Communications on Applied Mathematics and Computation (2023) 5:116–142
https://doi.org/10.1007/s42967-021-00150-5

ORIGINAL PAPER

High-Order Semi-Lagrangian WENO Schemes Based on Non-polynomial Space for the Vlasov Equation

Andrew Christlieb[1] · **Matthew Link**[1] · **Hyoseon Yang**[1] · **Ruimeng Chang**[2]

Received: 30 January 2021 / Revised: 10 June 2021 / Accepted: 14 June 2021 /
Published online: 24 August 2021

Abstract
In this paper, we present a semi-Lagrangian (SL) method based on a non-polynomial function space for solving the Vlasov equation. We find that a non-polynomial function based scheme is suitable to the specifics of the target problems. To address issues that arise in phase space models of plasma problems, we develop a weighted essentially non-oscillatory (WENO) scheme using trigonometric polynomials. In particular, the non-polynomial WENO method is able to achieve improved accuracy near sharp gradients or discontinuities. Moreover, to obtain a high-order of accuracy in not only space but also time, it is proposed to apply a high-order splitting scheme in time. We aim to introduce the entire SL algorithm with high-order splitting in time and high-order WENO reconstruction in space to solve the Vlasov-Poisson system. Some numerical experiments are presented to demonstrate robustness of the proposed method in having a high-order of convergence and in capturing non-smooth solutions. A key observation is that the method can capture phase structure that require twice the resolution with a polynomial based method. In 6D, this would represent a significant savings.

Keywords Semi-Lagrangian methods · WENO schemes · High-order splitting methods · Non-polynomial basis · Vlasov equation · Vlasov-Poisson system

Mathematics Subject Classification 35Q83 · 65D05 · 65D15 · 65M06 · 65M22

✉ Hyoseon Yang
hyoseon@msu.edu

Andrew Christlieb
christli@msu.edu

Matthew Link
linkmat1@msu.edu

Ruimeng Chang
ruimeng.chang17@student.xjtlu.edu.cn

[1] Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

[2] Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou 215123, Jiangsu, China

1 Introduction

Kinetic equations represent a fundamental class of models that describe the evolution of a distribution of particles. In particular, kinetic models describe a range of phenomenon we encounter, from collisionless gases through collision dominated flow. In collision dominated flow, the models reduce to dynamics described by fluid models. In the collisionless limit, kinetic models describe a class of dynamics known as kinetic effects, which in the class of plasma's includes: Landau damping, two-stream instability, bump-on-tail instability, etc. While kinetic models characterize a larger class of effects than fluid models, they are computationally much more demanding. What makes them more demanding is that they are naturally six dimensional (three space, three velocity) plus time, the fastest velocities greatly restrict the time step in explicit methods, and the models for inter-particle collisions can be very computationally demanding. In this paper, we examine non-polynomial semi-Lagrangian (SL) methods as a way of improving computability in the collisionless limit. Hence, as a prototypical model, this paper addresses a numerical scheme to solve the Vlasov-Poisson (VP) system of equations. In the tests presented in the results section, the method shows improved accuracy over polynomial based methods, suggesting that for the same accuracy, the non-polynomial method can work at effectively one half the resolution in each dimension of the polynomial based approach, which would be a nontrivial computational savings in 6D. In this paper we consider the 1D-1V VP model with mobile electrons and a fixed ion background:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E}(\mathbf{x}, t) \cdot \nabla_{\mathbf{v}} f = 0 \tag{1}$$

and

$$\mathbf{E}(\mathbf{x}, t) = -\nabla_{\mathbf{x}} \phi(\mathbf{x}, t), \tag{2a}$$

$$-\Delta_{\mathbf{x}} \phi(\mathbf{x}, t) = \rho(\mathbf{x}, t), \tag{2b}$$

where $\mathbf{x} \times \mathbf{v} \in \Omega_{\mathbf{x}} \times \Omega_{\mathbf{v}} \subset \mathbb{R}^m \times \mathbb{R}^m$. Here, $f(\mathbf{x}, \mathbf{v}, t)$ is a distribution function describing a probability of finding an electron with velocity $\mathbf{v}$ at position $\mathbf{x}$ and time t,

$$\rho(\mathbf{x}, t) = \int_{\Omega_{\mathbf{v}}} f(\mathbf{x}, \mathbf{v}, t) \mathrm{d}\mathbf{v} - 1 \tag{3}$$

is the charge density, $\mathbf{E}$ is the electric field, and ϕ is the electrostatic potential. The equation (1) can also be written in the following conservative form:

$$f_t + \nabla_{\mathbf{x}}(\mathbf{v} f) + \nabla_{\mathbf{v}}(\mathbf{E}(\mathbf{x}, t) f) = 0.$$

The simulations for solving the VP equations are typically performed using one of two classes of numerical methods. The one class is the Lagrangian approach, where phase space is represented as a collection of moving points tracing out characteristic trajectories [2, 6, 9, 10, 15, 17, 18, 22, 28, 33, 42, 46, 47, 49]. These include Lagrangian approaches that trace both forward or backward along un-split trajectories, often an adaptive representation of phase space and a mapping or remapping used to mitigate complexity. The most popular of these approaches, for both its conservative properties as well as its ease of implementation, is the particle-in-cell (PIC) methods [2, 9, 10, 28, 42, 46, 47, 49], which are based on tracing the motion of the plasma by macro-particles. More recently,

the energy conserving fully implicit PIC methods have been developed [9, 10, 32]. These methods are combined with symplectic integrators and lead to provably conservative methods. The other class of methods for solving the VP system is the Eulerian approach, which discretize the equations on a fixed grid of the phase space. The SL method is one of the most widely used algorithms among the Eulerian approaches. The SL method is a very attractive method for simulating kinetic equations, because it does not contain the Courant-Friedrichs-Lewy (CFL) condition unlike other Eulerian methods. Early works on using SL schemes for Vlasov systems were based on the splitting scheme of Cheng and Knorr [11]. Various schemes were invented but all of the schemes were found to be roughly equally accurate [20, 21, 27, 35, 44]. Later, SL schemes focused on creating high-order schemes and incorporating weighted essentially non-oscillatory (WENO) methods into the SL framework [29]. A pointwise WENO scheme was designed for handling interpolation for backward SL schemes [8]. The WENO was applied to the SL method by writing the scheme in a flux difference form [37, 39]. Later, a Hermite WENO (HWENO) scheme was tried with the flux difference form [7]. An alternative SL high-order approach came from combining the discontinuous Galerkin (DG) method with the SL framework to get SL DG methods for Vlasov models [40, 41]. In a forward SL framework, the convected scheme (CS), first proposed in [27], was extended to obtain arbitrarily high-order by correcting the fluxes [23]. The CS was further improved by incorporating WENO [43]. An alternate Eulerian approach involved the development of a fully implicit conservative DG approach on a fixed mesh [12, 13]. The approach used splitting to keep the cost of the implicit solves down. Alternative Eulerian SL methods have looked at wavelet based methods. The method proposed in [1] made use of an adaptive wavelet-based method for the Vlasov Maxwell system. More recently, Eulerian SL methods developed in [38, 51] bluer the lines between Eulerian and Lagrangian updates using un-split characteristics to advance between time levels of a fixed mesh.

In this work, we introduce a generalized WENO interpolation based on a non-polynomial function space to apply to an SL method. While this approach could be incorporated into un-split methods, such as thous in [38, 51], here we limit our attention to split SL methods. It has been observed that approximation schemes constructed based on a non-polynomial basis provide improved numerical solutions for a range of problems. We note that this strategy has been previously employed with WENO schemes. Zhu and Qiu [53, 54] developed trigonometric WENO methods and used them as limiters for Runge-Kutta DG methods. Ha et al. [24–26] suggested to use exponential functions based WENO and Christlieb et al. [14, 16] utilized exponential functions and radial basis functions as basis functions for WENO to have optimal convergence orders. Other examples include work in areas such as extended finite element methods (XFEMs). The XFEM has been developed by generalizing the solution spaces of finite element numerical schemes for treating discontinuities [4, 31, 34, 45] in problems such as crack propagation in solid mechanics.

To demonstrate the utility of the approach in plasma problems, we combined our non-polynomial WENO method with an SL based Vlasov solver using a fourth-order splitting method. In the original high-order SL WENO method [37], the Strang splitting scheme was used. The method is only second order of accuracy in time. However, this choice was made as it is often the spatial error that dominates in VP over the time error [37]. As noted in [23], when taking large time steps, high-order in time matters. In the papers [19, 23, 36, 48], the Vlasov equations are solved based on high-order symplectic integrators. Hence, we have also made use of high-order splitting schemes for the time integrator. In [3, 52], the authors presented high-order symplectic integrators, symmetric symplectic partitioned

Runge-Kutta (PRK) methods and Runge-Kutta-Nyström (RKN) methods. Here, we implement both the fourth- and sixth-order RKN methods.

The rest of the paper is organized as follows. In Sect. 2.1, we review SL schemes for solving 1D Vlasov equations and in Sect. 2.2, we formulate the novel SL scheme based on non-polynomial WENO schemes. The time splitting methods are reviewed in Sect. 3 and we suggest the high-order splitting SL algorithms in Sect. 4. Then, the numerical results are presented in Sect. 5.

2 Construction of SL WENO Based on Non-polynomial Space

In this paper, we use splitting methods by decomposing the equation (1). A simple first order in time splitting is as follows:

$$\begin{cases} \partial_t f + v\partial_x f = 0, \\ \partial_t f + E[f]\partial_v f = 0\,. \end{cases} \tag{4}$$

High-order time splitting is reviewed in Sect. 3. What is important to note is that this kind of splitting converts the system into one-dimensional uncoupled problems. That is, both equations from (4) are in the advection form so that they can be solved using a simple characteristic update, this is also true for high-order splitting. We solve these advection equations using SL, "characteristic update", method based a WENO reconstruction. Here, WENO reconstruction is an essentially non-oscillatory interpolate. By using WENO reconstruction, we are leveraging the WENO interpolate to construct the solution at the foot of the characteristics in a non-oscillatory manner. We now review this method and show how we develop a non-polynomial reconstruction.

