第51卷第3期 中国科学技术大学学报 Vol. 51, No. 3
2021年3月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Mar. 2021
Received: 2021-01-04; Revised: 2021-03-20 doi:10.52396/JUST-2021-0007

A hybrid HWENO-based method of lines transpose approach for Vlasov simulations

WANG Kaipeng, JIANG Yan*, ZHANG Mengping
School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China
* Corresponding author. E-mail: jiangy@ustc.edu.cn

Abstract: A new type hybrid Hermite weighted essentially non-oscillatory (HWENO) schemes in the implicit method of lines transpose (MOL^T) framework is designed for solving one-dimensional linear transport equations and further applied to the Vlasov-Poisson (VP) simulations via dimensional splitting. Compared with the WENO-based MOL^T method given in *J. Comput. Phys.* [2016, 327: 337-367], the new proposed hybrid HWENO-based MOL^T scheme has two advantages. The first is the HWENO schemes using the stencils narrower than those of the WENO schemes with the same order of accuracy. The second is that the schemes can adapt between the linear scheme and the HWENO scheme automatically. In summary, the hybrid HWENO scheme keeps the simplicity and robustness of the simple WENO scheme, while it has higher efficiency with less numerical errors in smooth regions and less computational costs as well. Benchmark examples are given to demonstrate the robustness and good performance of the proposed scheme.

Keywords: method of lines transpose; implicit time discretization; HWENO methodology; hybrid scheme; Vlasov-Poisson system
CLC number: O242.1 **Document code**: A
2010 Mathematics Subject Classification: 65M99

1 Introduction

In this paper, we develop efficient numerical solvers for the Vlasov-Poisson (VP) system, which is considered a fundamental model in plasma physics describing the dynamics of charged particles due to the self-consistent electric force. The VP system is given as follows:

$$f_t + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f + E(\boldsymbol{x},t)\cdot\nabla_{\boldsymbol{v}} f = 0,\ \boldsymbol{x}\times\boldsymbol{v}\in\Omega_{\boldsymbol{x}}\times\Omega_{\boldsymbol{v}} \tag{1a}$$

$$\boldsymbol{E}(\boldsymbol{x},t) = -\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x},t),\ -\Delta_{\boldsymbol{x}}\phi(\boldsymbol{x},t) = -1+\rho(\boldsymbol{x},t) \tag{1b}$$

where $f(\boldsymbol{x},\boldsymbol{v},t)$ describes the probability of finding a particle with velocity $\boldsymbol{v}$ at position $\boldsymbol{x}$ at time t, $\boldsymbol{E}$ is the electrostatic field, ϕ is the self-consistent electrostatic potential, and $\rho(\boldsymbol{x},t) = \int_{\Omega_{\boldsymbol{v}}} f(\boldsymbol{x},\boldsymbol{v},t)\mathrm{d}\boldsymbol{v}$ is the electron charge density and the 1 represents the uniformly distributed infinitely massive ions in the background.

Besides the famous "curse of dimensionality", the main challenge of VP system is that the solution may develop filamentation solution structures under the long-range forces. Examples will be shown later. Consequently, numerical scheme needs careful design such that it can effectively capture filamentation structures without producing spurious oscillations.

There exist a large amount of successful VP solvers in the literature, such as Eulerian type approaches[1-4], and the semi-Lagrangian approach[5-17], and the Lagrangian methods, which including the particle-in-cell (PIC) method[18-20] and Lagrangian particle methods[21,22]. However, each method has its own limitations. For example, Eulerian approaches are suffering from the CFL time step restriction; general boundary conditions are difficult to handle in a semi-Lagrangian setting; and the Lagrangian methods are known to suffer from low order sampling noise. Many of these existing methods are in the method of line (MOL) framework, meaning that the spatial variable is first discretized, then the numerical solution is updated in time by coupling a suitable time integrator.

Recently, an alternative approach to advance the solution called the method of line transpose (MOL^T) is well designed to solve the VP system[23]. This method is also known as the Rothe's method or transversal method of lines in the literature[24,25]. In this framework, the discretization is first carried out for

Citation: WANG Kaipeng, JIANG Yan, ZHANG Mengping. A hybrid HWENO-based method of lines transpose approach for Vlasov simulations. J. Univ. Sci. Tech. China, 2021, 51(3): 202-215.

temporal variable, resulting in a boundary value problem (BVP) at the discrete time levels. Each BVP can be inverted analytically in an integral formulation based on a kernel function and then the numerical solution is updated accordingly. A notable advantage of such a scheme is that, even though it is implicit in time, we do not need to explicitly solve a linear system, while the BVP is inverted analytically in an integral formulation. Moreover, this method is capable of conveniently handling general boundary conditions. This MOL^T approach has been applied to solve the heat equation, the Allen-Cahn equation[26], Maxwell's equations[27], transport equation and the VP system[23]. Note that the MOL^T framework is rarely applied to general nonlinear problems, mainly because efficient algorithms of inverting nonlinear BVPs are lacking. In Ref. [23], the nonlinear multi-dimensional Vlasov equation is first split into several lower-dimensional linear transport equations, which have the analytical BVP solver. Furthermore, in order to capture sharp gradient of the solution of the transport equations and distinctive filamentation solution structures of the VP system effectively without producing spurious oscillations, a robust weighted essential non-oscillatory (WENO) methodology is incorporated when inverting the BVP in the integral formulation. The key idea of WENO methodology is a nonlinear combination of numerical approximations on all candidate stencils.

In this paper, we consider to design a suitable hybrid Hermite weighted essentially non-oscillatory (HWENO) for the integral formulation. The main difference of HWENO schemes from WENO schemes is that both the function and its first-order derivative values are evolved in time and used in the integration, not like the WENO schemes in which only the function values are evolved and used in the integration. This allows the HWENO schemes to obtain the same order of accuracy as the WENO schemes with narrower stencils. Hence, less ghost points are needed near the boundaries. Note that the cost of the computation of smoothness indicators and nonlinear weights in HWENO methodology is very high. Hence, we propose the hybrid HWENO scheme in this paper, which can use the optimal linear weights and avoid the computation of the nonlinear weights in the smooth region. Therefore, the hybrid WENO scheme keeps the robustness of the simple WENO scheme, while it is very easy to implement in practice and has higher efficiency.

The rest of the paper is organized as follows. In Section 2, we give the review of the MOL^T approach for one-dimensional linear transport equations. Details of the hybrid HWENO method are given in Section 3. In Section 4, the method is extended to two-dimensional transport equations via dimensional splitting. Several benchmark tests are presented in Section 5 to verify the performance of the proposed scheme, including rigid body rotation, as well as Landau damping, two-stream instabilities, and bump-on-tail for VP simulations. We conclude the paper in Section 6.

2 MOL^T framework

In this section, we give a short review of the MOL^T framework for the following one-dimensional advection equation,

$$\left.\begin{aligned} &u_t + cu_x = 0,\ x \in [a,b],\\ &u(x,0) = u_0(x) \end{aligned}\right\} \tag{2}$$

where c is constant denoting the wave propagation speed. The boundary condition is assumed to be periodic, or Dirichlet boundary condition,

$$\left.\begin{aligned} &u(a,t) = g_1(t), \text{ for } c > 0,\\ &\text{or } u(b,t) = g_2(t), \text{ for } c < 0 \end{aligned}\right\} \tag{3}$$

In the MOL^T framework, the time variable is discretized at first while leaving the spatial variable continuous. We take the first order backward Euler (BE) method as an example, and obtain

$$\frac{u^{n+1} - u^n}{\Delta t} + cu_x^{n+1} = 0 \tag{4}$$

i.e.,

$$u_x^{n+1} + \operatorname{sgn}(c) \cdot \alpha u^{n+1} = \operatorname{sgn}(c) \cdot \alpha u^n \tag{5}$$

where Δt denotes the time step and $\alpha = \dfrac{1}{|c\Delta t|}$. For the boundary value problem (BVP) (5), we can obtain an explicit representation for solution u^{n+1} at time level t^{n+1}.

$$u^{n+1}(x) = \begin{cases} \mathscr{I}^L[u^n,\alpha](x) := I^L[u^n,\alpha](x) + A^{n+1}\mathrm{e}^{-\alpha(x-a)}, \\ \qquad \text{if } c > 0, \\ \mathscr{I}^R[u^n,\alpha](x) := I^R[u^n,\alpha](x) + B^{n+1}\mathrm{e}^{-\alpha(b-x)}, \\ \qquad \text{if } c < 0 \end{cases} \tag{6}$$

where

$$I^L[u^n,\alpha](x) = \alpha\int_a^x \mathrm{e}^{-\alpha(x-y)}u^n(y)\,\mathrm{d}y \tag{7a}$$

$$I^R[u^n,\alpha](x) = \alpha\int_x^b \mathrm{e}^{-\alpha(y-x)}u^n(y)\,\mathrm{d}y \tag{7b}$$

A^{n+1} and B^{n+1} are constants and should be determined by the boundary conditions. The superscript L (or R) indicates that the characteristics traverse from the left to the right (or from the right to the left).

