SIAM J. SCI. COMPUT.
Vol. 33, No. 6, pp. 3509–3537

© 2011 Society for Industrial and Applied Mathematics

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

# A PARTICLE-IN-CELL METHOD WITH ADAPTIVE PHASE-SPACE REMAPPING FOR KINETIC PLASMAS[*]

B. WANG[†], G. H. MILLER[‡], AND P. COLELLA[§]

**Abstract.** We present a new accurate and efficient particle-in-cell (PIC) method for computing the dynamics of one-dimensional kinetic plasmas. The method overcomes the numerical noise inherent in particle-based methods by periodically remapping the distribution function on a hierarchy of locally refined grids in phase space. Remapping on phase-space grids also provides an opportunity to integrate a collisional model and an associated grid-based solver. The positivity of the distribution function is enforced by redistributing excess phase-space density in a local neighborhood. We demonstrate the method on a number of standard plasma physics problems. It is shown that remapping significantly reduces the numerical noise and results in a more consistent second-order method than the standard PIC method. An error analysis is presented which is based on prior results of Cottet and Raviart's work [*SIAM J. Numer. Anal.*, 21 (1984), pp. 52–76].

**Key words.** particle methods, adaptive mesh refinement, remapping, Vlasov–Poisson equation

**AMS subject classifications.** 35, 65, 76

**DOI.** 10.1137/100811805

## 1. Introduction.

Numerical solutions of the Vlasov equation play a significant role in understanding the nonlinear behavior of plasmas, which are of great interest in controlled thermonuclear fusion and accelerator modeling. Numerous numerical schemes have been developed, and generally they can be divided into two categories: particle methods and grid methods [24, 15, 19].

Particle methods have been used to simulate kinetic plasmas with much success [24, 6]. In these methods, the particles, a Lagrangian discretization of the distribution function, follow trajectories derived from the Vlasov equation. In the case of electrostatic forces, self-consistent fields are calculated via direct solution of Coulomb's law. At any time, the distribution function can be reproduced by summing a set of discrete delta functions in phase space represented by these particles. Particle methods are naturally adaptive, since particles occupy only the domain where the distribution function is not zero. However, the numerical errors in particle methods lead to the disorder of the particles, often referred to as *numerical noise*. The loss of accuracy can be explained by the convergence theorem for particle methods, which was first derived for vortex methods in fluid dynamics [22, 3, 4, 1, 37, 31] and later extended to particle methods in kinetic plasma simulation [13, 20, 44]. In short, the theorem says that particle overlap and regularization are necessary for convergence.

[*] Submitted to the journal's Computational Methods in Science and Engineering section October 15, 2010; accepted for publication (in revised form) September 15, 2011; published electronically December 15, 2011. This work was supported by the U.S. Department of Energy Office of Advanced Scientific Computing Research under contract DE-AC02-05CH11231 at the Lawrence Berkeley National Laboratory.

http://www.siam.org/journals/sisc/33-6/81180.html

[†] Department of Applied Science, University of California, 1 Shields Avenue, Davis, CA 95616 (beiwang@ucdavis.edu).

[‡] Department of Chemical Engineering and Materials Science, University of California, 1 Shields Avenue, Davis, CA 95616 (grgmiller@ucdavis.edu). The research of this author was supported by DOE grant DE-SC0001981.

[§] Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720 (colella@hpcrdm.lbl.gov).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

An alternative is to employ grid methods to solve the Vlasov equation on a grid in phase space. Such methods are free of particle noise and it is known how to achieve a high order of accuracy. The Vlasov equation is simply a nonlinear advection equation in high-dimensional space, so methods of solution can be guided by the well-established numerical analysis of classical partial differential equations. Grid-based methods for the Vlasov equation have developed mainly along three lines: transform methods, such as Fourier–Hermite transform [2] and Fourier–Fourier transform [26]; finite-volume methods, such as the flux balance method and the positive and flux conservative method [17, 18, 11]; and semi-Lagrangian methods, such as the cubic spline interpolation method, [9, 39], the cubic interpolated propagation method [35], and the forward semi-Lagrangian method [14]. Grid-based methods are accurate for problems with large dynamic ranges for which standard particle methods fail. Their chief drawback is the very high cost of grid-based discretization in high-dimensional spaces.