In the rest of this section, we restrict our attention to a one-dimensional linear advection equation

$$\partial_t f + a\partial_x f = 0, \quad x \in \Omega \subset \mathbb{R}^1 \text{ and } t \geqslant 0. \tag{5}$$

For the appropriately defined function $f(x, t)$, we introduce the domain Ω uniformly distributed with the cells $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ and the cell size $\Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ for $j \in \mathbb{Z}$. The cell centers are denoted by $x_j = (x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})/2$ and we use the notation f_j for the function value at the node x_j, i.e., $f_j := f(x_j)$. We also notate the function values at time $t = t^n$ as f_j^n for $n \in \mathbb{N} \cup \{0\}$ when it is necessary to indicate specific n-th time level where $t^n = \sum_{k=1}^{n} \Delta t^k$ with timestep size $\Delta t^k := t^{k+1} - t^k$ for each $k \in \mathbb{N} \cup \{0\}$. For simplicity, we apply a fixed timestep size $\Delta t^k = \Delta t, \forall k$.

2.1 An SL Scheme

In the previous work [37], the authors proposed an SL method with high-order WENO reconstruction in space. The conservative WENO reconstruction for their method is based on classical WENO introduced by Jiang and Shu [29] and the scheme can be extended to arbitrarily high-order. In this subsection, we review the SL scheme based on fixed stencil interpolation from [37].

We first consider the case $0 \leqslant a \leqslant \frac{\Delta x}{2\Delta t}$ for the speed a from the target equation (5) and start with a fifth-order Lagrange reconstruction for the function f at $x \in [x_j, x_{j+\frac{1}{2}}]$:

$$Af(x) := \sum_{i=j-3}^{j+2} f_i \ell_i(x) \tag{6}$$

with a fifth-order Lagrange function as the basis

$$\ell_i(x) = \prod_{k=j-3, k\neq i}^{j+2} \frac{x - x_k}{x_i - x_k}. \tag{7}$$

Assuming we have $\{f_j^n\}$, the SL update for f at x_j from $t = t^n$ to $t = t^{n+1}$ is given by

$$f_j^{n+1}(x) = Af^n(\bar{x}), \tag{8}$$

where $\bar{x} = x_j - a\Delta t$ using the characteristics formula. If we define a transformation function by

$$\xi(x) = \frac{x - x_j}{x_{j-1} - x_j},$$

then the update (8) is expressed in conservative form as

$$f_j^{n+1} = f_j^n - \xi(\bar{x})\left(\hat{f}^n_{j+\frac{1}{2}}(\xi(\bar{x})) - \hat{f}^n_{j-\frac{1}{2}}(\xi(\bar{x}))\right), \tag{9}$$

where $\hat{f}^n_{j\pm\frac{1}{2}}(\xi)$ is a flux function to be defined. We note that

$$\xi(\bar{x}) = \frac{\bar{x} - x_j}{x_{j-1} - x_j} = a\frac{\Delta t}{\Delta x}$$

so that $0 \leqslant \xi(\bar{x}) \leqslant \frac{1}{2}$. Rearranging terms in the SL update from (6) to (9),

$$\begin{aligned} f_j^{n+1} &= Af^n(\bar{x}) = \sum_{i=j-3}^{j+2} f_i^n \ell_i(\bar{x}) \\ &= f_j^n - \xi(\bar{x})\left(\hat{f}^n_{j+\frac{1}{2}}(\xi(\bar{x})) - \hat{f}^n_{j-\frac{1}{2}}(\xi(\bar{x}))\right), \end{aligned}$$

so that it is in a conservative form which reveals the flux function to be

$$\hat{f}^n_{j-\frac{1}{2}}(\xi) = \left[f^n_{j-3}, \cdots, f^n_{j+1}\right] \cdot C_L \cdot \left[1, \cdots, \xi^4\right]^{\mathrm{T}}, \tag{10}$$

where

$$C_L = \begin{bmatrix} \frac{1}{30} & 0 & -\frac{1}{24} & 0 & \frac{1}{120} \\ -\frac{13}{60} & -\frac{1}{24} & \frac{1}{4} & \frac{1}{24} & -\frac{1}{30} \\ \frac{47}{60} & \frac{5}{8} & -\frac{1}{3} & -\frac{1}{8} & \frac{1}{20} \\ \frac{9}{20} & -\frac{5}{8} & \frac{1}{12} & \frac{1}{8} & -\frac{1}{30} \\ -\frac{1}{20} & \frac{1}{24} & \frac{1}{24} & -\frac{1}{24} & \frac{1}{120} \end{bmatrix}. \tag{11}$$

For the case $-\frac{\Delta x}{2\Delta t} \leqslant a \leqslant 0$, the SL update for f at x_j from $t = t^n$ to $t = t^{n+1}$ is obtained by

$$f_j^{n+1} = f_j^n - a\frac{\Delta t}{\Delta x}\left(\hat{f}^n_{j+\frac{1}{2}}\left(a\frac{\Delta t}{\Delta x}\right) - \hat{f}^n_{j-\frac{1}{2}}\left(a\frac{\Delta t}{\Delta x}\right)\right), \tag{12}$$

where the flux function is defined by

$$\hat{f}^n_{j+\frac{1}{2}}(\xi) = \left[f^n_{j-2}, \cdots, f^n_{j+2}\right] \cdot C_R \cdot \left[1, \cdots, \xi^4\right]^{\mathrm{T}} \tag{13}$$

with the coefficient matrix

$$C_R(i,j) = C_L(6-i,j) \quad \text{for } 1 \leqslant i,j \leqslant 5.$$

Finally we remark that the SL update in a conservative form keeps its conservation property with the following proposition developed in reference [37].

Proposition 1 *The SL scheme in equations* (9) *and* (12) *with flux functions* (10) *and* (13) *conserves the total mass if periodic boundary conditions are imposed.*

2.2 Non-polynomial-Based SL WENO

We observed the SL scheme can be written in a conservative form (9) and we can apply the WENO reconstruction for the flux functions which preserves the conservation property. In this section, we introduce the non-polynomial based WENO reconstruction to compute the flux functions in (10) and (13). In [24], the authors introduced a sixth-order WENO scheme based on the exponential polynomial space. We modify this scheme to a fifth-order numerical method based on the trigonometric polynomial space and apply this method to approximate the flux function.

We consider the reconstruction of $\hat{f}_{j-\frac{1}{2}}(x)$ defined in (10) from (9) especially when $0 \leqslant a \leqslant \frac{\Delta x}{2\Delta t}$. For the opposite wind directional case, the numerical flux is obtained via the mirror-symmetric procedure with respect to x_j. The flux function $\hat{f}_{j-\frac{1}{2}}$ in (10) can be rewritten with respect to the variable ξ with a matrix C_L in (11):

$$\begin{aligned}
\hat{f}_{j-\frac{1}{2}}(\xi) = \;& \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot C_L \cdot \left[1, \cdots, \xi^4\right]^{\mathrm{T}} \\
= \;& \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot C_L(:,1) \cdot 1 \\
& + \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot C_L(:,2) \cdot \xi \\
& + \cdots \\
& + \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot C_L(:,5) \cdot \xi^4,
\end{aligned}$$

then we apply the WENO procedure for the zeroth order coefficients, that is, for the coefficients of $\xi^0 = 1$. In this section, the following will be developed in turn.

(i) Choose the non-polynomial basis to be used in interpolation process.
(ii) Compute the flux function based on the non-polynomial basis from Step (i) with fifth-order accuracy which will be a global solution for comparison.

(iii) We then construct three third-order overset approximations that compute the same flux as in Step 2. As in standard WENO, we compute the linear combination of the local approximations with linear weights that is equivalent to the form of Step (ii).
(iv) Next, we find the nonlinear weights using linear weights from Step (iii) for WENO procedure by defining smoothness indicators.
(v) Apply non-polynomial WENO to reconstruct the numerical fluxes.

First, we set the basis function to construct the numerical flux function other than Lagrange polynomials (7). We define a non-algebraic polynomial based function space Γ_5 by

$$\Gamma_5 := \text{span}\{\phi_i : i = 1, \cdots, 5\} \tag{14}$$

with functions ϕ_i and if the space Γ_5 constitutes an *extended Tchebysheff system* on $\mathbb{R}$, i.e.,

$$\det(\phi_i(s_j) : i, j = 1, \cdots, 5) \neq 0 \tag{15}$$

for any five-point stencil $\{s_j : j = 1, \cdots, 5\}$, the non-singularity of the interpolation matrix in (15) is guaranteed [30]. Given function values of f at a five-point stencil $\mathbf{S}_5 := \{x_{j-3}, \cdots, x_{j+1}\}$, the numerical flux based on Γ_5 is derived by the linear combination of $\phi_1, \cdots, \phi_5$ as follows. Let $\{\varphi_0, \cdots, \varphi_5\}$ be a set of non-algebraic polynomials satisfying $\Gamma_5 = \text{span}\{\varphi_0', \cdots, \varphi_5'\}$. In this paper, we introduce the set

$$\{\varphi_0, \cdots, \varphi_5\} := \{1, \sin \lambda x, \cos \lambda x, x^3, x^4, x^5\}, \tag{16}$$

so that the basis function space will be

$$\Gamma_5 = \text{span}\{\cos \lambda x, \sin \lambda x, x^2, x^3, x^4\},$$

which satisfies (15). Here the shape parameter λ is simply chosen to be $\lambda = 1/\Delta x$ for our experiments in Sect. 5, but it can be set to help the numerical scheme have optimal approximation order [16, 26].