To obtain the second order accuracy in time, we can use the Crank-Nicolson (CN) time discretization on 920 and obtain the semi-discrete scheme

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}c(u_x^{n+1} + u_x^n) = 0 \tag{8}$$

Based on the similar idea, solution of the BVP (8) can be given analytically,

$$u^{n+1}(x) = \begin{cases} 2\mathscr{I}^L[u^n,\alpha](x) - u^n(x), \text{ if } c > 0, \\ 2\mathscr{I}^R[u^n,\alpha](x) - u^n(x), \text{ if } c < 0 \end{cases} \tag{9}$$

with $\alpha=\frac{2}{|c\Delta t|}$. Again, A^{n+1} and B^{n+1} can be obtained through the boundary conditions.

For both semi-discrete schemes (6) and (9), we need to further discretize the spatial variable. Suppose the domain $[a,b]$ is discretized by $M+1$ uniformly distributed grid points

$$a = x_0 < x_1 < \cdots < x_{M-1} < x_M = b,$$

with mesh size $\Delta x=\frac{b-a}{M}$. We use u_i^n to denote the numerical solution at location x_i at time level t^n. Next, we will focus on how to evaluate the convolution integral $I^L[u^n,\alpha](x)$ or $I^R[u^n,\alpha]$ in on each grid point x_i, denoted by I_i^L or I_i^R, based on the point values $\{u_i\}$. Note that

$$I_i^L = I_{i-1}^L e^{-\alpha\Delta x} + J_i^L,\ i = 1,\cdots,M,\ I_0^L = 0 \quad (10a)$$

$$I_i^R = I_{i+1}^R e^{-\alpha\Delta x} + J_i^R,\ i = 0,\cdots,M-1,\ I_M^R = 0 \quad (10b)$$

where

$$\left.\begin{aligned} J_i^L &= \alpha\int_{x_{i-1}}^{x_i} u^n(y)e^{-\alpha(x_i-y)}\,dy,\\ J_i^R &= \alpha\int_{x_i}^{x_{i+1}} u^n(y)e^{-\alpha(y-x_i)}\,dy \end{aligned}\right\} \quad (11)$$

Therefore, once we have computed J_i^L or J_i^R for all i, I_i^L or I_i^R can be obtained via the recursive relation (10).

After computing the integration I_i^L or I_i^R, we further need to find the global coefficient A or B in (6) or (9) to enforce the boundary condition. For a Dirichlet boundary condition, the global coefficient can be obtained by

$$A^{n+1} = g_1(t^{n+1}),\ \text{or}\ B^{n+1} = g_2(t^{n+1}) \quad (12)$$

when coupling with the BE scheme (6), or

$$\left.\begin{aligned} A^{n+1} &= \frac{1}{2}(g_1(t^n)+g_1(t^{n+1})),\\ \text{or } B^{n+1} &= \frac{1}{2}(g_2(t^n)+g_2(t^{n+1})) \end{aligned}\right\} \quad (13)$$

when coupling with the CN scheme (9). For the periodic boundary condition, the coefficients can be determined by using the mass conservation property of the solutions, which leads to

$$A^{n+1} = \Big(\sum_{i=0}^{M-1} u_i^n - \sum_{i=0}^{M-1} I_i^L\Big) \Big/ \sum_{i=0}^{M-1} e^{-i\alpha\Delta x} \quad (14)$$

or

$$B^{n+1} = \Big(\sum_{i=1}^{M} u_i^n - \sum_{i=1}^{M} I_i^R\Big) \Big/ \sum_{i=1}^{M} e^{-(M-i)\alpha\Delta x} \quad (15)$$

3 HWENO methodology

The HWENO has been widely studied to solve the hyperbolic conservation laws in MOL framework, including the finite difference method[28-30], the finite volume method and the nonlinear limiter for the discontinuous Galerkin (DG) method[31-34]. Here, we will use the idea to approximate the integration.

In the Hermite framework, both the function and its spatial derivative are needed to be evolved in time. Denote the spatical derivative of function $u(x,t)$ as $v(x,t)$. Then from (2) and its spatial derivative, we have the governing equation:

$$\boldsymbol{U}_t + c\boldsymbol{U}_x = 0 \quad (16)$$

where

$$\boldsymbol{U}(x,t) = \begin{pmatrix} u(x,t)\\ v(x,t)\end{pmatrix}.$$

The boundary conditions can be obtained correspondingly, for instance, Dirichlet boundary condition

$$\left.\begin{aligned} \boldsymbol{U}(a,t) &= \begin{pmatrix} g_1(t)\\ -\frac{1}{c}g'_1(t)\end{pmatrix},\ \text{for } c>0,\\ \text{or } \boldsymbol{U}(b,t) &= \begin{pmatrix} g_2(t)\\ -\frac{1}{c}g'_2(t)\end{pmatrix},\ \text{for } c<0 \end{aligned}\right\} \quad (17)$$

Note that each component of system (16) satisfies the 1D advection equation (2). Hence, we can apply the MOL^T framework introduced in Section 2 directly on u and v respectively. For instance, when $c>0$ and coupling the BE scheme, we have

$$u^{n+1}(x_i) = I^L[u^n,\alpha](x_i) + A_u^{n+1}e^{-\alpha(x_i-a)} \quad (18a)$$

$$v^{n+1}(x_i) = I^L[v^n,\alpha](x_i) + A_v^{n+1}e^{-\alpha(x_i-a)} \quad (18b)$$

That is, we will compute the integral I^L for u and v, respectively. After that, coefficients A_u^{n+1} and A_v^{n+1} can be obtained with corresponding boundary conditions.

Instead of evaluating the increment J_i^L or J_i^R for u and v based on themselves respectively, the main difference of the HWENO algorithm is to calculate the increment using u and v at the same time. In particular, when the solution is smooth enough, we can use the linear Hermite interpolation method to involve J_i^*, where $*$ can be L or R. In the following, we will show how to get J_i^L in detail. The process to obtain J_i^R is mirror-symmetric to that for J_i^L with respect to the point x_i.

Here, taking the stencil $S=\{u_{i-1},\ u_i,\ v_{i-1},\ v_i\}$ as an example, there is a unique Hermite interpolation polynomial $p(x)$ of degree at most 3, such that

$$p(x_j) = u_j,\ p'(x_j) = v_j,\ j = i-1, i.$$

Then, we can obtain that the approximations

$$J_{u,i}^L \approx \alpha\int_{x_{i-1}}^{x_i} p(y)\ e^{-\alpha(x_i-y)}\,dy = \sum_{j=0}^{1} c_j u_{i-1+j} + \sum_{j=0}^{1} \tilde{c}_j\, v_{i-1+j} \quad (19)$$

$$J_{v,i}^L \approx \alpha\int_{x_{i-1}}^{x_i} p'(y)\ e^{-\alpha(x_i-y)}\,dy \quad (20)$$

where the coefficients c_j and $\tilde{c}_j$ depend on α and Δx, but not on u or v.

$$c_0 = \frac{e^{-\gamma}(12+6\gamma-\gamma^3)+6(-2+\gamma)}{\gamma^3},$$
$$c_1 = \frac{-e^{-\gamma}(12+6\gamma)+(12-6\gamma+\gamma^3)}{\gamma^3},$$
$$\tilde{c}_0 = \frac{e^{-\gamma}(6+4\gamma+\gamma^2)+(-6+2\gamma)}{\gamma^3}\Delta x,$$
$$\tilde{c}_1 = \frac{e^{-\gamma}(6+2\gamma)-(6-4\gamma+\gamma^2)}{\gamma^3}\Delta x,$$

with $\gamma=\alpha\Delta x$. The schemes (19) and (20) will give us third order accuracy in space. It is worth noting that we can perform the integration by parts (IBP) to the polynomial $p(x)$, and obtain that

$$\alpha\int_{x_{i-1}}^{x_i} p'(y)\, e^{-\alpha(x_i-y)}\,dy = p(x_i) - p(x_{i-1})e^{-\alpha\Delta x} - \alpha^2\int_{x_{i-1}}^{x_i} p(y)\, e^{-\alpha(x_i-y)}\,dy,$$

which means

$$J_{v,i}^L \approx u_i - u_{i-1}e^{-\alpha\Delta x} - \alpha J_{u,i}^L \tag{21}$$

Hence, we can obtain $J_{v,i}^L$ from $J_{u,i}^L$ and $\{u_i\}$ via (21) rather than computing the integral (20), which can reduce computational cost. This is also true for $J_{v,i}^R$ and $J_{u,i}^R$,

$$J_{v,i}^R \approx -u_i + u_{i+1}e^{-\alpha\Delta x} + \alpha J_{u,i}^R \tag{22}$$

Both linear interpolation method (19) and the IBP algorithm (21) to compute increment J_i^L work well for smooth problems. However, they may generate spurious oscillations when the solution has discontinuities. In order to control such undesired oscillations, we will incorporate a hybrid high order HWENO methodology to evaluate J_i^L.