In this paper, we present a new particle-in-cell (PIC) method based on periodically remapping the particle distribution on phase space. Such remapping has been used in vortex methods and smoothed particle hydrodynamics (SPH) to maintain regularity of the particle distribution and thereby improve accuracy [12, 7] but has not been widely used with the PIC method for plasma physics. The remapping scheme we have developed has two new features which contribute to accuracy, stability, and efficiency. First, we use high-order interpolation stencils which improve accuracy but do not preserve positivity. We propose two approaches to enforce the positivity of the distribution function. Both of these methods conserve total charge and have the property that a locally positive distribution remains unchanged. Second, we use remapping in concert with a hierarchy of locally refined grids. Remapping on a uniform grid tends to create a large number of particles with very small strength in the tail of the distribution function. The situation becomes worse with repeated remapping. We alleviate the problem by using locally refined grids. The reconstruction of the distribution function on grids through remapping also provides an opportunity to introduce a collisional model with a grid-based solver. Accordingly, we are able to describe a kinetic plasma under the weakly collisional domain, where the collisionless part is solved by the PIC method and the collisional part is solved by a grid-based method, e.g., a second-order finite-volume method.

In the domain of particle methods, the remapped PIC method is similar to Denavit's [15] in hybridizing a particle method and a grid-based remapping. However, in their papers, remapping is used to introduce numerical diffusion to the system to suppress fine structures created in velocity space, so they use low-order interpolation functions. Vadlamani et al. [43] developed a hybrid scheme as an extension to Denavit's. In the domain of grid methods, the algorithm is similar to the forward semi-Lagrangian method (FSLM) of Crouseilles, Respaud, and Sonnendrucker [14] in that both methods advance particles along the characteristics forward in time and periodically reconstruct the distribution in phase space. Compared with the FSLM, our interpolation is conservative, high-order, and positive without solving a linear system. In addition, mesh refinement is introduced in remapping to reduce the total number of small-strength particles.

The sections below are organized as follows. In section 3, we first review standard particle methods, e.g., the PIC method, for the solution of the Vlasov–Poisson equations. Then error bounds in approximating the Vlasov–Poisson equations using the PIC method are given. The error analysis is presented in Appendix A. In section 4, we describe the conservative, positive, and high-order remapping algorithm on a hierarchy of locally refined grids. A collision model, described by the simplified

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Fokker–Planck equation by Rathmann and Denavit [36], is presented in section 5. We demonstrate the accuracy of the method through a series of standard numerical tests in section 6. Some discussion of issues concerning implementation of the method in high dimensions is presented at the end.

## 2. The Vlasov–Poisson system.

We consider the dimensionless Vlasov–Poisson equation in one dimension. That is,

$$\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} - E \cdot \frac{\partial f}{\partial v} = 0 \tag{2.1}$$

and

$$\nabla \cdot E = -\Delta\phi = 1 - \int_{\mathbb{R}} f dv, \tag{2.2}$$

where $f(x, v, t)$ is the electron distribution function in phase space $(x, v)$, $E(x, t)$ is the self-consistent electrostatic field, and $\phi(x, t)$ is the potential. The ions form a fixed and uniform background. It is convenient to define the charge density $\rho(x, t)$ as

$$\rho = 1 - \int_{\mathbb{R}} f dv. \tag{2.3}$$

We complete the system by assuming that (1) boundary conditions are periodic in physical space $x \in (0, L)$ and (2) the initial distribution function $f(x, v, 0)$ has continuous, bounded first derivative and satisfies the conditions

$$|f(x, v, 0)| \leq c(1 + |v|)^{-\lambda}, \quad \lambda > 1, \quad x, v \in \mathbb{R}, \tag{2.4}$$

and

$$\frac{1}{L}\int_0^L \int_{-\infty}^{\infty} f(x, v, 0) dx dv = 1. \tag{2.5}$$

## 3. Particle methods.

### 3.1. Particle methods.

Particle methods are based on the Lagrangian description of (2.1). We first discretize the initial distribution function with a set of particles laid out on cell centers of a Cartesian grid in phase space,

$$f(x, v, 0) = \sum_k q_k \delta(x - x_k)\delta(v - v_k), \tag{3.1}$$

where $q_k = h_x h_v f(x_k, v_k, 0)$ denotes the weight of particle $k$, and $(x_k, v_k)$ denotes the cell center of the phase-space grid. $h_x$ and $h_v$ are the particle mesh spacing in physical space and velocity space, respectively. We assume that $h_x = O(h_v)$.

