol{\pi}(t) := \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t)$. Analogous to the scheme discussed in the previous section, one may define the modified Euler's polygon $\mathbf{y}(t)$ as follows:

$$\mathbf{y}(t) = \mathbf{x} + (t - t_{m+1})\mathbf{U}(t_m, \mathbf{x}^m), \tag{4.1}$$

where $\mathbf{x}^m := \mathbf{x}^m(\mathbf{x})$ is an approximation to $\boldsymbol{\pi}(t_m)$ satisfying

$$\mathbf{x}^m = \boldsymbol{\pi}(t_m) + \mathcal{O}(h^2). \tag{4.2}$$

For the function $\mathbf{U}^m(\mathbf{x})$ defined by (2.30), we define $\mathbf{y}^m := \mathbf{y}^m(\mathbf{x}) := \mathbf{x} - h\mathbf{U}^m(\mathbf{x}^m)$. Then, (4.1) and Lemma 2.6 give

$$\|\mathbf{y}(t_m) - \mathbf{y}^m\|_2 = \mathcal{O}\Big(\frac{h}{\Delta x}\mathbf{E}^m + h\Delta x^3\Big). \tag{4.3}$$

Combining (4.1), (4.2), and $\boldsymbol{\pi}(t_{m+1}) = \mathbf{x}$ with the Taylor's expansion of $\boldsymbol{\pi}(t_{m+1})$ about t_m, one can see

$$\begin{aligned}\mathbf{y}(t_m) - \mathbf{x}^m &= \mathbf{x} - \big(\mathbf{x}^m + h\mathbf{U}(t_m, \mathbf{x}^m)\big) \\ &= \boldsymbol{\pi}(t_{m+1}) - \big(\boldsymbol{\pi}(t_m) + h\mathbf{U}(t_m, \boldsymbol{\pi}(t_m))\big) + \mathcal{O}(h^2) = \mathcal{O}(h^2).\end{aligned} \tag{4.4}$$

Hence, the Taylor's series expansion gives

$$\mathbf{U}(t_m, \mathbf{y}(t_m)) - \mathbf{U}(t_m, \mathbf{x}^m) = \mathcal{O}(h^2)\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{x}^m) + \mathcal{O}(h^4). \tag{4.5}$$

As in the case $\mathbf{x} = \mathbf{x}_{i,j}$ discussed in the previous section, one may get the approximate value $\boldsymbol{\pi}^{m-1}$ for the departure point $\boldsymbol{\pi}(t_{m-1})$ together with the truncation error $\epsilon(\mathbf{x})$ generally defined by

$$\epsilon(\mathbf{x}) := h\mathbf{A}^m(\mathbf{x})\boldsymbol{\psi}(t_{m-1}) + 2h\mathbf{B}^m(\mathbf{x}) + \mathcal{O}(h^3), \tag{4.6}$$

where $\boldsymbol{\psi}(t) := \boldsymbol{\pi}(t) - \mathbf{y}(t)$, which has the asymptotic behavior

$$\boldsymbol{\psi}(t) = \mathcal{O}(h^2), \tag{4.7}$$

and $\mathbf{A}^m(\mathbf{x})$ and $\mathbf{B}^m(\mathbf{x})$ are defined by

$$\begin{aligned}\mathbf{A}^m(\mathbf{x}) &:= \mathcal{J}^m(\mathbf{y}^m) - \mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}(t_m)), \\ \mathbf{B}^m(\mathbf{x}) &:= \mathbf{U}^m(\mathbf{y}^m) - \mathbf{U}(t_m, \mathbf{y}(t_m)) - \mathbf{U}^m(\mathbf{x}^m) + \mathbf{U}(t_m, \mathbf{x}^m).\end{aligned} \tag{4.8}$$

Combining (2.33), (4.3), and Taylor's series expansion, one can estimate $\mathbf{A}^m(\mathbf{x})$ defined in (4.8) as

$$\begin{aligned}\|\mathbf{A}^m(\cdot)\|_2 &\le \|\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}^m) - \mathcal{J}^m(\mathbf{y}^m)\|_2 + \|\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}(t_m)) - \mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{y}^m)\|_2 \\ &\le C\Big(\frac{\mathbf{E}^m}{\Delta x^2} + \Delta x^2\Big),\end{aligned} \tag{4.9}$$

where C is a constant independent of h and Δx. In order to estimate the term $\mathbf{B}^m(\mathbf{x})$, it is observed that three values $\mathbf{y}(t_m)$, $\mathbf{y}^m$, and $\mathbf{x}^m$ are sufficiently close from (4.3) and (4.4). In particular, we can assume that these three values are in a neighborhood of a fixed cell. Hence, by Taylor's series expansion and (2.34), the term $\mathbf{B}^m(\mathbf{x})$ can be written by

$$\begin{aligned}\mathbf{B}^m(\mathbf{x}) &= \mathbf{U}^m(\mathbf{y}^m) - \mathbf{U}^m(\mathbf{y}(t_m)) + \mathbf{U}^m(\mathbf{y}(t_m)) - \mathbf{U}^m(\mathbf{x}^m) \\ &\quad - \Big(\mathbf{U}(t_m, \mathbf{y}(t_m)) - \mathbf{U}(t_m, \mathbf{x}^m)\Big) \\ &= \Big(\mathbf{y}^m - \mathbf{y}(t_m)\Big)\mathcal{J}^m(\mathbf{y}^m) + \big(\mathbf{x}^m - \mathbf{y}(t_m)\big)\big(\mathbf{U}_{\boldsymbol{\pi}}(t_m, \mathbf{x}^m) - \mathcal{J}^m(\mathbf{x}^m)\big) \\ &\quad + \mathcal{O}\Big(\Delta x^3 + \Big(\mathbf{y}^m - \mathbf{y}(t_m)\Big)^2 + \big(\mathbf{x}^m - \mathbf{y}(t_m)\big)^2\Big).\end{aligned}$$