The main procedures of the hybrid HWENO scheme are given as follows. On each cell, we will identify the troubled cell. However, different troubled-cell indicators may have different effects for the hybrid HWENO scheme. Here, we follow the idea in Ref. [30,35] that looking at the locations of all extreme points of the big interpolation polynomial. Then calculate the integrations by HWENO approximation in troubled cell and linear approximation in other regions. A series of unequal-sized hierarchical stencils are used in designing high order HWENO schemes. This kind of unequal-sized hierarchical stencils WENO scheme is proposed in Ref. [36] for solving hyperbolic conservation laws with the good properties such as the linear weights can also be any positive numbers if their sum equals one.

In the following, a third order scheme to obtain J_i^L is shown as an example. Here, we take the big stencil as $S=\{u_{i-1}, u_i, v_{i-1}, v_i\}$.

Algorithm 3.1 The hybrid HWENO method for the 1D system (16).

Step 1 The discontinuity indicator is applied to identify troubled cell, namely the locations of discontinuity of the numerical solution. Here, the discontinuity indicator relies on the locations of all extreme points of the Hermite interpolation polynomial $p(x)$ on the big spatial stencil S. Since the integral $J_{v,i}^L$ depends on $p'(x)$, we will look at the extreme points of $p'(x)$ as well.

If all of the extreme points of $p(x)$ and $p'(x)$ are outside the interval $I_i=[x_{i-1}, x_i]$, which means the polynomials are monotone in this cell, then we will use linear schemes (19) and (21).

However, if the interval I_i is indexed as a troubled cell, then, we approximate the integrations $J_{u,i}^L$ and $J_{v,i}^L$ based on the HWENO approximation.

Step 2 We first consider the HWENO approximation of $J_{u,i}^L$ based on the big stencil is S. Two candidate stencils are chosen as $S_1=\{u_{i-1}, u_i\}$ and $S_2=S$. The procedure of the algorithm is as below:

Step 2.1 On each candidate stencil S_r, construct (Hermite) interpolation polynomials with degree at most $2r-1$, denoted by $p_r(x)$. The integration can be approximated by

$$\tilde{J}_{u,i}^{L,(r)} = \alpha\int_{x_{i-1}}^{x_i} e^{\alpha(x_i-y)}p_r(y)\,dy \tag{23}$$

In particular,

$$\tilde{J}_{u,i}^{L,(1)} = \frac{1-e^{-\gamma}(1+\gamma)}{\gamma}u_{i-1} + \frac{-1+\gamma+e^{-\gamma}}{\gamma}u_i,$$

and $\tilde{J}_{u,i}^{L,(2)}$ is exactly the same as (19).

Step 2.2 Obtain equivalent expressions for these (Hermite) interpolation polynomials of different degrees, denoted by $q_r(x)$.

$$q_1(x)=p_1(x),\ q_2(x)=\frac{1}{\theta_2}p_2(x)-\frac{\theta_1}{\theta_2}q_1(x) \tag{24}$$

with $\theta_2>0$, $\theta_1\geqslant 0$ and $\theta_1+\theta_2=1$. Moreover, we can define a linear weight $d_r=\theta_r$, $r=1,2$, such that

$$p_2(x) = d_1\,q_1(x) + d_2\,q_2(x).$$

Hence, we can see that all linear weights are positive and $d_1+d_2=1$. Note that the choice of θ_r is not unique. Here, we take them as $\theta_1=\frac{1}{101}$ and $\theta_2=\frac{100}{101}$. This means that

$$q_2(x) = 1.01p_2(x) - 0.01q_1(x).$$

Moreover, we can obtain the corresponding integral

$$J_{u,i}^{L,(r)} = \alpha\int_{x_{i-1}}^{x_i} e^{\alpha(x_i-y)}q_r(y)\,dy.$$

In particular,

$$J_{u,i}^{L,(1)} = \tilde{J}_{u,i}^{L,(1)},\ J_{u,i}^{L,(2)} = 1.01\tilde{J}_{u,i}^{L,(2)} - 0.01\,J_{u,i}^{L,(1)}.$$

Step 2.3 Compute the nonlinear weights based on the linear weights and the smoothness indicators β_r, which are given as

$$\beta_r = \sum_{l=1}^{2(l-1)} \int_{x_{i-1}}^{x_i} \Delta x^{2l-1} \left(\frac{d^l p_r(x)}{dx^l} \right)^2 dx.$$

The explicit formulations of β_r are given as following,

$$\beta_1 = (u_{i-1} - u_i)^2,$$

$$\beta_2 = \frac{781}{20}(2u_{i-1} - 2u_i + \Delta x v_{i-1} + \Delta x v_i)^2 + \frac{13}{12}\Delta x^2 (v_{i-1} - v_i)^2 + (u_{i-1} - u_i)^2.$$

And we can obtain the nonlinear weights as

$$\omega_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \ r = 1,2 \tag{25}$$

where

$$\alpha_r = d_r \left(1 + \frac{\tau}{\epsilon + \beta_r} \right), \ \tau = (\beta_1 - \beta_2)^2 \tag{26}$$

Here, we take $\epsilon = 10^{-10}$ to avoid the denominator becoming zero.

Step 2.4 Finally, the integral can be obtained as

$$J_{u,i}^L = \omega_1 J_{u,i}^{L,(1)} + \omega_2 J_{u,i}^{L,(2)} \tag{27}$$

Step 3 Construct the HWENO approximation of $J_{v,i}^L$. Here, the big stencil S and $\widetilde{S}_2$ are taken as the same as those for $J_{u,i}^L$, that is $\widetilde{S}_2 = S = \{u_{i-1}, u_i, v_{i-1}, v_i\}$. But we change the small stencil S_1 to $\widetilde{S}_1 = \{v_{i-1}, v_i\}$.

Step 3.1 On each small stencils, construct the approximation, denoted by $\widetilde{J}_{v,i}^{L,(r)}$. In particular, on S_2, $\widetilde{J}_{v,i}^{L,(r)}$ are obtained with the help of integration by parts to reduce the computational cost,

$$\widetilde{J}_{v,i}^{L,(2)} = u_i - e^{-\alpha \Delta x} u_{i-1} - \alpha \widetilde{J}_{u,i}^{L,(2)}.$$

While, on S_1, we approximate $\widetilde{J}_{v,i}^{L,(1)}$ based the polynomial interpolating the derivative function directly,

$$\widetilde{p}_1(x) = v_{i-1} \frac{x - x_i}{x_{i-1} - x_i} + v_i \frac{x - x_{i-1}}{x_i - x_{i-1}},$$

and

$$\widetilde{J}_{v,i}^{L,(1)} = \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(y - x_i)} \widetilde{p}_1(y) dy = \frac{1 - e^{-\gamma}(1 + \gamma)}{\gamma} v_{i-1} + \frac{-1 + \gamma + e^{-\gamma}}{\gamma} v_i,$$

with $\gamma = \alpha \Delta x$.

Step 3.2 Obtain equivalent expressions for these integrals with different accuracy, denoted by $J_{v,i}^{L,(r)}$

$$J_{v,i}^{L,(1)} = \widetilde{J}_{v,i}^{L,(1)}, \ J_{v,i}^{L,(2)} = 1.01 \widetilde{J}_{v,i}^{L,(2)} - 0.01 J_{v,i}^{L,(1)} \tag{28}$$

Step 3.3 Compute the nonlinear weights based on the linear weights and the smoothness indicators.

$$\widetilde{\omega}_r = \frac{\widetilde{\alpha}_r}{\widetilde{\alpha}_1 + \widetilde{\alpha}_2}, \ r = 1,2 \tag{29}$$

where

$$\widetilde{\alpha}_r = d_r \left(1 + \frac{\tau}{\epsilon + \widetilde{\beta}_r} \right), \ \tau = (\widetilde{\beta}_1 - \widetilde{\beta}_2)^2$$

and the linear weights are given as

$$d_1 = \frac{1}{101}, \ d_2 = \frac{100}{101}.$$

The smoothness indicators $\widetilde{\beta}_r$ are computed as

$$\left.\begin{aligned} \widetilde{\beta}_1 &= \int_{x_{i-1}}^{x_i} \Delta x \left(\frac{d\widetilde{p}_1(x)}{dx} \right)^2 dx = \Delta x^2 (v_{i-1} - v_i)^2, \\ \widetilde{\beta}_2 &= \sum_{l=2}^{3} \int_{x_{i-1}}^{x_i} \Delta x^{2l-1} \left(\frac{d^l p_2(x)}{dx^l} \right)^2 dx = \\ & 39(2u_{i-1} - 2u_i + \Delta x v_{i-1} + \Delta x v_i)^2 + \Delta x^2 (v_{i-1} - v_i)^2 \end{aligned}\right\} \tag{30}$$

Note that $p_2(x)$ is the interpolation polynomial obtained in Step 2.