Each particle $k$ then follows the trajectory of the flow starting at $(\tilde{X}_k(0) = x_k, \tilde{V}_k(0) = v_k)$ with

$$\begin{aligned} \frac{dq_k}{dt}(t) &= 0, \\ \frac{d\tilde{X}_k}{dt}(t) &= \tilde{V}_k(t), \\ \frac{d\tilde{V}_k}{dt}(t) &= -\tilde{E}_k(t), \end{aligned} \tag{3.2}$$

where $(\tilde{X}_k(t), \tilde{V}_k(t))$ is the computed trajectory and $\tilde{E}_k(t)$ is the field induced by all particles. These equations are then advanced in time through numerical integration.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

At any time, the distribution function can be reproduced through a sum of discrete delta functions represented by the particles in phase space. That is,

$$\tilde{f}(x,v,t) = \sum_k q_k \delta_{\varepsilon_x}(x - \tilde{X}_k(t))\delta_{\varepsilon_v}(v - \tilde{V}_k(t)). \tag{3.3}$$

The discrete delta functions have compact support and are normalized. For example,

$$\int_{\mathbb{R}} \delta_{\varepsilon_x}(x)dx = 1 \tag{3.4}$$

and

$$\delta_{\varepsilon_x}(x) = \frac{1}{\varepsilon_x} u\left(\frac{x}{\varepsilon_x}\right),$$

where $u$ is any interpolation function with stencil size $\varepsilon_x$.

The self-consistent field at $t > 0$ can be obtained by convolution as

$$\begin{aligned} \tilde{E}(x,t) &= \int_0^L K(x-y)\left[1 - \int_{-\infty}^{\infty} \tilde{f}(y,v,t)dv\right] dy \\ &= \int_0^L K(x-y)dy - \sum_k q_k \int_0^L K(x-y)\delta_{\varepsilon_x}(y - \tilde{X}_k(t))dy, \end{aligned} \tag{3.5}$$

where $K(x-y) = -\frac{\partial G}{\partial x}(x,y)$ and $G(x,y)$ is the Green's function for the one-dimensional Laplacian operator with periodic boundary conditions.

### 3.2. The particle-in-cell method.

There are many ways to approximate the self-consistent fields (3.5) in particle methods, e.g., fast multipole methods [21]. Here we use the PIC method [24, 6], where the charge density and the electric field are defined and solved on a grid. The use of a grid for the Poisson equation accelerates the method by fast Poisson solvers, such as fast Fourier transformations (FFTs) or multigrid methods. The PIC method reduces the complexity of pure particle methods from $O(N \log N)$ to $O(M \log M + N)$, where $N$ is the number of particles and $M$ is the number of mesh points. Usually, $M$ is much smaller than $N$. Compared to FFTs, multigrid methods are easier to use with mesh refinement. One time step of the PIC algorithm is processed as follows:

1. Calculate the physical space charge density on a grid. This includes (i) represent the distribution function using a discrete delta function, e.g., B-spline of order two, and integrate it along velocity space; and (ii) get the charge density value on the grid. Usually, the mesh spacing of the grid is chosen to be the same as the discrete delta stencil size, $\Delta x = \varepsilon_x$. Here and in what follows, $\Delta x$ and $\varepsilon_x$ are interchangeable. The charge assignment procedure can be represented as

$$\tilde{\rho}(x_j,t) = 1 - \int_{-\infty}^{\infty} \sum_k q_k \delta_{\Delta x}(x_j - \tilde{X}_k(t))\delta_{\varepsilon_v}(v - \tilde{V}_k(t))dv \tag{3.6}$$

$$= 1 - \sum_k q_k \delta_{\Delta x}(x_j - \tilde{X}_k(t)) \tag{3.7}$$

and

$$\delta_{\Delta x}(x) = \frac{1}{\Delta x} u_1\left(\frac{x}{\Delta x}\right), \tag{3.8}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where

$$u_1(y) = \begin{cases} 1 - |y|, & 0 \le |y| \le 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

$j$ is the grid node index.