Thus, using (2.33) and (4.3), $\mathbf{B}^m(\mathbf{x})$ can be estimated by

$$\begin{aligned}\|\mathbf{B}^m(\cdot)\|_2 &\le C\left(\frac{h}{\Delta x}\mathbf{E}^m + h\Delta x^3 + \|\mathbf{y}(t_m) - \mathbf{x}^m\|_2\|\mathbf{U}_{\boldsymbol{\pi}}(\mathbf{x}^m) - \mathcal{J}^m(\mathbf{x}^m)\|_2\right) \\ &\le C\left(\left(\frac{h}{\Delta x} + \frac{h^2}{\Delta x^2}\right)\mathbf{E}^m + \Delta x^3 + h^2\Delta x^2\right).\end{aligned} \tag{4.10}$$

Summarizing (4.6), (4.7), (4.9), and (4.10) leads to the following lemma.

LEMMA 4.1. *The truncation term* $\epsilon(\mathbf{x})$ *defined by* (4.6) *satisfies*

$$\|\epsilon\|_2 = \mathcal{O}\Big(h\mathbf{E}^m + h\Delta x^3 + h^3\Big)$$

provided $\frac{h}{\Delta x} = \mathcal{O}(1)$.

For the approximate value $\boldsymbol{\pi}^{m-1}$, let $\rho^{m+1}(\mathbf{x})$ be the approximate value of $\rho(t_{m+1}, \mathbf{x})$ defined by

$$\rho^{m+1}(\mathbf{x}) := \mathcal{P}\boldsymbol{\rho}^{m+1}(\mathbf{x}) = \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}^{m-1}). \tag{4.11}$$

By combining (1.5) with Lemma 4.1 and using Taylor's expansion, the error $\mathbf{E}^{m+1}$ defined by (2.26) can be estimated by

$$\begin{aligned}
\mathbf{E}^{m+1} &= \sqrt{\int_\Omega \Big(\rho(t_{m+1}, \mathbf{x}) - \mathcal{P}\boldsymbol{\rho}^{m+1}(\mathbf{x})\Big)^2 d\mathbf{x}} \\
&= \sqrt{\begin{array}{l}\int_\Omega \Big(\rho(t_{m+1}, \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m+1})) - \mathcal{P}\boldsymbol{\rho}^{m+1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m+1}))\Big)^2 \\ \quad \times d\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m+1})\end{array}} \\
&= \sqrt{\begin{array}{l}\int_\Omega \Big(\rho(t_{m-1}, \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})) - \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}) - \mathcal{O}(\epsilon(\mathbf{x})))\Big)^2 \\ \quad \times J(t_{m+1}, \mathbf{x}; t_{m-1}) d\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})\end{array}} \\
&\le \sqrt{\begin{array}{l}\int_\Omega \Big(\rho(t_{m-1}, \boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})) - \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}))\Big)^2 \\ \quad \times d\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})\end{array}} \\
&\quad + \sqrt{\begin{array}{l}\int_\Omega \Big(\mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})) - \mathcal{P}\boldsymbol{\rho}^{m-1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}) - \mathcal{O}(\epsilon(\mathbf{x})))\Big)^2 \\ \quad \times d\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1})\end{array}} \\
&\le \mathbf{E}^{m-1} + \mathcal{O}(\gamma_1 \mathbf{E}^m + \gamma_2),
\end{aligned}$$

where $\gamma_1 = \mathcal{O}(h)$ and $\gamma_2 = \mathcal{O}(h^3 + h\Delta x^3 + \Delta x^4)$. Here, the term Δx^4 in γ_2 is included to consider the truncation error from the interpolation. That is, we have

$$\mathbf{E}^{m+1} \le \mathbf{E}^{m-1} + \gamma_1 \mathbf{E}^m + \gamma_2. \tag{4.12}$$

The difference relation (4.12) can be solved as follows.

THEOREM 4.2. *Assume* $\mathbf{E}^0 := \mathcal{O}(\Delta x^4)$ *and* $\mathbf{E}^1 = \mathcal{O}(h^2 + h\Delta x^3 + \Delta x^4)$. *Then, the error* $\mathbf{E}^m := \|\rho(t_m, \mathbf{x}) - \mathcal{P}\boldsymbol{\rho}^m(\mathbf{x})\|_2$ *satisfies*

$$\mathbf{E}^m = \mathcal{O}\Big(h^2 + \Delta x^3 + \frac{\Delta x^4}{h}\Big). \tag{4.13}$$

Proof. At first, we note that the inequality (4.12) can be written by a homogeneous form

$$\Big(\mathbf{E}^{m+1} + \gamma_3\Big) \le \Big(\mathbf{E}^{m-1} + \gamma_3\Big) + \gamma_1\Big(\mathbf{E}^m + \gamma_3\Big), \tag{4.14}$$

where γ_3 is given by

$$\gamma_3 = \frac{\gamma_2}{\gamma_1} = \mathcal{O}\Big(h^2 + \Delta x^3 + \frac{\Delta x^4}{h}\Big). \tag{4.15}$$

From the characteristic equation of (4.14), one can compute its roots as follows:

$$r_1 := \frac{\gamma_1 - \sqrt{\gamma_1^2+4}}{2}, \quad r_2 := \frac{\gamma_1 + \sqrt{\gamma_1^2+4}}{2}. \tag{4.16}$$