To approximate the flux function $\hat{f}_{j-\frac{1}{2}}$, we define a function h implicitly through

$$f(x) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(s)\mathrm{d}s \tag{17}$$

which is obviously approximated by the numerical flux $\hat{f}$ such that

$$\left.\frac{\partial f}{\partial x}\right|_{x=x_j} = \frac{h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}}}{\Delta x} \approx \frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x}.$$

It is obvious that the cell average of h at x_ℓ, say $\bar{h}(x_\ell)$, is given from the definition (17) and the supposition of the problem:

$$\bar{h}(x_\ell) = \frac{1}{\Delta x} \int_{x_{\ell-\frac{1}{2}}}^{x_{\ell+\frac{1}{2}}} h(s)\mathrm{d}s = f(x_\ell). \tag{18}$$

We define a primitive function H of the function h by

$$H(x) = \int_{x_{j-3-\frac{1}{2}}}^{x} h(s)\mathrm{d}s \tag{19}$$

of which point values of functions at $x_{i+\frac{1}{2}}$ are explicitly given by (18) as

$$H\left(x_{i+\frac{1}{2}}\right) = \Delta x \sum_{\ell=j-3}^{i} \bar{h}(x_\ell) = \Delta x \sum_{\ell=j-3}^{i} f_\ell \tag{20}$$

for $i = j-3, \cdots, j+2$. Then, we can find an approximant $AH(x)$ to $H(x)$ based on the set of basis $\{\varphi_n\}$ (16) of form

$$AH(x) := \sum_{n=0}^{5} \alpha_n \varphi_n(x) \tag{21}$$

which satisfies

$$AH\left(x_{i+\frac{1}{2}}\right) = H\left(x_{i+\frac{1}{2}}\right), \quad \forall i = j-3, \cdots, j+2, \tag{22}$$

so that

$$AH(x) = H(x) + \mathcal{O}\left(\Delta x^6\right) \tag{23}$$

in smooth region. Then, the derivative of $AH(x)$ approximates $H'(x)$ as

$$\begin{aligned}(AH)'(x) &= \sum_{n=1}^{5} \alpha_n \varphi_n'(x) = H'(x) + \mathcal{O}\left(\Delta x^5\right) \\ &= h(x) + \mathcal{O}(\Delta x^5)\end{aligned} \tag{24}$$

by the relation (23) and the definition (19), where the first equality of the equation (24) is satisfied by the choice of the basis function (16) such that $\varphi_0'(x) = (1)' = 0$. Solving the linear system (22) with (20) and differentiating the solution (21) at $x_{j-\frac{1}{2}}$ gives us the coefficients $\mathbf{C} = [c_1, \cdots, c_5]^{\mathrm{T}}$ such that

$$\begin{aligned}\left.\frac{\partial(AH)}{\partial x}\right|_{x=x_{j-\frac{1}{2}}} &= \sum_{n=1}^{5} \alpha_n \varphi_n'(x_{j-\frac{1}{2}}) \\ &= \sum_{n=1}^{5} c_n f_{j-4+n} = \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot \mathbf{C}.\end{aligned}$$

Therefore, we can obtain a numerical flux $\hat{f}_{j-\frac{1}{2}}$ defined on $\mathbf{S}_5$, say $\hat{f}_{j-\frac{1}{2}}^{\mathbf{S}_5}$ as

$$\begin{aligned}\hat{f}_{j-\frac{1}{2}}^{\mathbf{S}_5} &= \left[f_{j-3}, \cdots, f_{j+1}\right] \cdot \mathbf{C} \\ &= h_{j-\frac{1}{2}} + \mathcal{O}\left(\Delta x^5\right),\end{aligned} \tag{25}$$

where the last equality is satisfied on the smooth region. The coefficients $\mathbf{C}$ is given in Appendix A. We remark that if we choose the basis function space Γ_5 (14) as a fifth-order algebraic polynomial function space, then $\mathbf{C} = C_L(:, 1)$ from (11).

Now, we subdivide $\mathbf{S}_5$ into three 3-point substencils $S_k = \{x_{j-3+k}, x_{j-2+k}, x_{j-1+k}\}$ for $k = 0, 1, 2$ and then in each substencil, the local numerical flux $\hat{f}^k_{j-1/2}$ which has third-order convergence can be computed similar to above:

$$\begin{aligned}\hat{f}^k_{j-\frac{1}{2}} &= \left[f_{j-3+k}, f_{j-2+k}, f_{j-1+k}\right] \cdot \mathbf{c}^k \\ &= h_{j-\frac{1}{2}} + \mathcal{O}\left(\Delta x^3\right)\end{aligned} \tag{26}$$

with the vectors of constants $\mathbf{c}^k = [c^k_1, c^k_2, c^k_3]^{\mathrm{T}}$ for $k = 0, 1, 2$, which are presented also in Appendix A. Then, we can find the linear weights $\{d_k : k = 0, 1, 2\}$ from (25) and (26) such that

$$\left[f_{j-3}, \cdots, f_{j+1}\right] \cdot \mathbf{C} = \sum_{k=0}^{2} d_k \left[f_{j-3+k}, f_{j-2+k}, f_{j-1+k}\right] \cdot \left[c^k_1, c^k_2, c^k_3\right]^{\mathrm{T}} + \mathcal{O}\left(\Delta x^5\right).$$

We note that once the basis set Γ_5 is chosen, d_k can be obtained by a direct calculation:

$$\begin{cases} d_0 = \mathbf{C}(1)/c^0_1, \\ d_1 = (\mathbf{C}(2) - d_0 c^0_1)/c^1_1, \\ d_2 = 1 - d_0 - d_1. \end{cases} \tag{27}$$

To introduce nonlinear weights for WENO construction, the smoothness indicator in each substencil can be defined by

$$\beta_k := \left|D^1_{S_k} f_j\right|^2 + \left|D^2_{S_k} f_j\right|^2, \tag{28}$$

with a p-th order undivided differential operator $D^p_{S_k}$ on each stencil S_k for $k = 0, 1, 2$, and the global smoothness indicator τ defined as

$$\tau = (\beta_0 - \beta_2)^2.$$

Then, nonlinear weights ω_k are defined as

$$\omega_k = \tilde{\omega}_k / \sum_{s=0}^{2} \tilde{\omega}_s \quad \text{and} \quad \tilde{\omega}_k = d_k \left(1 + \frac{\tau}{(\epsilon + \beta_k)^2}\right), \tag{29}$$

where $\epsilon > 0$ is a small number introduced to avoid the denominator becoming zero.

Finally, we construct the final numerical flux to be used in SL update based on nonlinear weights (29) as

$$\hat{f}^n_{j-\frac{1}{2}}(\xi) = \left[f^n_{j-3}, \cdots, f^n_{j+1}\right] \cdot \tilde{C}_L \cdot \left[1, \cdots, \xi^4\right]^{\mathrm{T}}, \tag{30}$$

where the coefficient matrix $\tilde{C}_L$ is defined by

$$\begin{aligned}\tilde{C}_L(:,1) &= \sum_{k=0}^{2} \omega_k \mathbf{c}_k,\\ \tilde{C}_L(:,2) &= C_L(:,2),\\ &\vdots\\ \tilde{C}_L(:,5) &= C_L(:,5)\end{aligned}$$

with a matrix C_L in (11) and local coefficient vectors

$$\begin{aligned}\mathbf{c}_0 &= [c_1^0, c_2^0, c_3^0, 0, 0]^{\mathrm{T}},\\ \mathbf{c}_1 &= [0, c_1^1, c_2^1, c_3^1, 0]^{\mathrm{T}},\\ \mathbf{c}_2 &= [0, 0, c_1^2, c_2^2, c_3^2]^{\mathrm{T}},\end{aligned}$$

computed from (26).

We wrap this section up with verifying that the nonlinear weights ω_k from (29) make the proposed scheme to be convergent in fifth order of accuracy in smooth region.

Proposition 2 *Assume that the function f is smooth and the linear weights* $\{d_k\}$ *and nonlinear weights* $\{\omega_k\}$ *are defined as in* (27) *and* (29)*, respectively. Then, the following convergence property is obtained*:

$$\omega_k - d_k = \mathcal{O}(\Delta x^4)$$

for $k = 0, 1, 2$.

The proof of this proposition is direct from the definitions which is provided in Appendix B.

3 High-Order Operator Splitting Methods

In this section, we review some time splitting methods used in split SL solutions to the equation (1). A much longer review can be found in [23]. Here, we review both the well-known Strang splitting method as well as two high-order methods based on RKN methods. All these methods are used for time march of the decomposed system, such as the approach we are taking to the equation (1). These time marching methods are used with both the polynomial and non-polynomial SL formulations and when SL methods are being compared, the exact same time stepping method is used.

3.1 Strang Splitting Scheme

Here, we give a brief review for the Strang spliting method for the VP system [11, 37]. The solution for the Vlasov equation (1) is preserved along the solution of the split equations (4):

$$\partial_t f + v\partial_x f = 0, \tag{31a}$$

$$\partial_t f + E[f]\partial_v f = 0. \tag{31b}$$

The above first-order splitting with a simple staggering of the solves. Equation (31), endowed with a Poisson structure, is solved with the following Strang splitting procedure.

Procedure 1: Strang splitting update from $f^n(x, v)$ to $f^{n+1}(x, v)$

(i) Solve the equation (31a) from $f^n(x, v)$ with timestep size $\frac{\Delta t}{2}$:
$f^*(x, v) = f^n\left(x - \frac{\Delta t}{2}v, v\right)$.
(ii) Compute $\rho(x)$ by summing $f^*(x, v)$ over v and then solve for the electric field $E^*(x)$ at $t = t^n + \frac{\Delta t}{2}$.
(iii) Solve the equation (31b) from $f^*(x, v)$ with timestep size Δt:
$f^{**}(x, v) = f^*(x, v - \Delta t E^*(x))$.
(iv) Solve the equation (31a) from $f^{**}(x, v)$ with timestep size $\frac{\Delta t}{2}$:
$f^{n+1}(x, v) = f^{**}\left(x - \frac{\Delta t}{2}v, v\right)$.

3.2 High-Order Splitting Method

As we develop high-order space schemes, we want to implement high-order splitting methods. In [3, 19], the authors presented fourth- and sixth order RKN methods.

The solution to equation (31) can be updated by

$$f(x, v) = f_0(x - tv, v), \quad f(x, v) = f_0(x, v - tE[f_0])$$

for a given initial condition $f_0(x, v) = f(x, v, t)$ so that it is naturally generalized. For $s > 0$, the splitting scheme with $2s + 1$ stages is defined by the relations

$$g_1(x, v) = f_0(x - b_0 v\Delta t, v), \tag{32a}$$

$$g_{2j}(x, v) = g_{2j-1}(x, v - b_{2j-1}E[g_{2j-1}](x)\Delta t), \tag{32b}$$

$$g_{2j+1}(x, v) = g_{2j}(x - b_{2j}v\Delta t, v), \tag{32c}$$

for $j = 1, \cdots, 2s$ where $f_1(x, v) := f(x, v, t + \Delta t)$ will be updated by $g_{2s+1}(x, v)$. For example, if we set $b_0 = b_2 = 0.5$ and $b_1 = 1$ with $s = 1$, the splitting procedure (32) with a Poisson equation gives us the Strang scheme introduced in Sect. 3.1. In Table 1, we review some optimized coefficients for the cases $s = 6, 11$ developed in [3].