Step 3.4 The new approximation is given by

$$J_{v,i}^L = \widetilde{\omega}_1 J_{v,i}^{L,(1)} + \widetilde{\omega}_2 J_{v,i}^{L,(2)}.$$

Remark 3.1 We want to remark that the third order WENO integration of J_i^L depends on the big stencil $S = \{x_{i-2}, \cdots, x_{i+1}\}$, leading to the necessity of ghost points near the boundaries. Therefore, special treatment, e.g. the WENO extrapolation, is required for complex boundary conditions, see Ref. [23]. But this is not needed for the HWENO method due to the narrow stencil, which is one main advantage of HWENO scheme.

Remark 3.2 The exact solution u of (16) satisfies the boundary-preserving principle, meaning that if the initial condition $m \leqslant u(x,0) \leqslant M$, then the exact solution $m \leqslant u(x,t) \leqslant M$ for any time $t > 0$. To maintain the numerical schemes also satisfy such a property on the discrete level, we will employ the boundary-preserving limiter introduced in Ref. [12]. It can work for both periodic and Dirichlet boundary conditions. Details would not be given here.

4 Two-dimensional implementation

In this section, we extend the algorithm to two-dimensional problems in the following form:

$$\left.\begin{aligned} & u_t + f(y,t) u_x + g(x,t) u_y = 0, \\ & (x,y) \in \Omega = [a_x, b_x] \times [a_y, b_y] \end{aligned}\right\} \tag{31}$$

For the Hermite method, we give not only the solution u but also its derivatives in x- and in y-directions. We have the following equations for u_x and u_y,

$$\left.\begin{aligned} & \frac{\partial u_x}{\partial t} + f(y,t) \frac{\partial u_x}{\partial x} + \frac{\partial (g(x,t) u_y)}{\partial x} = 0, \\ & \frac{\partial u_y}{\partial t} + \frac{\partial (f(y,t) u_x)}{\partial y} + g(x,t) \frac{\partial u_y}{\partial y} = 0 \end{aligned}\right\} \tag{32}$$

In this paper, the system (31) and (32) are solved

based on the Strang splitting method[8], which gives us the second order accurate in time. Firstly, we decouple the 2D system (31) and (32) into a sequence of independent 1D systems

$$\left.\begin{aligned} &\frac{\partial u}{\partial t}+f(y,t)\frac{\partial u}{\partial x}=0,\ (\mathrm{MOL}_x^T)\\ &\frac{\partial u_x}{\partial t}+f(y,t)\frac{\partial u_x}{\partial x}=0,\ (\mathrm{MOL}_x^T)\\ &\frac{\partial u_y}{\partial t}+\frac{\partial(f(y,t)\,u_x)}{\partial y}=0,\ (\mathrm{FD}_x) \end{aligned}\right\} \tag{33}$$

and

$$\left.\begin{aligned} &\frac{\partial u}{\partial t}+g(x,t)\frac{\partial u}{\partial y}=0,\ (\mathrm{MOL}_y^T)\\ &\frac{\partial u_y}{\partial t}+g(x,t)\frac{\partial u_y}{\partial y}=0,\ (\mathrm{MOL}_y^T)\\ &\frac{\partial u_x}{\partial t}+\frac{\partial(g(x,t)u_y)}{\partial x}=0.\ (\mathrm{FD}_y) \end{aligned}\right\} \tag{34}$$

Suppose Q_1 and Q_2 be the approximate solvers for each system, that is $\boldsymbol{U}^{n+1}=Q_1(\Delta t)\boldsymbol{U}^n$ is an approximation of (33), and $\boldsymbol{U}^{n+1}=Q_2(\Delta t)\boldsymbol{U}^n$ is an approximation of (34), with $\boldsymbol{U}=(u,u_x,u_y)^{\mathrm{T}}$. Then, second order semi-discrete scheme for one step evolution from $\boldsymbol{U}^n$ to $\boldsymbol{U}^{n+1}$ can be achieved in by employing Q_1 and Q_2 consecutively,

$$\boldsymbol{U}^{n+1}=Q_1(\frac{1}{2}\Delta t)\cdot Q_2(\Delta t)\cdot Q_1(\frac{1}{2}\Delta t)\boldsymbol{U}^n \tag{35}$$

Note that, the first two equations in system (33) and (34) can be solved by the 1D MOL^T-HWENO method given in Section 3. However, the third equations need different treatment. Here, Below, we focus our discussion on the solver Q_1 for (33). The solver Q_2 and would be similar.

Let (x_i,y_j) be the node of a 2D orthogonal grid with uniform mesh size in each direction

$$a_x=x_0<x_1<\cdots<x_{M_x}=b_x,$$
$$a_y=y_0<y_1<\cdots<y_{M_y}=b_y,$$

with $\Delta x=\frac{b_x-a_x}{M_x}$ and $\Delta y=\frac{b_y-a_y}{M_y}$. The numerical approximation to the solution $u(x_i,y_j)$, $u_x(x_i,v_j)$ and $u_y(x_i,y_j)$ at the time t^n are denoted by $u_{i,j}^n$, $(u_x)_{i,j}^n$ and $(u_x)_{i,j}^n$, respectively.

Algorithm 4.1 The solver $Q_1(\Delta t)$ for the system (33).

Step 1 Apply the 1D hybrid HWENO method on (MOL_x^T) with BE time discretization or CN time discretization, obtaining $u_{i,j}^{n+1}$ and $(u_x)_{i,j}^{n+1}$.

Step 2 Apply the same time discretization on (FD_x). For instance, when using the BE method, we have that

$$\frac{(u_y)_{i,j}^{n+1}-(u_y)_{i,j}^n}{\Delta t}+\frac{\partial}{\partial y}(f(y,t^{n+1})u_x^{n+1})_{i,j}=0.$$

Or we can obtain the following semi-discrete scheme when employing the CN time discretization,

$$\frac{(u_y)_{i,j}^{n+1}-(u_y)_{i,j}^n}{\Delta t}+\frac{1}{2}\frac{\partial}{\partial y}(f(y,t^{n+1})u_x^{n+1}+f(y,t^n)u_x^n)_{i,j}=0.$$

Note that $(u_x)_{i,j}^{n+1}$ is already known via Step 1. Hence, we can discretize the spatial derivative by the central scheme

$$\frac{\partial}{\partial y}(f(y,t^*)u_x^*)_{i,j}=\frac{1}{2\Delta y}(f(y_{j+1},t^*)(u_x)_{i,j+1}^*-f(y_{j-1},t^*)(u_x)_{i,j-1}^*),$$

where $*$ can be n or $n+1$.

Note that the 1D1V VP system can be split into two systems as well

$$\left.\begin{aligned} &\frac{\partial u}{\partial t}+f(y,t)\frac{\partial u}{\partial x}=0,\\ &\frac{\partial u_x}{\partial t}+f(y,t)\frac{\partial u_x}{\partial x}=0,\\ &\frac{\partial u_y}{\partial t}+\frac{\partial(f(y,t)\,u_x)}{\partial y}=0 \end{aligned}\right\} \tag{36a}$$

and

$$\left.\begin{aligned} &\frac{\partial u}{\partial t}+g(x,t)\frac{\partial u}{\partial y}=0,\\ &\frac{\partial u_y}{\partial t}+g(x,t)\frac{\partial u_y}{\partial y}=0,\\ &\frac{\partial u_x}{\partial t}+\frac{\partial(g(x,t)u_y)}{\partial x}=0 \end{aligned}\right\} \tag{36b}$$

Similarly, the VP system can be approximated by the Strang splitting method. And the electrostatic field $E(x,t)$ is obtained by solving Poisson's equation when it is needed.

5 Numerical results

In this section, we present the numerical results with the schemes described in the previous sections. Both backward Euler (BE) scheme and Crank-Nicolson (CN) scheme are used in time discretization, coupling with the 4-point hybrid HWENO integration. For 2D problems the time steps are chosen as

$$\Delta t=\mathrm{CFL}/\max(|c_x|/\Delta x,|c_y|/\Delta y) \tag{37}$$

with $|c_x|=\max_y|f(y,t)|$ and $|c_y|=\max_x|g(x,t)|$. We want to remark that the fully discrete schemes are unconditionally stable. Hence, CFL can be arbitrary positive number.