Hence, one may see that

$$\mathbf{E}^m + \gamma_3 \le A r_1^m + B r_2^m, \tag{4.17}$$

where the constants A and B satisfy

(4.18)

$$A = \frac{\gamma_3}{\sqrt{\gamma_1^2+4}}\left(\frac{\sqrt{\gamma_1^2+4}-\gamma_1}{2} - 1\right) = \mathcal{O}(\gamma_3), \qquad B = \frac{\gamma_3}{\sqrt{\gamma_1^2+4}}\left(\frac{\sqrt{\gamma_1^2+4}+\gamma_1}{2} + 1\right) = \mathcal{O}(\gamma_3).$$

Due to $|r_1| < 1$, the first part of the right-hand side of inequality in (4.17) is $\mathcal{O}(\gamma_3)$. So we now need to claim that there is a constant C independent of h such that $r_2^m \le C$ for sufficiently large m. Indeed, one can see that the definition of r_2 defined by (4.16) gives

$$\left(\frac{\gamma_1 + \sqrt{\gamma_1^2+4}}{2}\right)^m = \left(1 + \frac{2\gamma_1}{2 - \gamma_1 + \sqrt{\gamma_1^2+4}}\right)^m \le (1+\gamma_1)^m \le \left(1 + C_1\frac{T}{m}\right)^m \le \exp(C_1 T) := C$$

for sufficiently large m and a constant C_1 independent of h. Therefore, combining (4.18) with (4.15) and (4.17) gives

$$\mathbf{E}^m = \mathcal{O}(\gamma_3).$$

Thus, from the asymptotic behaviors of γ_3, one may complete the proof. ∎

Let $\widetilde{\boldsymbol{\rho}}^m$ be an analytic solution vector defined by

$$\widetilde{\boldsymbol{\rho}}^m := \left[\rho(t_m, \mathbf{x}_{0,0}), \rho(t_m, \mathbf{x}_{1,0}), \ldots, \rho(t_m, \mathbf{x}_{\widetilde{N}_1-1,0}), \ldots, \rho(t_m, \mathbf{x}_{\widetilde{N}_1-1,\widetilde{N}_2-1})\right]^T$$

and $\widetilde{\mathbf{E}}^m$ be the error defined by

$$\widetilde{\mathbf{E}}^m := \|\mathcal{P}\widetilde{\boldsymbol{\rho}}^m(\mathbf{x}) - \mathcal{P}\boldsymbol{\rho}^m(\mathbf{x})\|_{l_2,\widetilde{\Delta}x}, \tag{4.19}$$

where $\widetilde{N} \ge N$ and $\widetilde{\Delta}x := \max_k(\frac{L_k}{\widetilde{N}_k})$. Then, by using the triangle inequality, (2.19), and Theorem 4.2, one may have

$$\widetilde{\mathbf{E}}^m = \mathcal{O}\left(h^2 + \Delta x^3 + \frac{\Delta x^4}{h}\right). \tag{4.20}$$

5. Numerical tests. In this section, we simulate the guiding center problem defined on the computational domain $(0, 4\pi) \times (0, 2\pi)$ with the periodic boundary condition on both x_1 and x_2 directions. The initial condition is assumed to be

$$\rho(t=0, \mathbf{x}) = \sin(x_2) + 0.015\cos(kx_1),$$

TABLE 1

The error $Err(t,h,N)$ and convergence rates at time $t=5$, $t=10$, and $t=20$ obtained by employing the proposed method with local cubic interpolation.

Method	h	$Err(5,h,512)$	Rate	$Err(10,h,512)$	Rate	$Err(20,h,512)$	Rate
ECM-1	$\frac{1}{2}$	4.59e-2	-	2.45e-1	-	1.48e-0	-
	$\frac{1}{4}$	1.08e-2	2.09	5.73e-2	2.09	4.13e-1	1.85
	$\frac{1}{8}$	2.63e-3	2.04	1.39e-2	2.04	1.03e-1	2.0
	$\frac{1}{16}$	6.25e-4	2.08	3.31e-3	2.08	2.56e-2	2.01
	$\frac{1}{32}$	1.25e-4	2.32	6.61e-4	2.32	1.23e-2	1.06
ECM-2	$\frac{1}{2}$	4.59e-2	-	2.45e-1	-	1.48e-0	-
	$\frac{1}{4}$	1.08e-2	2.09	5.73e-2	2.09	4.13e-1	1.85
	$\frac{1}{8}$	2.63e-3	2.04	1.39e-2	2.04	1.03e-1	2.0
	$\frac{1}{16}$	6.25e-4	2.08	3.31e-3	2.08	2.56e-2	2.01
	$\frac{1}{32}$	1.25e-4	2.32	6.61e-4	2.32	1.23e-2	1.06

where $k=\frac{2\pi}{L_1}$ and $L_1=x_{1,max}-x_{1,min}$ is the domain size in x_1-direction.