Table 1 RKN coefficients

• $s = 6$ **(Order 4)** :
$[b_0, b_2, \cdots, b_{12}] = [0.083\,0, 0.396, -0.039\,1, 0.120, -0.039\,1, 0.396, 0.083\,0]$,
$[b_1, b_3, \cdots, b_{11}] = [0.245, 0.605, -0.350, -0.350, 0.605, 0.245]$.
• $s = 11$ **(Order 6)** :
$[b_0, b_2, \cdots, b_{22}] = [0.041\,5, 0.198, -0.04, 0.075\,3, -0.011\,5, 0.237, 0.237, -0.011\,5, 0.075\,3, -0.04, 0.198, 0.041\,5]$,
$[b_1, b_3, \cdots, b_{21}] = [0.123, 0.291, -0.127, -0.246, 0.357, 0.205, 0.357, -0.246, -0.127, 0.291, 0.123]$.

Here, the first three digits of the coefficients are given for simplicity but it is recommended to use sufficient digits in high-order accurate numerical schemes, see [3].

4 New Schemes for Solving VP Equation

In what follows, we suggest the scheme for solving VP equation (1) with (2) using the SL WENO scheme based on non-polynomial space from Sect. 2 with high-order splitting methods introduced in Sect. 3. Here, the proposed SL scheme based on non-polynomial WENO procedure is referred to as SL-WENO-NP methods.

4.1 Algorithm for VP

We summarize the algorithm to solve VP system using the SL-WENO-NP scheme in the following algorithm flowchart.

Algorithm 1: SL-WENO method based on non-polynomial space with high-order splitting

Solve (1) until the final time $t = T$. Let $\{b_i : i = 0, \cdots, 2s+1\}$ be given as in Table 1 for s. Denote the physical solution at the n-th time step $t = t^n$ by $f^n = f(x(t^n), v(t^n))$. Start with $n = 0$ and $t^0 = 0$.

While $t < T$, given f^n and Δt^n at $t = t^n$,

(i). Using the SL-WENO-NP scheme, perform a time step $b_0\Delta t^n$ shift along the x-axis (see (32a)):

$\tilde{f}^0(x, v) = f^n(x - (b_0\Delta t^n)v, v)$.

For $i = 1, \cdots, s$,

(ii) Compute the electric field $E(x, t)$ at time $t = t^n + b_{2i-2}\Delta t^n$ by substituting $\tilde{f}^{2i-2}$ from the above step in the Poisson equation (2) with (3):

$\nabla^2\phi(x, t) = \int \mathrm{d}v\tilde{f}^{2i-2}(x, v, t) - 1$.

(iii) Perform a time step $b_{2i-1}\Delta t^n$ along the v-axis using the SL-WENO-NP scheme (see (32b)):

$\tilde{f}^{2i-1}(x, v) = \tilde{f}^{2i-2}(x, v - (b_{2i-1}\Delta t^n)E[\tilde{f}^{2i-2}](x))$.

(iv) Perform a time step $b_{2i}\Delta t^n$ shift along the x-axis by applying SL-WENO-NP scheme (see (32c)):

$\tilde{f}^{2i}(x, v) = \tilde{f}^{2i-1}(x - (b_{2i}\Delta t^n)v, v)$.

Finally $f^{n+1}(x, v) := \tilde{f}^{2s}(x, v)$ is obtained. If $t = t^n + \Delta t^n < T$,

(v) Set the timestep size

$$\Delta t^{n+1} = \mathrm{CFL}\min\left(\frac{\mathrm{d}x}{\sup|v|}, \frac{\mathrm{d}v}{\sup|E|}\right) \quad \text{or} \quad \Delta t^{n+1} = \Delta x$$

depends on the test problems and here, if $t = t^n + \Delta t^{n+1} > T$, set $\Delta t^{n+1} = T - t^n$. With $t = t^{n+1}(= t^n + \Delta t^{n+1})$, go to Step **(i)** and execute the process.

Otherwise, we finish this algorithm.

4.2 Poisson Solvers

We have seen that the Poisson equation (2) is solved after each x-advection stage from Algorithm 1. There are various ways for solving the Poison equation to calculate the electric field. We used the standard central difference scheme, implemented through the Thomas algorithm, to solve the Poisson equation. The electric field is then computed by

taking the central difference of the potential. Other implementations such as using a fast Fourier transform (FFT) exist for solving for the electric field [23].

We briefly review our implementation with the Thomas algorithm here. The central difference scheme for the Poisson equation (2b)

$$-\Delta\phi = \rho$$

on the periodic spatial domain $x \in [a, b] \subset \mathbb{R}^1$ can be written as

$$-(\phi_{i+1} - 2\phi_i + \phi_{i-1}) = \Delta x^2 \rho_i, \tag{33}$$

when we discretize the domain $a = x_0 < x_1 < \cdots < x_N = b$ with a uniform cell size $\Delta x = x_i - x_{i-1}$ and say $\rho_i := \rho(x_i)$ and $\phi_i := \phi(x_i)$ for $i = 0, \cdots, N$.

We are solving the Poisson equation (33) with periodic boundary conditions which, in matrix form, gives

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 1 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \dots & 0 \\ & & \ddots & \ddots & \\ 1 & 0 & \dots & 1 & -2 \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{bmatrix} = -\Delta x^2 \begin{bmatrix} \rho_0 \\ \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{bmatrix}.$$

Since we are interested in the electric field, not the potential, we can add any constant to the potential. We choose to subtract ϕ_{N-1} from each ϕ_i and produce the resulting system

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \dots & 0 \\ & & \ddots & \ddots & \\ 1 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{bmatrix} = -\Delta x^2 \begin{bmatrix} \rho_0 \\ \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{bmatrix}. \tag{34}$$

It is shown that the strategy above is spectrally accurate by Boyd [5]. Solving the linear system (34), the algorithm for solving the potential is given by

$$\begin{cases} \phi_0 = -\dfrac{\Delta x^2}{N} \displaystyle\sum_{j=0}^{N-1} (j+1)\rho_j, \\ \phi_1 = -\Delta x^2 \rho_0 + 2\phi_0, \\ \phi_j = -\Delta x^2 \rho_{j-1} + 2\phi_{j-1} - \phi_{j-2} \qquad \text{for } j = 2, \cdots, N. \end{cases} \tag{35}$$

Now that we have potentials from (35), we can compute the electric field E from the equations (2a) by using appropriate differential operators. In the following numerical experiments, the second-order finite difference is employed but we remark that it is simple to make this fourth order or sixth order.

We simply provide here for the fourth-order central scheme for the Poisson equation:

$$-\frac{1}{12\Delta x^2}(-\phi_{i+2} + 16\phi_{i+1} - 30\phi_i + 16\phi_{i-1} - \phi_{i-2}) = \rho_i,$$

which can be solved in matrix form

$$\begin{bmatrix} -30 & 16 & -1 & 0 & 0 & \dots & 16 \\ 16 & -30 & 16 & -1 & 0 & \dots & -1 \\ -1 & 16 & -30 & 16 & 1 & \dots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 16 & -1 & 0 & \dots & -1 & 16 & -30 \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{bmatrix} = -12\Delta x^2 \begin{bmatrix} \rho_0 \\ \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{bmatrix}.$$

Similarly, we subtract ϕ_{N-1} from each ϕ_i then the resulting system is invertible.

5 Numerical Results

In this section, we provide some experimental results to demonstrate the performance of the proposed scheme described in Algorithm 1. In numerical experiments, we set the timestep size $\Delta t = \Delta x$ except for the refinement studies where the CFL number is set to be 1.2 and 2 to verify the convergence order of the proposed scheme with larger CFL numbers. In addition, the numerical results from the proposed method are compared with the SL with the classical WENO-based scheme [37]. Both schemes are performed with fourth-order time splitting methods using RKN coefficients [3].

5.1 Scalar Test Problems

First, we consider scalar test problems to verify the ability of the SL-WENO-NP schemes.

Example 1 (One-dimensional linear transport) Consider the one-dimensional linear equation

$$u_t + u_x = 0 \tag{36}$$

with a smooth initial function

$$u(x, 0) = \sin\left(\pi x + \frac{1}{\pi}\cos(\pi x)\right) \tag{37}$$

on periodic domain $[-1, 1]$. We deliberately chose a more complex initial function and computed the solution until the time $T = 2$. The numerical errors and convergence orders in L_∞ and L_1 norms are shown in Table 2 for the cases with two CFL numbers, 1.2 and 2. In both cases, we can observe that the fifth-order convergence in L_∞ and L_1 norms.

Example 2 (Two-dimensional linear transport) Consider the two-dimensional linear equation

$$u_t + u_x + u_y = 0. \tag{38}$$

We check the convergence order of the proposed scheme with a smooth initial function

$$u(x, y, 0) = \sin(\pi(x + y)) \tag{39}$$

Table 2 (Example 1) L_∞ and L_1-errors and orders of accuracy for (36) with (37) at $T = 2$

CFL	N_x	L_∞		L_1	
		Error	Order	Error	Order
1.2	20	1.206 1E−02	–	4.672 8E−03	–
	40	7.808 5E−04	3.95	2.143 0E−04	4.45
	80	3.172 2E−05	4.62	7.067 9E−06	4.92
	160	1.037 8E−06	4.93	2.243 3E−07	4.98
	320	3.302 3E−08	4.97	7.035 0E−09	4.99
2	20	1.844 1E−01	–	6.422 8E−02	–
	40	2.306 2E−02	3.00	1.363 7E−03	5.56
	80	1.078 6E−04	7.74	8.284 0E−06	7.36
	160	1.440 4E−06	6.23	2.084 9E−07	5.31
	320	3.085 2E−08	5.54	6.238 0E−09	5.06

on a periodic domain $[-1, 1] \times [-1, 1]$. The L_∞ and L_1 errors and approximation orders for the numerical solutions at time $T = 2$ are given in the Table 3. Clearly, fifth order of accuracy is observed for the cases with CFL= 1.2 and 2.