5.1 Basic 2D test

Here, we consider the rigid body rotation problem.

$$u_t+yu_x-xu_y=0,\ (x,y)\in\Omega=[-\frac{1}{2}\pi,\frac{1}{2}\pi]^2 \tag{38}$$

with two different initial conditions:

① continuous initial condition:

$$u(x,y,0) = 0.5B(\sqrt{x^2+8y^2}) + 0.5B(\sqrt{8x^2+y^2}) \tag{39}$$

where

$$B(r) = \begin{cases} \cos(r)^6, & \text{if } r \leqslant \frac{1}{2}\pi, \\ 0, & \text{otherwise;} \end{cases}$$

Table 1. Under periodic boundary condition, errors and orders of accuracy for solving (38) with continuous initial condition (39) at $T=2\pi$.

		$M_x \times M_y$	L_1 errors	order	L_∞ error	order
BE	CFL=0.5	20×20	1.272E−02	−	1.613E−01	−
		40×40	7.792E−03	0.706	8.850E−02	0.866
		80×80	4.839E−03	0.687	5.823E−02	0.604
		160×160	2.830E−03	0.774	3.478E−02	0.743
		320×320	1.573E−03	0.847	1.938E−02	0.844
	CFL=1	20×20	1.514E−02	−	1.993E−01	−
		40×40	1.084E−02	0.482	1.144E−01	0.801
		80×80	7.563E−03	0.519	8.746E−02	0.387
		160×160	4.794E−03	0.658	5.796E−02	0.594
		320×320	2.825E−03	0.763	3.473E−02	0.739
	CFL=2	20×20	1.956E−02	−	2.782E−01	−
		40×40	1.423E−02	0.459	1.363E−01	1.029
		80×80	1.075E−02	0.405	1.157E−01	0.236
		160×160	7.530E−03	0.514	8.739E−02	0.405
		320×320	4.790E−03	0.652	5.793E−02	0.593
CN	CFL=0.5	20×20	6.555E−03	−	1.088E−01	−
		40×40	1.092E−03	2.586	2.399E−02	2.181
		80×80	1.811E−04	2.592	3.265E−03	2.877
		160×160	2.801E−05	2.692	4.791E−04	2.769
		320×320	4.913E−06	2.512	5.853E−05	3.033
	CFL=1	20×20	6.831E−03	−	1.063E−01	−
		40×40	1.283E−03	2.412	2.303E−02	2.206
		80×80	2.819E−04	2.187	3.131E−03	2.879
		160×160	6.233E−05	2.177	6.173E−04	2.342
		320×320	1.458E−05	2.096	1.505E−04	2.036
	CFL=2	20×20	8.588E−03	−	1.027E−01	−
		40×40	2.572E−03	1.739	3.250E−02	1.660
		80×80	7.561E−04	1.766	9.582E−03	1.762
		160×160	2.118E−04	1.836	2.401E−03	1.996
		320×320	5.504E−05	1.944	5.933E−04	2.017

Table 2. Under Dirichlet boundary condition, errors and orders of accuracy for solving (38) with continuous initial condition at $T=2\pi$.

		$M_x\times M_y$	L_1 errors	order	L_∞ error	order
BE	CFL=0.5	20×20	1.083E−02	−	1.177E−01	−
		40×40	7.500E−03	0.530	8.763E−02	0.426
		80×80	4.792E−03	0.646	5.777E−02	0.601
		160×160	2.824E−03	0.763	3.472E−02	0.735
		320×320	1.572E−03	0.845	1.937E−02	0.842
	CFL=1	20×20	1.338E−02	−	1.707E−01	−
		40×40	1.062E−02	0.333	1.147E−01	0.573
		80×80	7.530E−03	0.496	8.739E−02	0.393
		160×160	4.789E−03	0.653	5.793E−02	0.593
		320×320	2.825E−03	0.762	3.472E−02	0.739
	CFL=2	20×20	1.692E−02	−	2.629E−01	−
		40×40	1.384E−02	0.290	1.372E−01	0.938
		80×80	1.068E−02	0.374	1.158E−01	0.245
		160×160	7.519E−03	0.506	8.738E−02	0.406
		320×320	4.789E−03	0.651	5.793E−02	0.593
CN	CFL=0.5	20×20	6.550E−03	−	1.027E−01	−
		40×40	1.085E−03	2.593	2.399E−02	2.098
		80×80	1.799E−04	2.593	3.265E−03	2.877
		160×160	2.794E−05	2.687	4.791E−04	2.769
		320×320	2.794E−05	2.687	4.791E−04	2.769
	CFL=1	20×20	6.549E−03	−	1.008E−01	−
		40×40	1.286E−03	2.348	2.303E−02	2.130
		80×80	2.840E−04	2.180	3.131E−03	2.879
		160×160	6.263E−05	2.181	6.173E−04	2.342
		320×320	1.461E−05	2.100	1.505E−04	2.036
	CFL=2	20×20	8.494E−03	−	1.019E−01	−
		40×40	2.551E−03	1.736	3.239E−02	1.653
		80×80	7.823E−04	1.705	9.656E−03	1.746
		160×160	2.154E−04	1.861	2.402E−03	2.007
		320×320	5.536E−05	1.960	5.933E−04	2.017

② discontinuous initial condition:

$$u(x,y,0)=\begin{cases}1, & (x,y)\in[-\frac{3}{8}\pi,\frac{3}{8}\pi]\times[-\frac{1}{8}\pi,\frac{1}{8}\pi]\cup\\ & \quad[-\frac{1}{8}\pi,\frac{1}{8}\pi]\times[-\frac{3}{8}\pi,\frac{3}{8}\pi];\\ 0, & \text{otherwise.}\end{cases}$$

We simulate both problems with periodic boundary condition and zero Dirichlet boundary condition.

In Tables 1 and 2, we summarize the convergence study of the continuous problem at the final time $T=2\pi$, where the exact solution is as the same as the initial condition. The norm of the error is computed according to

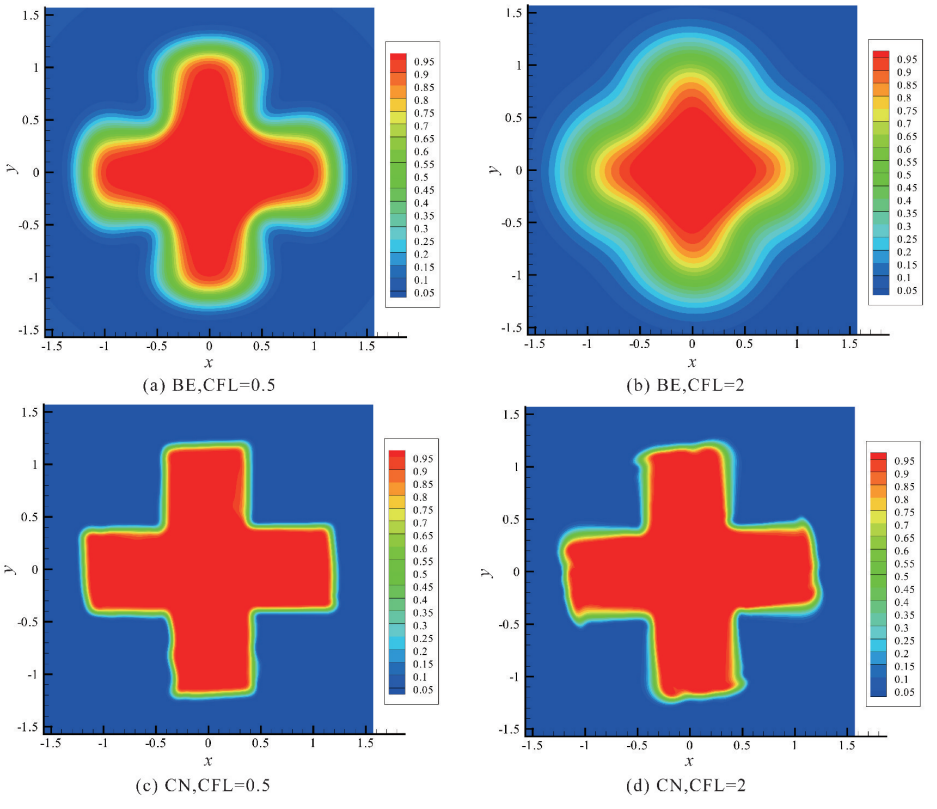

Figure 1. Under periodic boundary condition, numerical solutions for solving (38) with discontinuous initial condition at $T=2\pi$. 160×160 grid points.