In order to support the theoretical analysis, we evaluate the error $Err(t,h,N)$ defined as

$$Err(t_m,h,N):=\widetilde{\mathbf{E}}^m=\|\mathcal{P}\widetilde{\boldsymbol{\rho}}^m(\mathbf{x})-\mathcal{P}\boldsymbol{\rho}^m(\mathbf{x})\|_{l_2,\widetilde{\Delta}x}, \tag{5.1}$$

where $\mathcal{P}\boldsymbol{\rho}^m$ is the interpolation operator constructed with approximation solutions $\{\rho^m(\mathbf{x}_{i,j})\}_{i=0,j=0}^{N,N}$ and $\mathcal{P}\widetilde{\boldsymbol{\rho}}^m(\mathbf{x})$ is a reference solution obtained with the finest grid resolution $\widetilde{N}=\widetilde{N}_1=\widetilde{N}_2=1024$ and smallest time step $h=\frac{1}{64}$. Here, we use $N=N_1=N_2$. The approximations of $\mathbf{x}_{i,j}^m$ in (3.7) can be done in many different ways. In this paper, we consider the following two choices. The first one is to construct $\mathbf{x}_{i,j}^m$ as follows:

$$\mathbf{x}_{i,j}^m:=\begin{cases}\mathbf{x}_{0,0}, & i=j=0,\\ \mathbf{y}_{i-1,j}(t_m)\ \left(\text{or}\ \ \mathbf{y}_{i,j-1}(t_m)\right) & \text{otherwise,}\end{cases} \tag{5.2}$$

where $\mathbf{y}_{i,j}(t)$ is a modified Euler's polygon defined in (3.8). The other one is to set $\mathbf{x}_{i,j}^m$ as

$$\mathbf{x}_{i,j}^m:=\mathbf{x}_{i,j}-h\mathbf{U}(t_m,\mathbf{x}_{i,j}). \tag{5.3}$$

The proposed schemes corresponding to the choice of (5.2) and (5.3) are denoted as ECM-1 and ECM-2, respectively. For a numerical comparison, we adopt the iteration free method proposed by McGregor [17], which can be stated as follows:

$$\boldsymbol{\pi}(t_{m+1},\mathbf{x};t_m)=\mathbf{x}-h\mathbf{U}(t_{m+1},\mathbf{x})+\frac{h^2}{2}\mathbf{U}_{\boldsymbol{\pi}}(t_{m+1},\mathbf{x})\mathbf{U}(t_{m+1},\mathbf{x})+\mathcal{O}(h^3),$$

where

$$\mathbf{U}(t_{m+1},\mathbf{x})=\frac{3}{2}\mathbf{U}(t_m,\mathbf{x})-\frac{1}{2}\mathbf{U}(t_{m-1},\mathbf{x})+\mathcal{O}(h^2).$$

In Table 1, we numerically estimate the temporal convergence of the proposed schemes at time $t=5,10,20$ by varying the time step sizes $h=2^{-k}$, $k=1,2,\ldots,5$, with the fixed spatial grid resolution $N_1=N_2=512$. From the figures of Table 1,

TABLE 2
The error $Err(t, h, N)$ and convergence rates at time $t = 5$, $t = 10$, and $t = 20$ obtained by employing the proposed method with local cubic polynomial interpolation.

Method	N	$Err(5, \frac{1}{64}, N)$	Rate	$Err(10, \frac{1}{64}, N)$	Rate	$Err(20, \frac{1}{64}, N)$	Rate
ECM-1	32	8.47e-4	-	7.06e-3	-	8.26e-1	-
	64	1.06e-4	3	9.28e-4	2.93	3.43e-1	1.27
	128	1.23e-5	3.1	1.09e-4	3.09	1.41e-1	1.29
	256	1.23e-6	3.32	1.13e-5	3.27	5.17e-2	1.45
	512	9.31e-8	3.72	7.86e-7	3.85	1.34e-2	1.95
ECM-2	32	8.70e-4	-	7.22e-3	-	8.31e-1	-
	64	1.06e-4	3.03	9.33e-4	2.95	3.44e-1	1.27
	128	1.23e-5	3.11	1.09e-4	3.09	1.41e-1	1.29
	256	1.23e-6	3.33	1.13e-5	3.28	5.17e-2	1.45
	512	9.30e-8	3.72	7.84e-7	3.85	1.34e-2	1.95

one can readily confirm that the proposed schemes have the second-order numerical convergence in time, which guarantees the theoretical convergence analysis. In order to check the convergence order in space, Table 2 lists the numerical results which are obtained with the spatial resolutions $N_1 = N_2 = 2^{4+k}$, $k = 1, 2, \ldots, 5$, and fixed time step size $h = \frac{1}{64}$. As shown in the figures, the proposed schemes have the numerical order of convergence at least 3 in space. However, the numerical results failed to support the theoretical analysis of the proposed schemes when $t = 20$. In Figure 1, we can see that the reference solution develops steep gradients with increasing time ($t \geq 20$), which can cause spurious spatial oscillations and therefore deviations from the theoretical analysis.

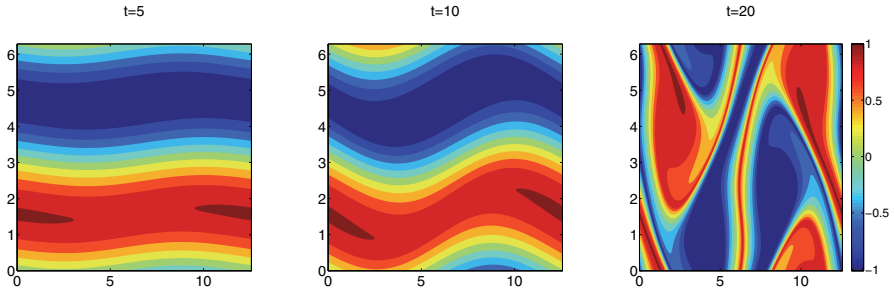

FIG. 1. *Reference solution plots of the guiding center problem at time $t = 5, 10$, and 20.*

Next, we compare the numerical results obtained by applying the conventional second-order leap-frog method, the iteration free method proposed by McGregor (McG-BSL), and the proposed methods (ECM-1 and ECM-2). For the nonlinear equations arising from the leap-frog method, we employ the fixed point iteration (Mid-Fixed) and Newton iteration (Mid-Newton) techniques. To ensure the convergence of the iterations, we perform two sets of simulations with different tolerance levels, $tol = 10^{-6}$ and $tol = 10^{-8}$, and compare the results. We examine the conservation