Example 5.3 (Rigid body rotation) Consider the two-dimensional equation

$$u_t - yu_x + xu_y = 0. \tag{40}$$

First with a smooth initial function

$$u(x, y, 0) = \exp\left(-x^2 - y^2\right) \tag{41}$$

on periodic domain $[-2\pi, 2\pi] \times [-2\pi, 2\pi]$, we examine the refinement study. In Table 4, we present both the L_∞ and L_1 numerical errors and convergence orders of the proposed

Table 3 (Example 2) L_∞ and L_1-errors and orders of accuracy for (38) with (39) at $T = 2$

CFL	$N_x \times N_y$	L_∞		L_1	
		Error	Order	Error	Order
1.2	20×20	4.331 9E−03	–	2.664 9E−03	–
	40×40	1.608 1E−04	4.75	8.462 1E−05	4.98
	80×80	5.195 8E−06	4.95	2.645 0E−06	5.00
	160×160	1.597 2E−07	5.02	8.260 5E−08	5.00
	320×320	4.898 4E−09	5.03	2.579 8E−09	5.00
2	20×20	4.663 6E−03	–	2.646 8E−03	–
	40×40	1.631 7E−04	4.84	8.171 9E−05	5.02
	80×80	5.188 2E−06	4.97	2.510 5E−06	5.02
	160×160	1.586 6E−07	5.03	7.837 3E−08	5.00
	320×320	4.765 1E−09	5.06	2.447 4E−09	5.00

Table 4 (Example 3) L_∞ and L_1 -errors and orders of accuracy for (40) with (41) at $T = 2\pi$

CFL	$N_x \times N_y$	L_∞		L_1	
		Error	Order	Error	Order
1.2	18×18	2.128 7E−01	–	6.578 1E−03	–
	36×36	3.195 1E−02	2.74	7.160 5E−04	3.20
	72×72	2.292 4E−03	3.80	4.439 1E−05	4.01
	144×144	8.638 4E−05	4.73	1.848 1E−06	4.59
	288×288	2.791 0E−06	4.95	6.330 5E−08	4.87
2	18×18	2.157 2E−01	–	6.639 2E−03	–
	36×36	3.217 5E−02	2.75	7.148 0E−04	3.22
	72×72	2.289 3E−03	3.81	4.455 7E−05	4.00
	144×144	8.630 6E−05	4.73	1.847 5E−06	4.59
	288×288	2.787 9E−06	4.95	6.323 7E−08	4.87

scheme. The fifth order of accuracy is observed not only when using the CFL number 1.2 but also when the CFL is 2.

Second, we apply the following discontinuous initial function:

$$u(x, y, 0) = \begin{cases} 1, & \text{if } (x, y) \in [-1, 1] \times [-4, 4] \cup [-4, 4] \times [-1, 1], \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

We solve the equation up to $T = 2\pi$. The numerical results with 90×90 grids and 60×60 grids are shown in Figs. 1 and 2, respectively. In both cases, the proposed scheme preforms better particularly for the edge of the rigid body than the previous scheme.

5.2 The VP System

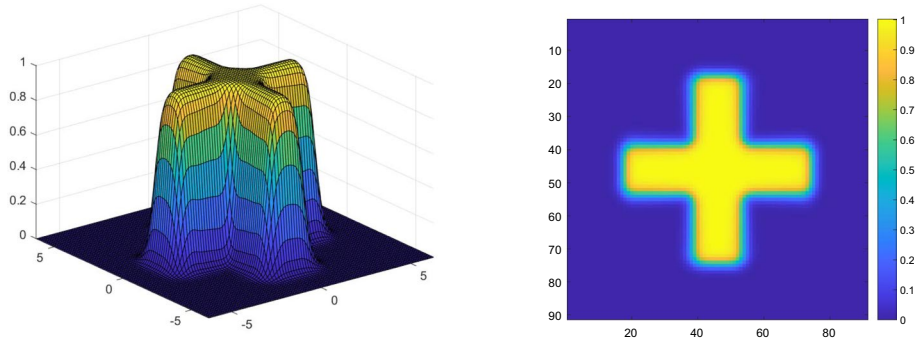
In this section, we consider the 1D-1V VP model (1) and (2) with initial functions $f_0(x, v) = f(x, v, t = 0)$. Periodic boundary conditions are imposed to x-domain $[a, b]$ and zero boundary conditions are imposed to v-domain $[-V, V]$ for all test examples.

Example 4 (Weak/strong Landau damping) We simulate the well-known Landau damping problem which is the collisionless damping of electrostatic waves in a plasma. This case has been computed by many authors [11]. Here, we test our VP problem on the initial conditions

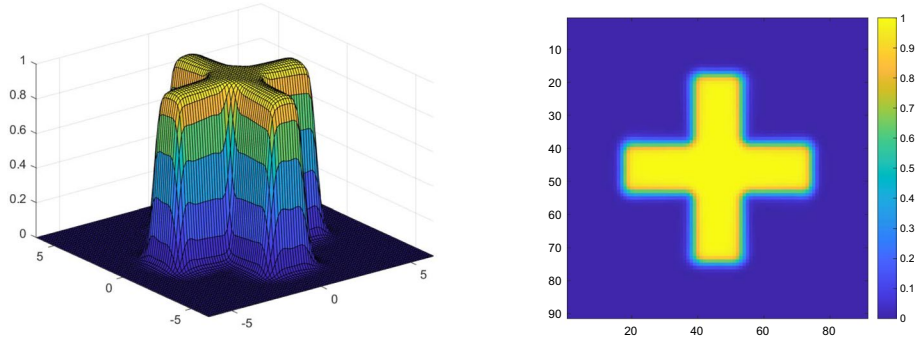
$$f_0(x, v) = (1 + \alpha \cos(kx)) \frac{e^{-v^2/2}}{\sqrt{2\pi}},$$

where α is the strength of the perturbation and $k = 0.5$. The domain boundaries are $a = 0, b = 4\pi$ and $V = 5$ in x and v space, respectively.

We choose a small perturbation $\alpha = 0.01$ first which will give us weak Landau damping. The L_2 norms of the electric fields with 64×128 and 128×256 grids are shown in Fig. 3 with the theoretical damping rate $\gamma = -0.153\,3$ (in the solid yellow line), and the L_1 norms,

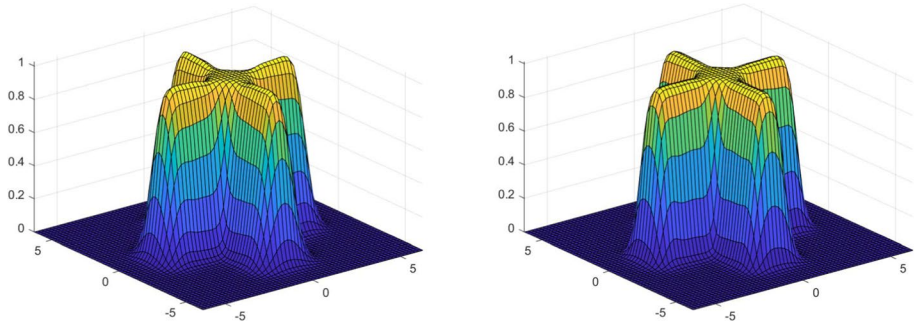


(a) Previous



(b) Proposed

Fig. 1 (Example 3) The numerical solutions of equation (40) with a function (42) at $T = 2\pi$ using 90×90 grids



(a) Previous **(b)** Proposed

Fig. 2 (Example 3) The numerical solutions of equation (40) with a function (42) at $T = 2\pi$ using 60×60 grids

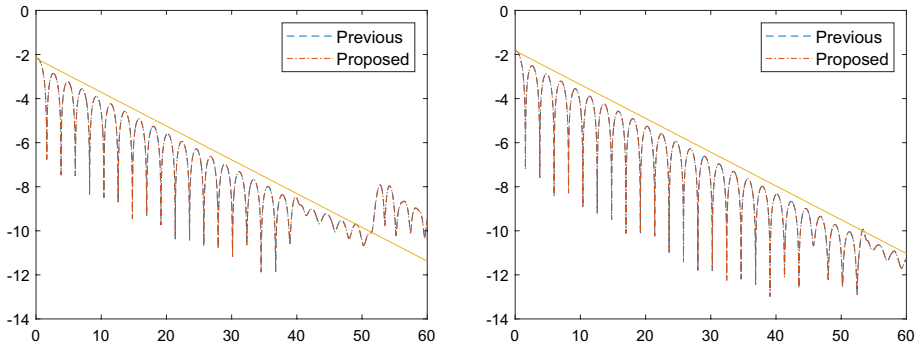


Fig. 3 (Example 4: weak Landau damping) L_2 norm of the electric field with 64×128 (left) and 128×256 grids (right). Yellow line indicates the damping rate

L_2 norms, energy and entropy of the solutions with 64×128 grids are given in Fig. 4. The proposed scheme produced the same results in electric fields, L_1 and L_2 norms as the previous one, and slightly better results in energy and entropy than the previous method.

Second, we choose a perturbation $\alpha = 0.5$ for seeing the strong Landau damping. The plots for the phase space at $T = 40, 80$ and 120 with the previous and proposed scheme

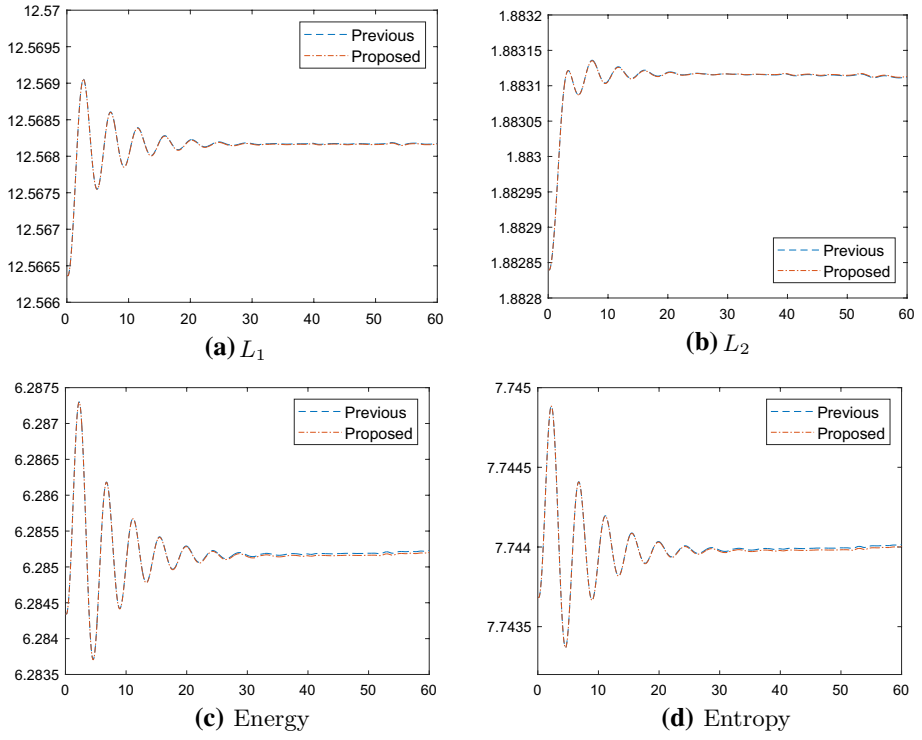


Fig. 4 (Example 4: weak Landau damping) (a) L_1 norms, (b) L_2 norms, (c) energy and (d) entropy evolution with 64×128 grids

using 128×256 grids are given in Fig. 5. To observe the shape of the phase space following the time history with several resolutions, we present plots from the proposed scheme using 64×128grid, 128×256 and 256×512 grids in Figs. 6, 7 and 8, respectively. Finally, L_2 norms of the electric fields with 64×128 and 128×256 grids are shown in Fig. 9 and also the L_1 norms, L_2 norms, energy and entropy of the solutions with 64×128 grids are