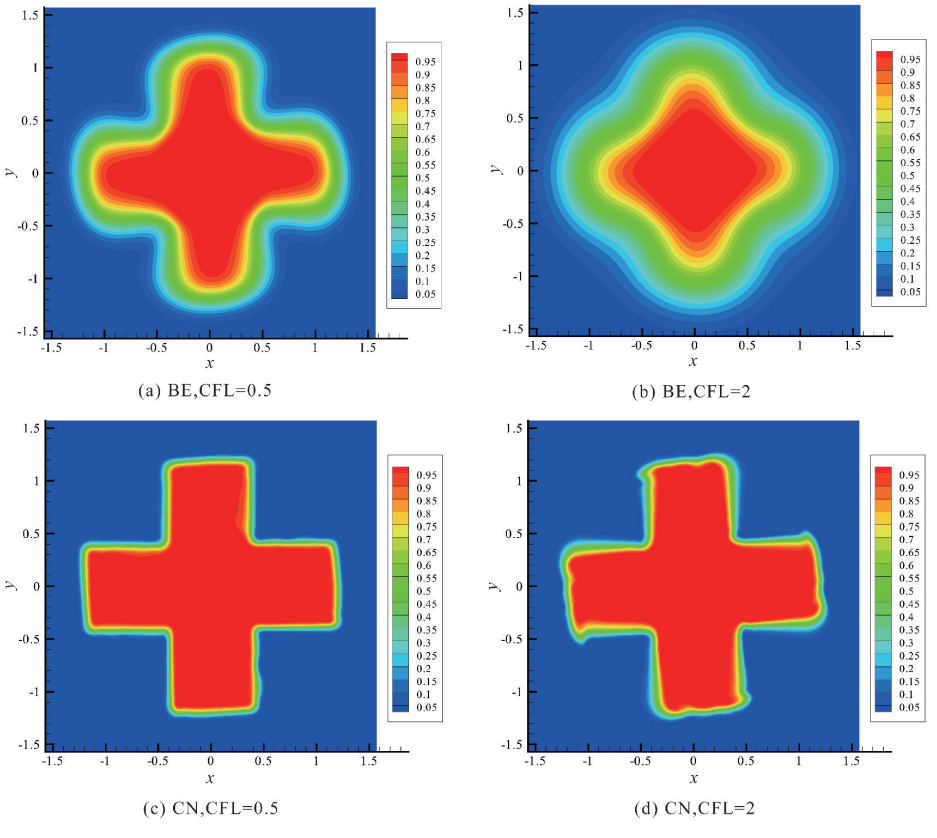

Figure 2. Under Dirichlet boundary condition, numerical solutions for solving (38) with discontinuous initial condition at $T=2\pi$. 160×160 grid points.

$$L_1 = \frac{1}{M_x \times M_y} \sum_{i=0}^{M_x} \sum_{j=0}^{M_y} | u_{i,j}^n - u^e(x_i, y_j, t) |,$$

$$L_\infty = \max_{0 \leqslant i \leqslant M_x,\ 0 \leqslant j \leqslant M_y} | u_{i,j}^n - u^e(x_i, y_j, t) |,$$

where $u^e(x_i, y_j, t)$ denotes the exact solution. We test both the BE scheme and the CN scheme with CFL = 0.5, 1, 2. We can see that all schemes are stable and converge to the designed order of accuracy. Moreover, when using CN scheme with CFL = 2, numerical results show higher order than 2. This because spatial errors play a leading role in errors due to the small time step, and the 4-point HWENO scheme is third order accuracy.

For the discontinuous problems, we test our scheme with CFL = 0.5 and 2. The HWENO methodology removes unphysical oscillations as expected (Figures 1 and 2). And the CN schemes give sharper interface than the BE schemes.

5.2 The VP systems

Here, we will consider the Vlasov-Poisson equation (1) with the following initial condition:

① Strong Landau damping:

$$f(x,v,0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha\cos(kx))\exp(-\frac{v^2}{2}),$$

$$x \in [0,L],\ v \in [-V_c, V_c],$$

where $\alpha = 0.5$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

② Two-stream instability I:

$$f(x,v,0) = \frac{2}{7\sqrt{2\pi}}(1 + 5v^2)(1 + \alpha((\cos(2kx) + \cos(3kx))/1.2 + \cos(kx)))\exp(-\frac{v^2}{2}),$$

$$x \in [0,L],\ v \in [-V_c, V_c],$$

where $\alpha = 0.01$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

③ Two-stream instability II:

$$f(x,v,0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha\cos(kx))v^2\exp(-\frac{v^2}{2}),$$

$$x \in [0,L],\ v \in [-V_c, V_c],$$

where $\alpha = 0.05$, $k = 0.5$, $L = 4\pi$ and $V_c = 2\pi$.

④ Bump-on-tail instability:

$$f(x,v,0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha\cos(kx))\ (0.9\exp(-0.5v^2) + 0.2\exp(-4(v-4.5)^2)),$$

$$x \in [-L,L],\ v \in [-V_c, V_c],$$

where $\alpha = 0.04$, $k = 0.3$, $L = \frac{10}{3}\pi$ and $V_c = 10$.

In our numerical tests, the periodic boundary condition is imposed in x-direction and the zero boundary condition is imposed in v-direction. A fast Fourier transform (FFT) is used to solve the 1D Poisson equation. We only test the VP systems with the CN scheme and CFL = 1. Numerical results are plotted in Figures 3−6. It is observed that solution will develop filamentation solution structures after a long time evolution, and the scheme is able to effectively capture those structures without producing spurious oscillations.

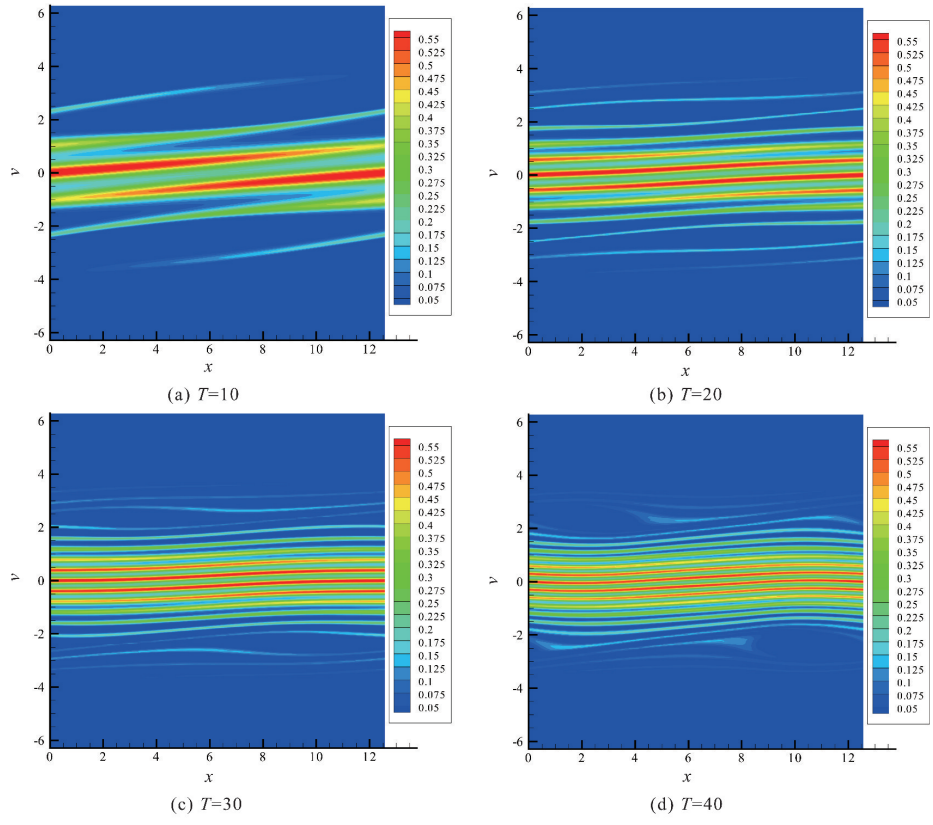

Figure 3. Strong Landau damping. 128×256 grid points.