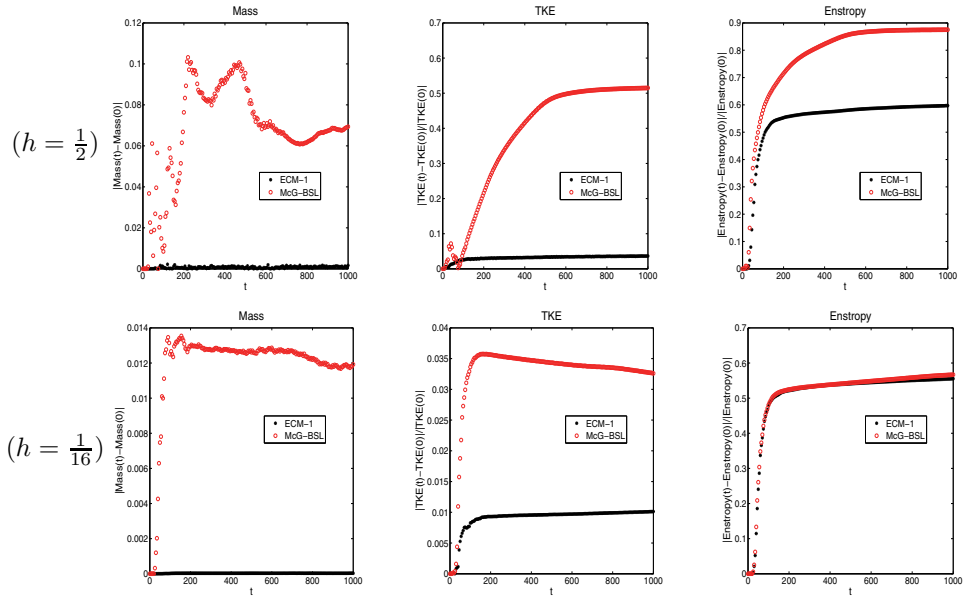

FIG. 2. *Comparisons of the conservation of mass, total kinetic energy, and enstrophy among two iteration free methods when* $N_1 = N_2 = 512$.

properties of the total mass, kinetic energy, and enstrophy defined as

$$\frac{d}{dt}\,\mathrm{Mass}(t) = \frac{d}{dt}\Big(\int_\Omega \rho(t,x,y)dxdy\Big) = 0,$$
$$\frac{d}{dt}\,\mathrm{TKE}(t) = \frac{d}{dt}\Big(\int_\Omega \big(U_1^2(t,x,y) + U_2^2(t,x,y)\big)dxdy\Big) = 0,$$
$$\frac{d}{dt}\,\mathrm{Enstrophy}(t) = \frac{d}{dt}\Big(\int_\Omega \rho(t,x,y)^2 dxdy\Big) = 0,$$

which are very useful tools for testing and comparison of the performance of numerical algorithms.

Figure 2 shows the time variations of the three quantities obtained by the two iteration free methods (McG-BSL and ECM-1) during the simulations with large time step $h = \frac{1}{2}$ and small time step $h = \frac{1}{16}$. As observed in the figure, the proposed scheme has better conservation properties than McG-BSL [17]. To investigate the conservation properties among the proposed method and two iteration methods, Figures 3 and 4 show the time variations of the three quantities during the simulations with different time step sizes $h = 2^{-k}$, $k = 1, 2, 3, 4$, when the tolerances are 10^{-6} and 10^{-8}, respectively.

As observed in the figures, the simulation results with different tolerance levels agree very well and suggest that the iterations are well converged. The three methods show similar level of conservation properties for the total kinetic energy and the enstrophy. Since the total kinetic energy depends on the spatial derivatives of the potential and the enstropy is mainly controlled by the spatial grid resolution and interpolation method, it is not surprising to see the similar results regarding the conservation of the two quantities. On the other hand, ECM-2 shows similar mass