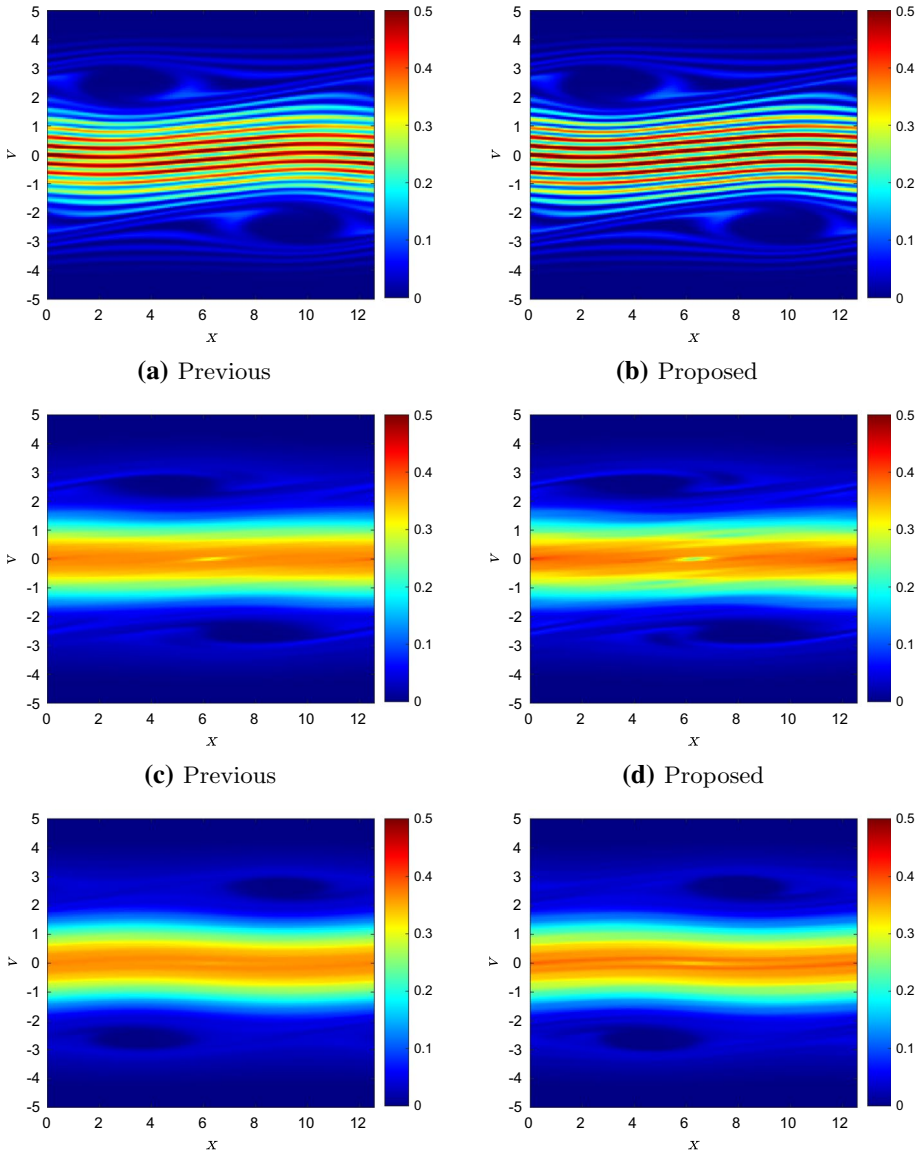


Fig. 5 (Example 4: strong Landau damping) The phase space at $T = 40$ ((a), (b)), $T = 80$ ((c), (d)) and $T = 120$ ((e),(f)) with previous scheme ((a), (c), (e)) and proposed scheme ((b), (d), (f)) using 128×256 grids

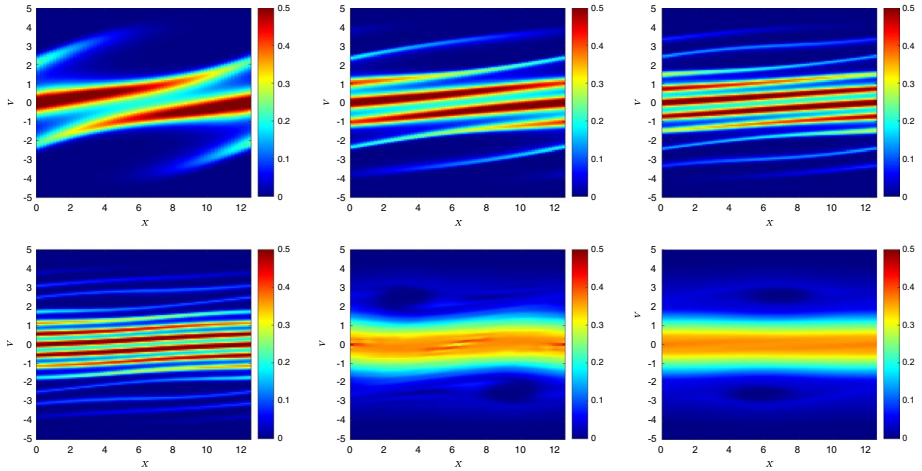


Fig. 6 (Example 4: strong Landau damping) The phase space of the time evolution until $T = 80$ ($T = 5, 10, 15, 20, 40$ and 80) using 64×128 grids

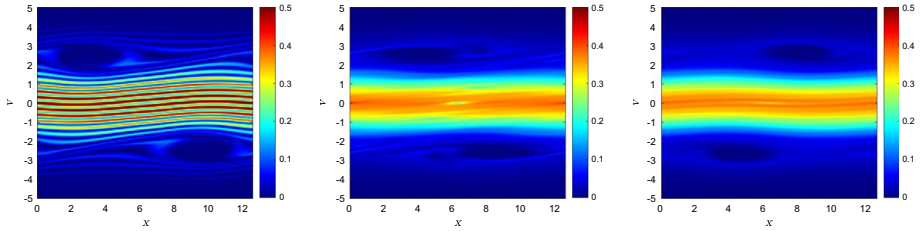


Fig. 7 (Example 4: strong Landau damping) The phase space of the time evolution until $T = 120$ ($T = 40, 80$ and 120) using 128×256 grids

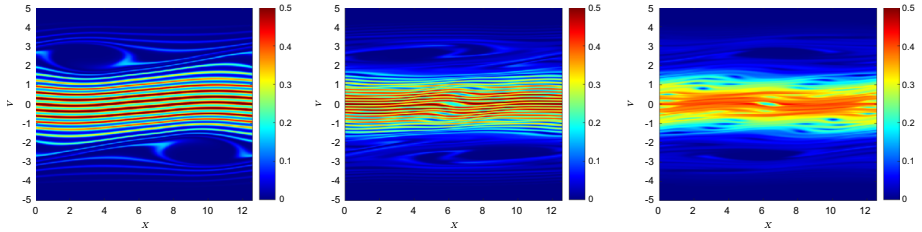


Fig. 8 (Example 4: strong Landau damping) The phase space of the time evolution until $T = 120$ ($T = 40, 80$ and 120) using 256×512 grids

given in Fig. 10. We can observe that the scheme does not preserve the discrete quantities very well, which is a well-known problem with the mesh-based method, but the proposed scheme preserves the energy and entropy better than the previous scheme.

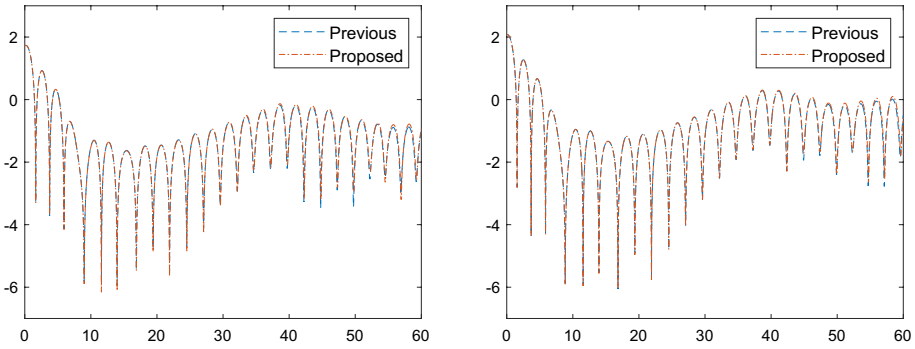


Fig. 9 (Example 4: strong Landau damping) L_2 norm of the electric field with 64×128 (left) and 128×256 (right) grids

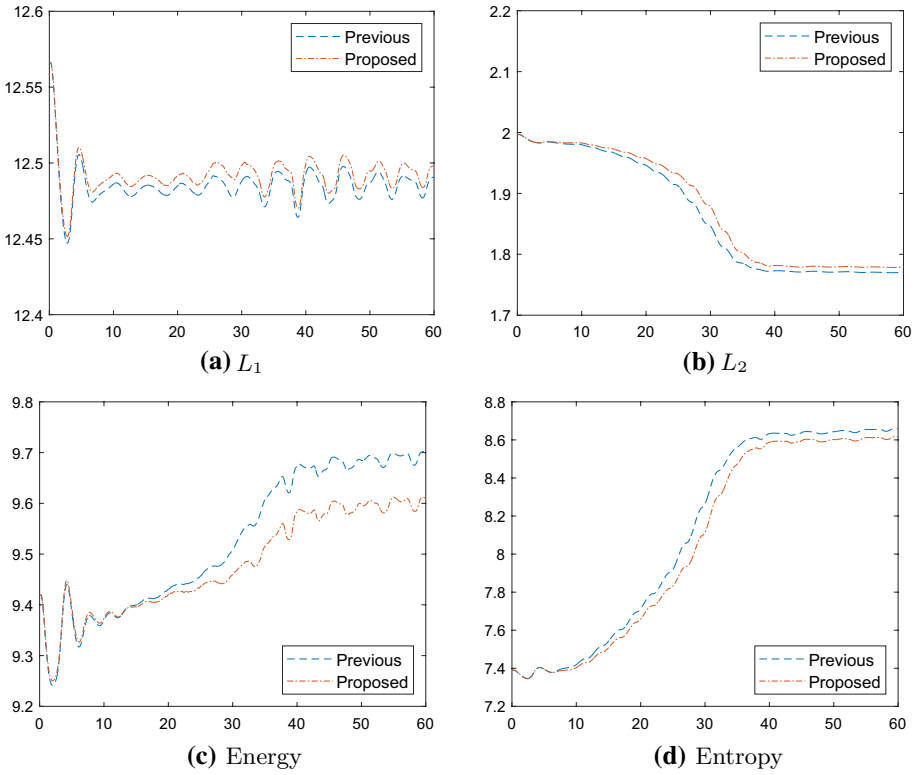


Fig. 10 (Example 4: strong Landau damping) (a) L_1 norms, (b) L_2 norms, (c) energy and (d) entropy evolution with 64×128 grids