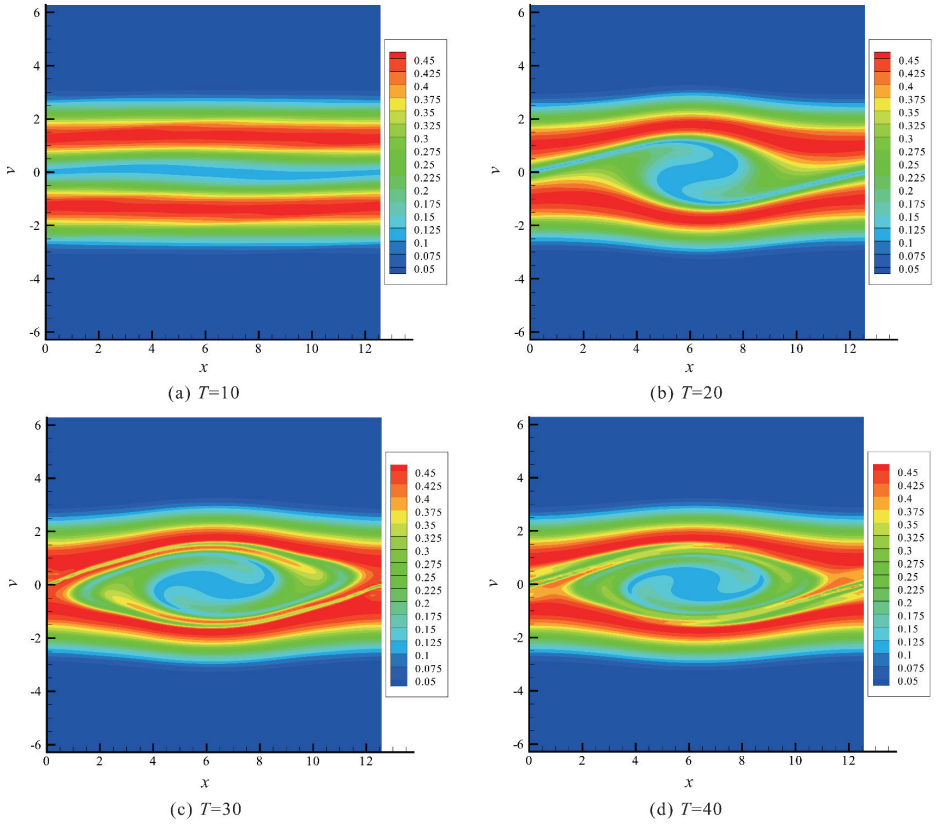

Figure 4. Two-stream instability I. 128×256 grid points.

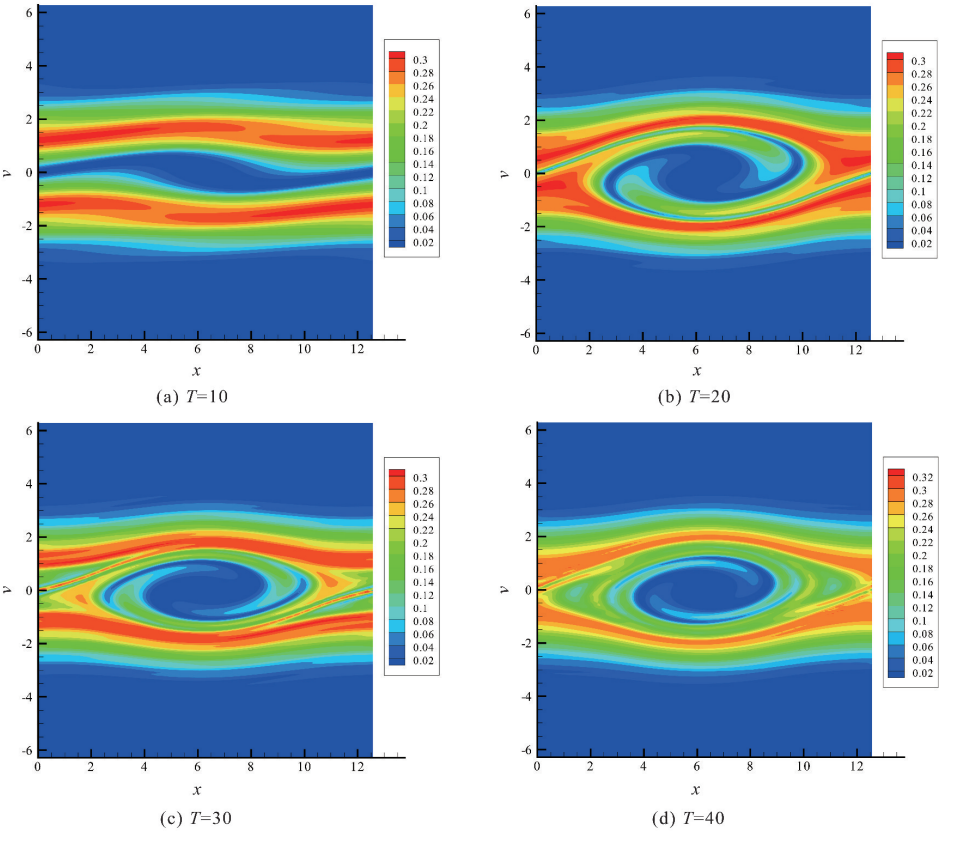

Figure 5. Two-stream instability II. 128×256 grid points.

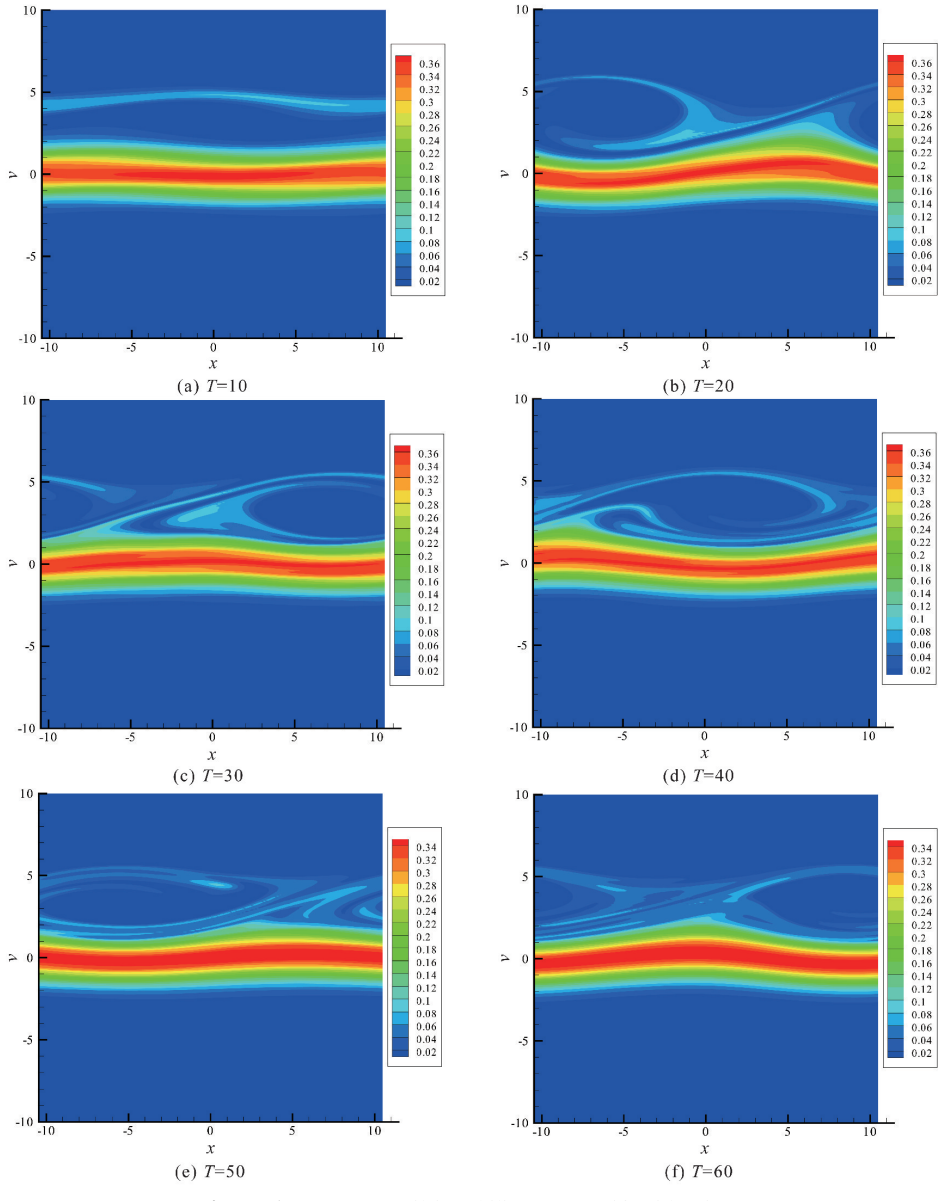

Figure 6. Bump-on-tail instability. 128×256 grid points.