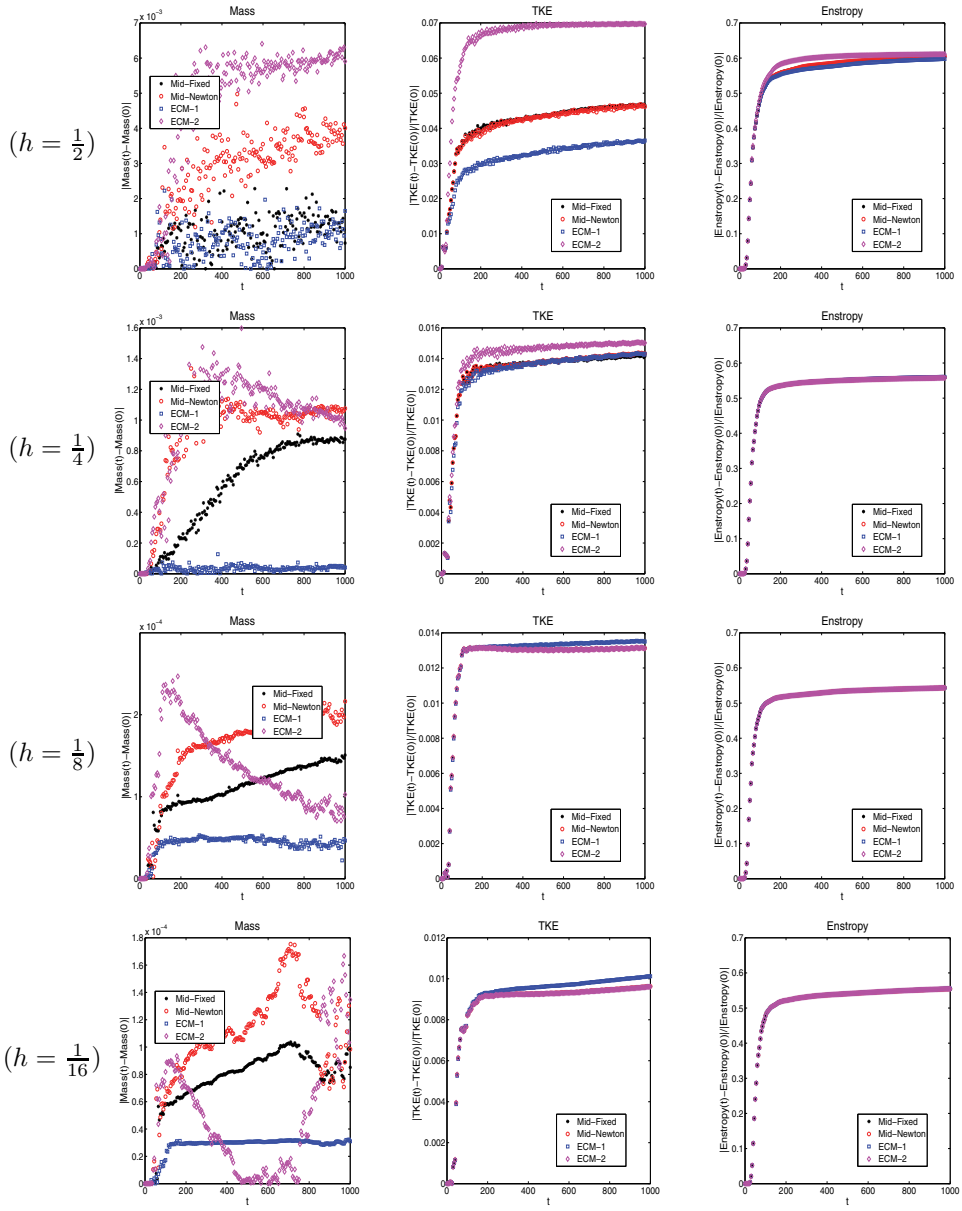

FIG. 3. *Comparisons of the conservation of mass, total kinetic energy, and enstrophy among the proposed schemes, fixed-point and Newton iteration methods, when* $N_1 = N_2 = 512, tol = 10^{-6}$.

conservation with iteration methods, while ECM-1 shows superior mass conservation compared to the other methods as the step size h becomes smaller. It is noteworthy that the conservation of the total mass is improved with the ratio h^2 for all three schemes, which suggests the accuracy of the time integration scheme is crucial for the conservation and the proposed scheme is more accurate compared to the conventional second schemes.