Example 5 (Two-stream instability) Finally we simulate the symmetric warm two-stream instability problem [20]:

$$f_0(x,v) = \left(1 + \alpha\left(\frac{\cos(2kx) + \cos(3kx)}{1.2} + \cos(kx)\right)\right)\frac{2}{7\sqrt{2\pi}}\mathrm{e}^{-v^2/2}(1+5v^2),$$

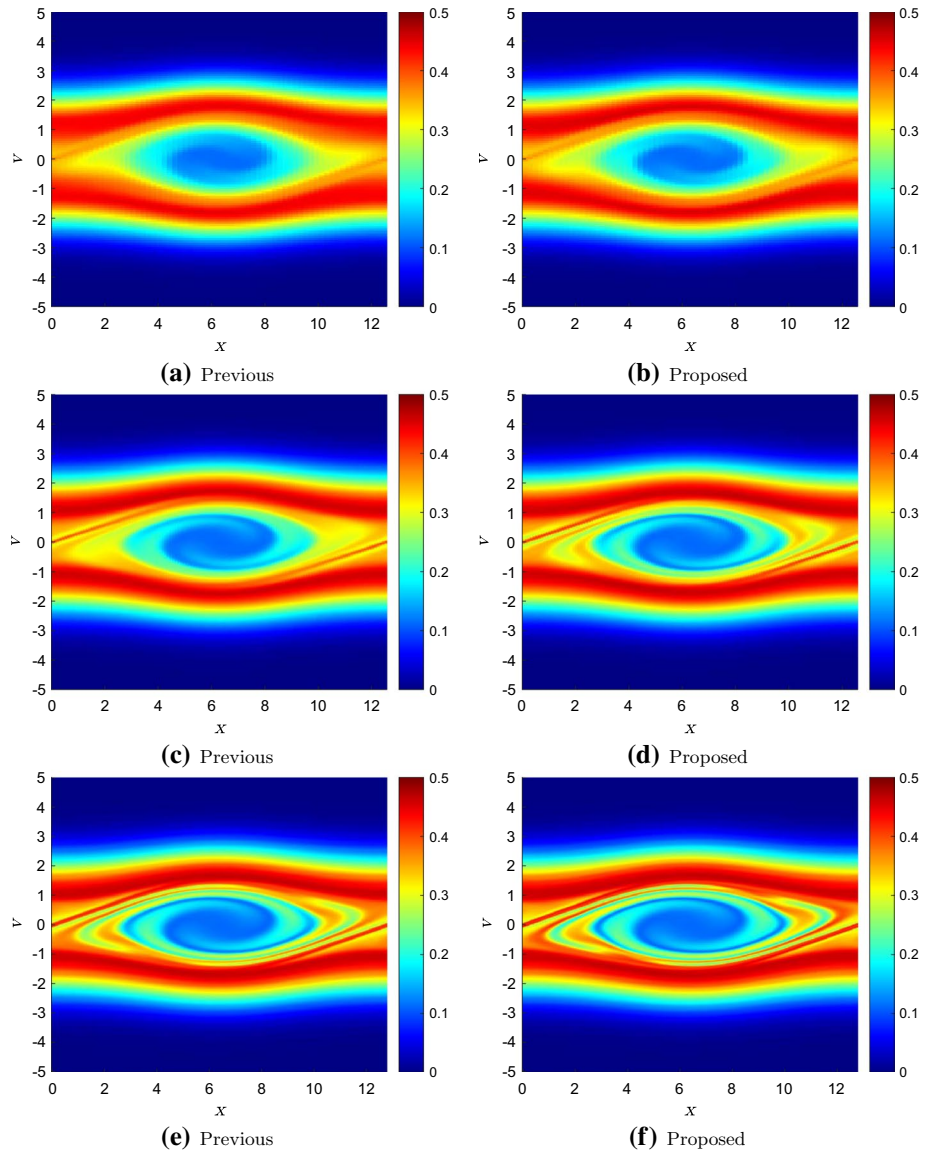


Fig. 11 (Example 5: two-stream instability) Comparison of the phase spaces from previous and proposed schemes with 64×128 ((a), (b)), 128×256 ((c), (d)) and 256×512 ((e), (f)) grids

where α is the strength of the perturbation. Here, we set $\alpha = 0.01$ and $k = 0.5$. The domain boundaries are $a = 0, b = 4\pi$ and $V = 5$ in x and v spaces, respectively.

The results from the previous and proposed schemes are given in Fig. 11. In Fig. 11 (b) which is a result from the proposed scheme with 64×128 resolution, there is a structure around velocity $v = \pm 1$ that is not well supported in (a), which is a result from the previous scheme, but is observed in (c), 128×256 resolution with the polynomial-based method. Likewise, if we look at Fig. 11 (d), proposed results for 128×256, there are structures around $x = 3$ and $x = 11$ between velocity $v = -1$ and $v = 1$ which are not supported by the previous method (c) but show up in (e), 256×512 resolution in the polynomial-based method. It is clear that the new method is supporting finer structure on meshes at half the resolution of the polynomial-based method. In 6D, this would serve to be a substantial savings.

6 Conclusion

In this paper, we proposed an SL method based on non-polynomial function space for solving the Vlasov equation. More accurate approximation was achieved by exploiting the trigonometric polynomial basis at the stage of the construction of the WENO scheme. The proposed schemes were applied to solve various scalar test problems containing 2D advection equation and rigid body rotation problem, and the VP system containing weak and strong Landau damping and two-stream instability problems. While the proposed methods demonstrated improved accuracy over the previous SL method with classical WENO approaches, the use of high-order splitting methods in time also helps for problems exhibiting discontinuous structures. A highlight of this work is reflected in the two-stream instability problem (see Example 3 in Sect. 5), where the proposed method with lower resolution presents the results similar to those of the previous results with higher resolution. In 6D, this would imply that the new non-polynomial method would have substantial cost savings over traditional SL methods based on polynomial reconstruction. In future work, we plan to develop the proposed scheme to incorporate the positivity preserving property, as suggested in [50] and also investigate an SL scheme capable of choosing the optimal parameters using local properties of the underlying data to further increase accuracy, as in [14, 16].

Appendix A Formulation of C and $\mathbf{c}_k$

The numerical flux $\hat{f}_{j-\frac{1}{2}}$ in (25) based on the function space $\Gamma_5 = \text{span}\{\cos x, \sin x, x^2, x^3, x^4\}$ is constructed by using the coefficients

$$\mathbf{C} = \left[\frac{550}{16\,403}, -\frac{157}{724}, \frac{1\,587}{2\,057}, \frac{281}{599}, -\frac{155}{3\,016}\right]^{\mathrm{T}}.$$

The local numerical flux $\hat{f}^k_{j-\frac{1}{2}}$ in (26) can be obtained based on algebraic function space $\Pi_3 = \text{span}\{1, x, x^2\}$ with the constants

$$\mathbf{c}^0 = \left[\frac{1}{3}, -\frac{7}{6}, \frac{11}{6}\right]^{\mathrm{T}},$$
$$\mathbf{c}^1 = \left[-\frac{1}{6}, \frac{5}{6}, \frac{1}{3}\right]^{\mathrm{T}},$$
$$\mathbf{c}^2 = \left[\frac{1}{3}, \frac{5}{6}, -\frac{1}{6}\right]^{\mathrm{T}},$$

or based on the function space $\mathrm{span}\{1, \sin x, \cos x\}$ with the coefficients

$$\mathbf{c}^0 = \left[\frac{109}{280}, -\frac{478}{471}, \frac{1411}{868}\right]^{\mathrm{T}},$$
$$\mathbf{c}^1 = \left[-\frac{826}{4031}, \frac{1973}{2419}, \frac{109}{280}\right]^{\mathrm{T}},$$
$$\mathbf{c}^2 = \left[\frac{109}{280}, \frac{1973}{2419}, -\frac{826}{4031}\right]^{\mathrm{T}}.$$

Appendix B Proof of Proposition 2

Assuming f smooth enough around the stencil, we can represent the smoothness indicators (28) using Taylor expansion as

$$\beta_0 = \left(\Delta x f'_{j-\frac{1}{2}} - \frac{23}{24}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2 + \left(\Delta x^2 f''_{j-\frac{1}{2}} - \frac{3}{2}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2,$$
$$\beta_1 = \left(\Delta x f'_{j-\frac{1}{2}} + \frac{1}{24}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2 + \left(\Delta x^2 f''_{j-\frac{1}{2}} - \frac{1}{2}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2,$$
$$\beta_2 = \left(\Delta x f'_{j-\frac{1}{2}} + \frac{1}{24}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2 + \left(\Delta x^2 f''_{j-\frac{1}{2}} + \frac{1}{2}\Delta x^3 f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^4)\right)^2.$$

Then for each $k = 0, 1, 2$, we can obtain

$$\begin{aligned}
1 + \frac{\tau}{(\epsilon + \beta_k)^2} &= 1 + \frac{(\beta_2 - \beta_0)^2}{(\epsilon + \beta_k)^2} \\
&= 1 + \frac{\left(2\Delta x^4 f'_{j-\frac{1}{2}} f'''_{j-\frac{1}{2}} + \mathcal{O}(\Delta x^5)\right)^2}{(\epsilon + \beta_k)^2} \\
&= 1 + C\Delta x^4 + \mathcal{O}(\Delta x^5)
\end{aligned}$$

with a constant C, by choosing $\epsilon = \Delta x^2$, so that it is straightforward by the definition of ω_k in (29).