6 Conclusions

In this paper, we proposed a new type hybrid Hermite weighted essentially non-oscillatory (HWENO) schemes coupled with the implicit method of lines transpose (MOL^T) framework for the Vlasov-Poisson (VP) system for plasma simulations. The main idea of the hybrid HWENO scheme is that if all extreme points of the big interpolation polynomial and its derivative are located outside of the big stencil, then we will use linear approximation straightforwardly to increase the efficiency of the scheme, otherwise use the HWENO procedure to avoid the oscillations nearby discontinuities. The algorithm was extended to multi-dimensional problems and the VP system via dimensional splitting. Comparing them with the WENO-based MOL^T schemes[23], the new HWENO-based MOL^T schemes are more efficient and compact, hence, less ghost points are needed near the boundary. In this paper, we consider the first order backward Euler scheme and the second order Crank-Nicolson scheme, coupling with a 4-points HWENO integration, which are unconditionally stable. A collection of numerical tests demonstrated good performance of the proposed scheme. Higher order numerical scheme will be considered in the future.

Acknowledgments

The work is supported by National Natural Science Foundation of China(Nos. 11901555, 11871448).

Conflict of interest

The authors declare no conflict of interest.

Author information

WANG Kaipeng is currently a PhD candidate under the tutelage of Prof. Jiang Yan and Prof. Zhang Mengping at University of Science and Technology of China. His research interests focus on numerical solution of partial differential equation and big data science.

JIANG Yan (corresponding author) is a Research Professor at School of Mathematical Sciences, University of Science and Technology of China (USTC). She received her PhD degree in computational mathematics in 2015 from USTC. From 2015 to 2018, she worked as a Postdoctoral Fellow at Michigan State University. In 2018, she joined USTC. Her research area is numerical analysis and scientific computing.

References

[1] Zhou T, Guo Y, Shu C W. Numerical study on Landau damping. Physica D: Nonlinear Phenomena, 2001, 157(4): 322-333.

[2] Filbet F, Sonnendrücker E. Comparison of Eulerian Vlasov solvers. Computer Physics Communications, 2003, 150(3): 247-266.

[3] Cheng Y, Gamba I M, Morrison P J. Study of conservation and recurrence of Runge-Kutta discontinuous Galerkin schemes for Vlasov-Poisson systems. Journal of Scientific Computing, 2013, 56, 319-349.

[4] Cheng Y, Christlieb A J, Zhong X. Numerical study of the two-species Vlasov-Ampère system: Energy-conserving schemes and the current-driven ion-acoustic instability. Journal of Computational Physics, 2015, 288: 66-85.

[5] Cai X, Guo W, Qiu J M. A high order semi-Lagrangian discontinuous Galerkin method for Vlasov-Poisson simulations without operator splitting. Journal of Computational Physics, 2018, 354: 529-551.

[6] Begue M, Ghizzo A, Bertrand P. Two-dimensional Vlasov simulation of Raman scattering and plasma beatwave acceleration on parallel computers. Journal of Computational Physics, 1999, 151(2): 458-478.

[7] Besse N, Segré J, Sonnendrücker E. Semi-Lagrangian schemes for the two-dimensional Vlasov-Poisson system on unstructured meshes. Transport Theory and Statistical Physics, 2005, 34(3-5): 311-332.

[8] Cheng C, Knorr G. The integration of the Vlasov equation in configuration space. Journal of Computational Physics, 1976, 22(3): 330-351.

[9] Christlieb A, Guo W, Morton M, et al. A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations. Journal of Computational Physics, 2014, 267: 7-27.

[10] Crouseilles N, Mehrenberger M, Sonnendrücker E. Conservative semi-Lagrangian schemes for Vlasov equations. Journal of Computational Physics, 2010, 229(6): 1927-1953.

[11] Crouseilles N, Respaud T, Sonnendrücker E. A forward semi-Lagrangian method for the numerical solution of the Vlasov equation. Computer Physics Communications, 2009, 180(10): 1730-1745.

[12] Filbet F, Sonnendrücker E, Bertrand P. Conservative numerical schemes for the Vlasov equation. Journal of Computational Physics, 2001, 172(1): 166-187.

[13] Guo W, Qiu J M. Hybrid semi-Lagrangian finite element-finite difference methods for the Vlasov equation. Journal of Computational Physics, 2013, 234: 108-132.

[14] Parker G J, Hitchon W N G. Convected scheme simulations of the electron distribution function in a positive column plasma. Japanese Journal of Applied Physics, 1997, 36(7S): 4799.

[15] Qiu J M, Shu C W. Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov-Poisson system. Journal of Computational Physics, 2011, 230(23): 8386-8409.

[16] Rossmanith J, Seal D. A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. Journal of Computational Physics, 2011, 230(16): 6203-6232.

[17] Sonnendrücker E, Roche J, Bertrand P, et al. The semi-Lagrangian method for the numerical resolution of the Vlasov equation. Journal of Computational Physics, 1999, 149(2): 201-220.

[18] Birdsall C K, Langdon A B. Plasma Physics Via Computer Simulaition. Boca Raton, FL: CRC Press, 2005.

[19] Verboncoeur J P. Particle simulation of plasmas: Review and advances. Plasma Physics and Controlled Fusion, 2005, 47(5A): A231-A260.

[20] Hockney R W, Eastwood J W. Computer Simulation Using Particles. Boca Raton, FL: CRC Press, 2010.

[21] Barnes J, Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm. Nature, 1986, 324: 446-449.

[22] Evstatiev E G, Shadwick B A. Variational formulation of particle algorithms for kinetic plasma simulations. Journal of Computational Physics, 2013, 245: 376-398.

[23] Christlieb A, Guo W, Jiang Y. A WENO-based method of lines transpose approach for Vlasov simulations. Journal of Computational Physics, 2016, 327: 337-367.

[24] Schemann M, Bornemann F. An adaptive Rothe method for the wave equation. Computing and Visualization in Science, 1998, 1(3): 137-144.

[25] Salazar A J, Raydan M, Campo A. Theoretical analysis of the exponential transversal method of lines for the diffusion equation. Numerical Methods for Partial Differential Equations, 2000, 16(1): 30-41.

[26] Causley M F, Cho H, Christlieb A J, et al. Method of lines transpose: High order L-stable $\mathscr{O}(N)$ schemes for parabolic equations using successive convolution. SIAM Journal on Numerical Analysis, 2016, 54(3): 1635-1652.

[27] Cheng Y, Christlieb A J, Guo W, et al. An asymptotic preserving Maxwell solver resulting in the Darwin limit of electrodynamics. Journal of Scientific Computing, 2017, 71(3): 959-993.

[28] Liu H, Qiu J. Finite difference Hermite WENO schemes for hyperbolic conservation laws. Journal of Scientific Computing, 2015, 63(2): 548-572.

[29] Liu H, Qiu J. Finite difference Hermite WENO schemes for conservation laws, II: An alternative approach. Journal

of Scientific Computing, 2016, 66(2): 598-624.
[30] Zhao Z, Zhu J, Chen Y, et al. A new hybrid WENO scheme for hyperbolic conservation laws. Computers & Fluids, 2019, 179: 422-436.
[31] Qiu J, Shu C W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: One-dimensional case. Journal of Computational Physics, 2004, 193(1): 115-135.
[32] Zhao Z, Chen Y, Qiu J. A hybrid Hermite WENO scheme for hyperbolic conservation laws. Journal of Computational Physics, 2020, 405: 109175.
[33] Zhu J, Qiu J. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method, III: Unstructured meshes. Journal of Scientific Computing, 2009, 39(2): 293-321.
[34] Qiu J, Shu C W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case. Computers & Fluids, 2005, 34(6): 642-663.
[35] Zhu J, Qiu J. A new type of modified WENO schemes for solving hyperbolic conservation laws. SIAM Journal on Scientific Computing,2017, 39(3): A1089-A1113.
[36] Zhu J, Shu C W. A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. Journal of Computational Physics, 2018, 375: 659-683.

用于 Vlasov 模拟的一类基于混合 HWENO 的转秩直线法

王恺鹏,蒋琰*,张梦萍
中国科学技术大学数学科学学院,安徽合肥 230026
* 通讯作者. E-mail: jiangy@ustc.edu.cn

摘要: 在隐式转秩直线法(MOL^T)框架下,针对一维线性输运方程设计了一类新型的混合 Hermite 加权本质无震荡(HWENO)格式,并进一步用于求解 Vlasov-Poisson (VP) 方程组. 相较于之前的基于加权本质无震荡(WENO)的 MOL^T 方法[J. Comput. Phys., 2016,327:337-367],该新方法主要有两个优点:第一,在满足相同精度的情况下,HWENO 格式比 WENO 格式使用更窄的模版;第二,该方法可以自动调节选取线性格式或 HWENO 格式. 因此,混合 HWENO 方法既保持了 WENO 方法的简便性和鲁棒性,同时,又能在光滑区域减小计算误差,降低计算时间,提高计算效率. 我们将设计的算法用于模拟一系列基准算例,以展示其鲁棒性和高效性.

关键词: 转秩直线法;隐式时间离散;HWEMO 方法;混合方法;Vlasov-Poisson 方程组