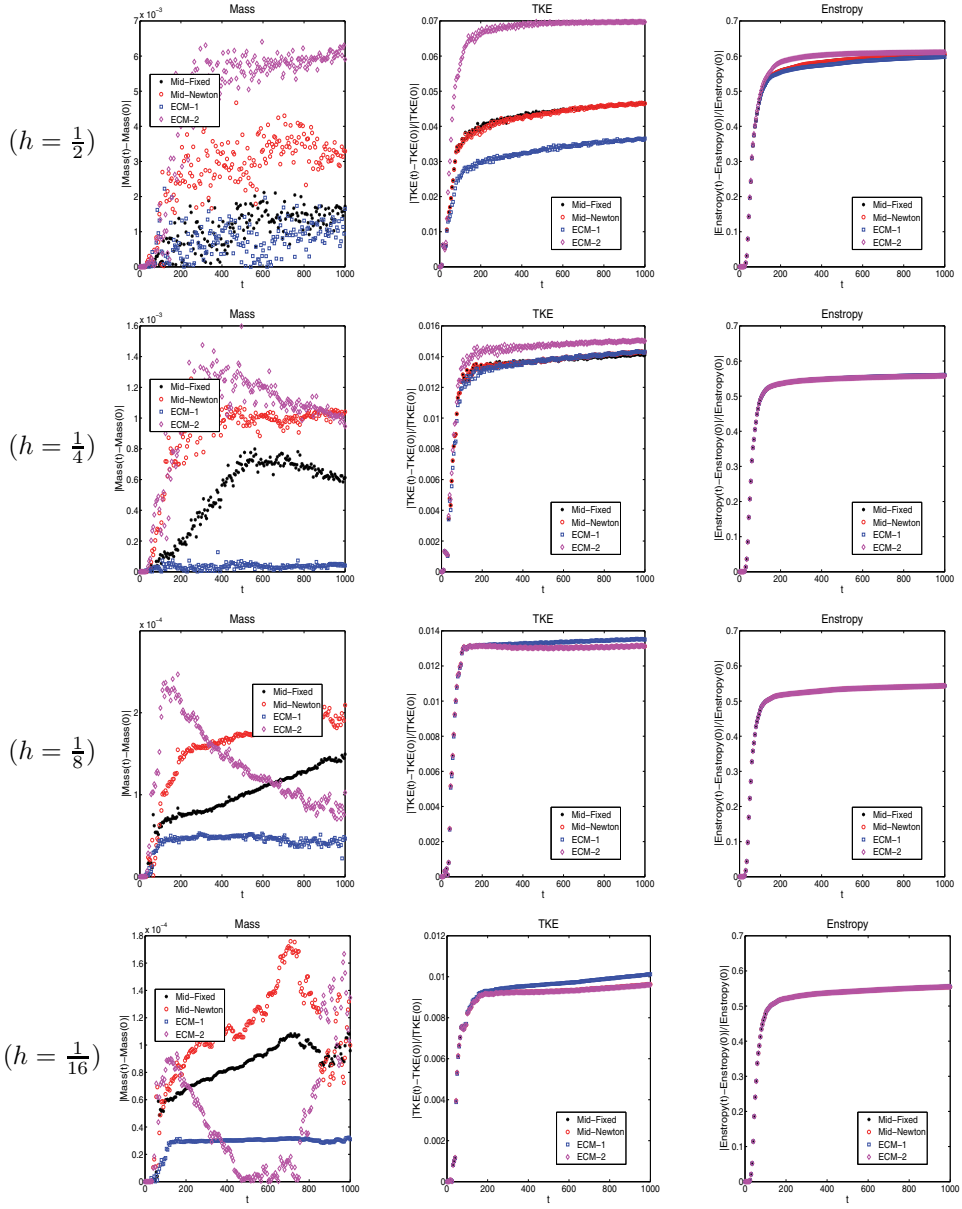

FIG. 4. *Comparisons of the conservation of mass, total kinetic energy, and enstrophy among the proposed schemes, fixed-point and Newton iteration methods, when* $N_1 = N_2 = 512, tol = 10^{-8}$.

In Figure 5, we also list and compare the computational costs of four different methods. One may see that the proposed iteration free methods are more efficient compared to the other two iteration methods. In particular, the iteration methods require more computational costs as the tolerance level becomes smaller.

To investigate further details of the simulation results, we compare the contour plots of $\rho(t, \mathbf{x})$. In Figures 6 and 7, the contour plots at time t = 60,200 slices are presented for the cases with difference choices of simulation methods and time step

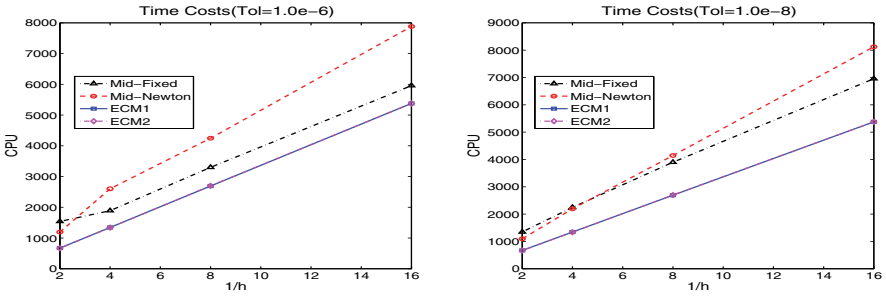

FIG. 5. *Comparisons of the required time costs among four scheme when* $N_1 = N_2 = 512, tol = 10^{-6}, 10^{-8}$, *and* $T = 1000$.

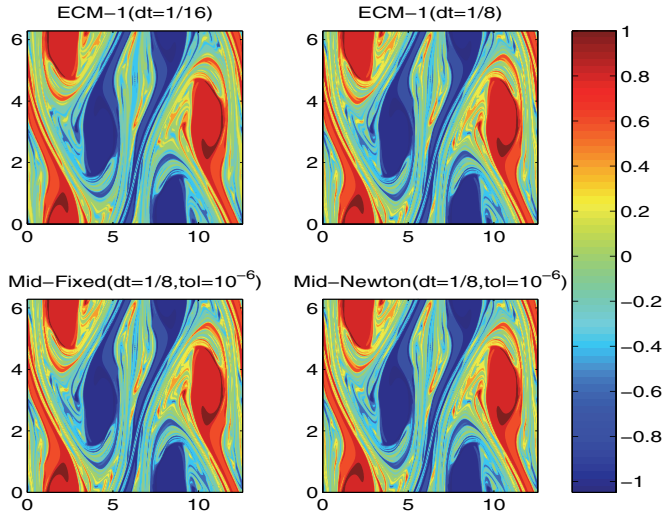

FIG. 6. *Phase space plots of the guiding center problem at time* $t = 60$ *when* $h = \frac{1}{8}, N_1 = N_2 = 512$.

size $h = 1/8$. As a reference, we put contour plots obtained by applying the proposed scheme (ECM-1) with $h = 1/16$ in the upper left of the figures. The spatial size and location of major vortices are very similar for different schemes at each time slice. In an early stage of the simulations, small-scale filamentary structures are developed around the major vortices, as we can see in Figure 6. As time goes on, these small-scale structures are subject to the rapid mixing driven by the self-consistent flows and smoothed out eventually (see Figure 7). We note that the active development and smoothing of the small-scale structures mainly occur during $t \leq 150$, which coincides with the time interval for the rapid accumulation of errors in the total kinetic energy conservation. This suggests that the total kinetic energy conservation is strongly linked to the spatial grid and interpolation method affecting the resolution of the small-scale structures as we discussed before.

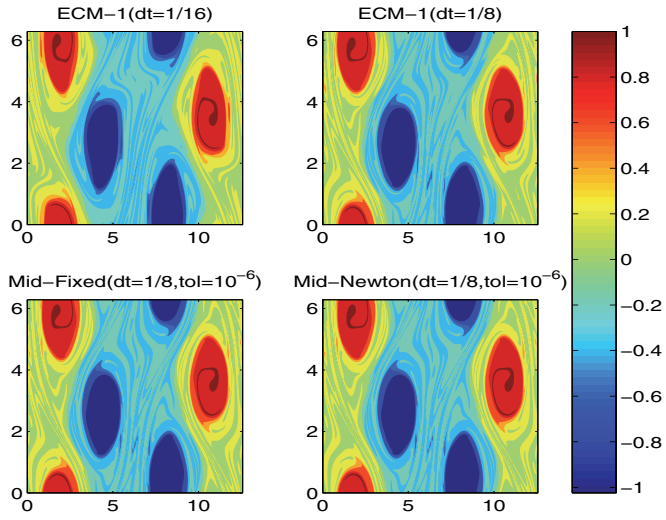

FIG. 7. *Phase space plots of the guiding center problem at time* $t = 200$ *when* $h = \frac{1}{8}, N_1 = N_2 = 512$.