Acknowledgements We would like to thank AFOSR and NSF for their support of this work under grants FA9550-19-1-0281 and FA9550-17-1-0394 and NSF grant DMS 191218.

Compliance with Ethical Standards

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Besse, N., Latu, G., Ghizzo, A., Sonnendrücker, E., Bertrand, P.: A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov-Maxwell system. J. Comput. Phys. **227**(16), 7889–7916 (2008)
2. Birdsall, C.K., Langdon, A.B.: Plasma Physics Via Computer Simulation. McGraw-Hill, New York (1985)
3. Blanesa, S., Moanb, P.C.: Practical symplectic partitioned Runge-Kutta and Runge-Kutta-Nystrom methods. J. Comput. Appl. Math. **142**, 313–330 (2002)
4. Bordas, S., Nguyen, P.V., Dunant, C., Guidoum, A., Nguyen-Dang, H.: An extended finite element library. Int. J. Numer. Methods Eng. **71**(6), 703–732 (2007)
5. Boyd, J.: Chebyshev and Fourier Spectrum Methods. Springer, New York (1989)
6. Burgan, J., Gutierrez, J., Fijalkow, E., Navet, M., Feix, M.: Self-similar solutions for Vlasov and water-bag models. J. Plasma Phys. **19**(1), 135–146 (1978)
7. Cai, X., Qiu, J., Qiu, J.-M.: A conservative semi-Lagrangian HWENO method for the Vlasov equation. J. Comput. Phys. **323**, 95–114 (2016)
8. Carrillo, J.A., Vecil, F.: Nonoscillatory interpolation methods applied to Vlasov-based models. SIAM J. Sci. Comput. **29**(3), 1179–1206 (2007)
9. Chen, G., Chacon, L.: A multi-dimensional, energy-and charge-conserving, nonlinearly implicit, electromagnetic Vlasov-Darwin particle-in-cell algorithm. Comput. Phys. Commun. **197**, 73–87 (2015)
10. Chen, G., Chacón, L., Barnes, D.C.: An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm. J. Comput. Phys. **230**(18), 7018–7036 (2011)
11. Cheng, C.-Z., Knorr, G.: The integration of the Vlasov equation in configuration space. J. Comput. Phys. **22**(3), 330–351 (1976)
12. Cheng, Y., Christlieb, A.J., Zhong, X.: Energy-conserving discontinuous Galerkin methods for the Vlasov-Ampere system. J. Comput. Phys. **256**, 630–655 (2014)
13. Cheng, Y., Christlieb, A.J., Zhong, X.: Energy-conserving discontinuous Galerkin methods for the Vlasov-Maxwell system. J. Comput. Phys. **279**, 145–173 (2014)
14. Christlieb, A., Guo, W., Jiang, Y., Yang, H.: A moving mesh WENO method based on exponential polynomials for one-dimensional conservation laws. J. Comput. Phys. **380**, 334–354 (2019)
15. Christlieb, A., Hitchon, W., Lawler, J., Lister, G.: Integral and Lagrangian simulations of particle and radiation transport in plasma. J. Phys. D Appl. Phys. **42**(19), 194007 (2009)
16. Christlieb, A., Sands, W., Yang, H.: Superconvergent non-polynomial approximations (2020). https://arxiv.org/abs/2011.02654v2. Accessed 17 Jun 2021
17. Christlieb, A.J., Krasny, R., Verboncoeur, J.P., Emhoff, J.W., Boyd, I.D.: Grid-free plasma simulation techniques. IEEE Trans. Plasma Sci. **34**(2), 149–165 (2006)
18. Colombi, S., Touma, J.: Vlasov-Poisson: the waterbag method revisited. Commun. Nonlinear Sci. Numer. Simul. **13**(1), 46–52 (2008)
19. Crouseilles, N., Faou, E., Mehrenberger, M.: High order Runge-Kutta-Nystrom splitting methods for the Vlasov-Poisson equation. inria-00633934 (2011)
20. Filbet, F., Sonnendrücker, E.: Comparison of Eulerian Vlasov solvers. Comput. Phys. Commun. **150**(3), 247–266 (2003)
21. Filbet, F., Sonnendrücker, E., Bertrand, P.: Conservative numerical schemes for the Vlasov equation. J. Comput. Phys. **172**(1), 166–187 (2001)
22. Gibbon, P., Speck, R., Karmakar, A., Arnold, L., Frings, W., Berberich, B., Reiter, D., Mašek, M.: Progress in mesh-free plasma simulation with parallel tree codes. IEEE Trans. Plasma Sci. **38**(9), 2367–2376 (2010)
23. Güçlü, Y., Christlieb, A.J., Hitchon, W.N.: Arbitrarily high order convected scheme solution of the Vlasov-Poisson system. J. Comput. Phys. **270**, 711–752 (2014)
24. Ha, Y., Kim, C.H., Yang, H., Yoon, J.: Sixth-order weighted essentially nonoscillatory schemes based on exponential polynomials. SIAM J. Sci. Comput. **38**, 1987–2017 (2016)

25. Ha, Y., Kim, C.H., Yang, H., Yoon, J.: A sixth-order weighted essentially nonoscillatory schemes based on exponential polynomials for Hamilton-Jacobi equations. J. Sci. Comput. **75**, 1675–1700 (2018)
26. Ha, Y., Kim, C.H., Yang, H., Yoon, J.: Improving accuracy of the fifth-order WENO scheme by using the exponential approximation space. SIAM J. Numer. Anal. **59**, 143–172 (2021)
27. Hitchon, W., Koch, D., Adams, J.: An efficient scheme for convection-dominated transport. J. Comput. Phys. **83**(1), 79–95 (1989)
28. Hockney, R.W., Eastwood, J.W.: Computer Simulation Using Particles. McGraw-Hill, New York (1981)
29. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**(1), 202–228 (1996)
30. Karlin, S., Studden, W.: Tchebycheff Systems: with Applications in Analysis and Statistics. Interscience Publishers, Geneva (1966)
31. Laborde, P., Pommier, J., Renard, Y., Salaün, M.: High-order extended finite element method for cracked domains. Int. J. Numer. Methods Eng. **64**(3), 354–381 (2005)
32. Markidis, S., Lapenta, G.: The energy conserving particle-in-cell method. J. Comput. Phys. **230**, 7037–7052 (2011)
33. Matyash, K., Schneider, R., Sydora, R., Taccogna, F.: Application of a grid-free kinetic model to the collisionless sheath. Contrib. Plasma Phys. **48**(1/2/3), 116–120 (2008)
34. Moës, N., Dolbow, J., Belytschko, T.: A finite element method for crack growth without remeshing. Int. J. Numer. Methods Eng. **46**(1), 131–150 (1999)
35. Nakamura, T., Yabe, T.: Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space. Comput. Phys. Commun. **120**(2/3), 122–154 (1999)
36. Pohn, E., Shoucri, M., Kamelander, G.: Eulerian Vlasov codes. Comput. Phys. Commun. **166**, 81–93 (2005)
37. Qiu, J.-M., Christlieb, A.: A conservative high order semi-Lagrangian WENO method for the Vlasov equation. J. Comput. Phys. **229**(4), 1130–1149 (2010)
38. Qiu, J.-M., Russo, G.: A high order multi-dimensional characteristic tracing strategy for the Vlasov-Poisson system. J. Sci. Comput. **71**(1), 414–434 (2017)
39. Qiu, J.-M., Shu, C.-W.: Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation. Commun. Comput. Phys. **10**, 979–1000 (2011)
40. Qiu, J.-M., Shu, C.-W.: Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: theoretical analysis and application to the Vlasov-Poisson system. J. Comput. Phys. **230**(23), 8386–8409 (2011)
41. Rossmanith, J.A., Seal, D.C.: A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. J. Comput. Phys. **230**(16), 6203–6232 (2011)
42. Siddi, L., Lapenta, G., Gibbon, P.: Mesh-free Hamiltonian implementation of two dimensional Darwin model. Phys. Plasmas **24**(8), 082103 (2017)
43. Sirajuddin, D., Hitchon, W.N.: A truly forward semi-Lagrangian WENO scheme for the Vlasov-Poisson system. J. Comput. Phys. **392**, 619–665 (2019)
44. Sonnendrücker, E., Roche, J., Bertrand, P., Ghizzo, A.: The semi-Lagrangian method for the numerical resolution of the Vlasov equation. J. Comput. Phys. **149**(2), 201–220 (1999)
45. Tabarraei, A., Sukumar, N.: Extended finite element method on polygonal and quadtree meshes. Comput. Methods Appl. Mech. Eng. **197**(5), 425–438 (2008)
46. Verboncoeur, J.P.: Particle simulation of plasmas: review and advances. Plasma Phys. Control. Fusion **47**(5A), A231 (2005)
47. Wang, B., Miller, G.H., Colella, P.: A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas. SIAM J. Sci. Comput. **33**(6), 3509–3537 (2011)
48. Watanabe, T.H., Sugama, H.: Vlasov and drift kinetic simulation methods based on the symplectic integrator. Transp. Theory Stat. Phys. **34**, 287–309 (2005)
49. Wolf, E.M., Causley, M., Christlieb, A., Bettencourt, M.: A particle-in-cell method for the simulation of plasmas based on an unconditionally stable field solver. J. Comput. Phys. **326**, 342–372 (2016)
50. Xiong, T., Qiu, J.-M., Xu, Z.F., Christlieb, A.: High order maximum principle preserving semi-Lagrangian finite difference WENO schemes for the Vlasov equation. J. Comput. Phys. **273**, 618–639 (2014)
51. Xiong, T., Russo, G., Qiu, J.-M.: Conservative multi-dimensional semi-Lagrangian finite difference scheme: stability and applications to the kinetic and fluid simulations. J. Sci. Comput. **79**(2), 1241–1270 (2019)
52. Yoshida, H.: Construction of higher order symplectic integrators. Phys. Lett. A **150**, 262–268 (1990)

53. Zhu, J., Qiu, J.: Trigonometric WENO schemes for hyperbolic conservation laws and highly oscillatory problems. Commun. Comput. Phys. **8**, 1242–1263 (2010)
54. Zhu, J., Qiu, J.: WENO schemes and their application as limiters for RKDG methods based on trigonometric approximation spaces. J. Sci. Comput. **55**, 606–644 (2013)