6. Conclusion. We have developed an iteration free backward semi-Lagrangian method with the order of temporal convergence 2. The proposed method allowed us to perform faster simulations compared to the conventional second-order methods. We want to emphasize that the new method does not require the iteration procedures which are computationally costly and prone to error accumulation in long time simulation. As we confirmed in the previous section, the new methods show better conservation properties. This is significant to the prospect of long time simulation, which requires robust conservation of key physical quantities such as total mass, energy, and enstrophy, etc.

In future works, we plan to develop a higher-order (i.e., $n > 2$) time integration scheme based on the proposed second-order scheme. Like the construction of higher-order Runge–Kutta schemes based on the second order, the proposed scheme is expected to play a central role in the development. In this work, we primarily focused on the time discretization scheme and paid relatively little attention to the spatial discretization. However, as discussed in the previous section, there is evidence showing the importance of the spatial discretization and interpolation scheme to capture fine-scale spatial structures and minimize spatial oscillations. We will revisit the simulations with various interpolation techniques for better performance and optimization of simulation results in future.

REFERENCES

[1] J. A. CARRILLO AND F. VECIL, *Nonoscillatory interpolation methods applied to Vlasov-based models*, SIAM J. Sci. Comput., 29 (2007), pp. 1179–1206.

[2] G. H. COTTET AND P. A. RAVIART, *Particle methods for the one dimensional Vlasov-Poisson equations*, SIAM. J. Numer. Anal., 21 (1984), pp. 52–76.

[3] N. CROUSEILLES, G. LATU, AND E. SONNENDRÜCKER, *Hermite spline interpolation on patches for parallelly solving the Vlasov-Poisson equation*, Int. J. Appl. Math. Comput. Sci., 17 (2007), pp. 335–349.

[4] P. J. Davis, *Circulant Matrices*, Wiley-Interscience, New York, 1979.
[5] B. Fornberg, *A Practical Guide to Psedospectral Methods*, Cambridge University Press, Cambridge, UK, 1996.
[6] P. G. D. Sastre and R. Bermejo, *Error analysis for hp-FEM semi-Lagrangian second order BDF method for convection-diffusion problems,* J. Sci. Comput., 49 (2011), pp. 211–237.
[7] V. Grandgirard, Y. Sarazin, P. Angelino, A. Bottino, N. Crouseilles, G. Darmet, G. Dif-Pradalier, X. Garbet, Ph. Ghendrih, S. Jolliet, G. Latu, E. Sonnendrucker, and L. Villard, *Global full-f grokinetic simulations of plasma turbulence*, Plasma Phys. Control. Fusion, 49 (2007), pp. B173–B182.
[8] R. Hockney and J. Eastwood, *Computer Simulation Using Particles, IOP Publishing*, Bristol, UK, 1988.
[9] R. A. Horn and R. J. Chaples, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991,
[10] Y. Idomura, M. Ida, S. Tokuda, and L. Villard, *New conservative gyrokinetic full-f Vlasov code and its comparison to gyrokinetic δf particle-in-cell code*, J. Comput. Phys., 226 (2007), pp. 244–262.
[11] G. Jacobs and J. Hesthaven, *High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids*, J. Comput. Phys., 214 (2006), pp. 96–121.
[12] G. S. Jiang and C. W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
[13] P. Kim, X. Piao, and S. D. Kim, *An error corrected Euler method for solving stiff problems based on Chebyshev collocation*, SIAM J. Numer. Anal., 49 (2011) pp. 2211–2230.
[14] S. D. Kim, X. Piao, and P. Kim, *Convergence on error correction methods for solving initial value problems*, J. Comput. Appl. Math., 236 (2012), pp. 4448–4461.
[15] J. G. Liu and C. W. Shu, *A high-order discontinuous Galerkin method for 2D incompressible flows*, J. Comput. Phys., 126 (2000), pp. 577–596.
[16] X. D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
[17] J. L. McGregor, *Economical determination of departure points for semi-Lagrangian models*, Monthly Weather Rev., 121 (1992), pp. 221–230.
[18] S. E. Parker and W. W. Lee, *A fully nonlinear characteristic method for gyrokinetic simulation*, Phys. Fluids B, 5 (1993), pp. 77–86.
[19] J. M. Qiu and A. Christlieb, *A conservative high order semi-Lagrangian WENO method for the Vlasov-equation*, J. Comput. Phys., 229 (2010), pp. 1130–1149.
[20] J. M. Qiu and C. W. Shu, *Conservative high order semi-Lagrnagian finite difference WENO methods for advection in incompressible flow*, J. Comput. Phys., 230 (2011), pp. 863–889.
[21] K. Sebastian and C. W. Shu, *Multidomain WENO finite difference method with interpolation at subdomain interfaces*, J. Sci. Comput., 19 (2003), pp. 405–438.
[22] C. W. Shu and S. Osher, *Efficient implementation of essnsically non-oscillatory shock capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[23] C. W. Shu and S. Osher, *Efficient implementation of essnsically non-oscillatory shock capturing schemes* II, J. Comput. Phys., 83 (1989), pp. 32–78.
[24] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo, *The semi-Lagrangian method for the numerical resolution of Vlasov equations*, J. Comput. Phys., 149 (1999), pp. 201–220.