2. Solve the Poisson equation on the grid with a second-order finite-difference method

$$-\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2} = \tilde{\rho}_j \tag{3.10}$$

$$-\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} = \tilde{E}_j. \tag{3.11}$$

3. Interpolate the grid-based fields to the particle location with the same interpolation function (3.9).
4. Numerically integrate the particle equations of motion. We use the fourth-order Runge–Kutta method.

**3.3. The accuracy of particle methods for the one-dimensional Vlasov–Poisson system.** A rigorous study of particle methods for the one-dimensional Vlasov–Poisson system has been performed by Cottet and Raviart [13]. The convergence analysis shows that the accuracy depends on a number of factors such as the initial particle discretization of the system and the choice of the discrete delta function as well as the specific physical problem. We extend the approach to the PIC method for the one-dimensional Vlasov–Poisson equation. Our analysis differs from Cottet and Raviart's in the way we estimate the consistency error of the electric field. Instead of estimating the consistency error of the electric field by convolution, we treat it as the field induced by the consistency error of the charge density. This enables the use of the smooth interpolation analysis by Schoenberg [38, 32] instead of the quadrature rules for the discretization error (the consistency error includes the moment error and the discretization error). Compared to the quadrature rules, the smooth interpolation analysis results in a higher-order error bound for the same interpolation function, e.g., the B-spline of order two ($M_2$). The stability error of the electric field is estimated by using convolution as in [13]. The derivation of the error estimates for the PIC method is presented in Appendix A.

Here we summarize the error bound for the electric field. Assuming that $h_x = O(h_v)$, when $0 \le t \le T$ for $T > 0$, we have

$$\begin{aligned} e^E(x,t) &\\ &= |E(x,t) - \tilde{E}(x,t)| \\ &\le C(T)\left(\underbrace{\Delta x^2 + \Delta x^2\left(\frac{h_x}{\Delta x}\right)^2}_{\sim e_c^E(x,t)} + \underbrace{\left(\Delta x^2 + \Delta x^2\left(\frac{h_x}{\Delta x}\right)^2\right)(\exp(at)-1)}_{\sim e_s^E(x,t)}\right), \end{aligned} \tag{3.12}$$

where $a$ depends on $\|\frac{\partial E}{\partial x}(\cdot,t)\|_{L^\infty(\mathbb{R})}$. Parameters $C(T)$ are independent of $\Delta x$ and $h_x$. The subscript $c$ and $s$ denote the consistency error and the stability error, respectively. Compared to Cottet and Raviart's results (equation 4.23 of [13]), our error estimate is one order higher. For example, the consistency error of the electric field in their paper is $O(\Delta x^2 + h_x(\frac{h_x}{\Delta x})^2)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The consistency error for the PIC method depends on the discrete delta function, $\delta_{\Delta x}$, assuming that the charge density function is smooth (A.1). It is very important that the initial particle mesh spacing in physical space, $h_x$, be less than or equal to the Poisson solver mesh spacing, $\Delta x$. If the B-spline of order two (3.9) is used as the discrete delta function, the consistency error is second order. The stability error is amplified by a time dependent term $\exp(at) - 1$.

Equation (3.12) provides guidance for the design of particle methods in solving the Vlasov–Poisson equation and for the choice of optimal parameters. First, for convergence, the ratio of the initial particle mesh spacing in physical space and the Poisson solver mesh spacing needs to be bounded such that $\frac{h_x}{\Delta x} \leq 1$. This is the so-called *overlapping* condition in particle methods. Second, we notice that the consistency error is amplified by a time dependent term. This implies that particle methods can lose accuracy for some problems, e.g., the problem with large dynamic ranges. It is very important that we have some mechanism in particle methods to control this factor. Section 4 provides such a mechanism.

**4. Remapping.** One strategy for controlling stability error is to periodically restart the problem by representing the distribution function on a uniform grid and creating a new set of particles from that. This method, called remapping or regridding, has been used extensively in particle methods, such as vortex methods and SPH, for improving accuracy in fluid dynamics [12, 27, 7], and to a much more limited extent in the PIC method in plasma physics [15, 43, 8].

Assuming that we reconstruct the particle distribution on a uniform grid through interpolation, the error in replacing the old particle distribution with a new set of particles on the cell center of a grid will depend on the accuracy of the interpolation function $u$. For simplicity, the analysis in this section is for remapping in one dimension. In real simulations, we apply remapping in all dimensions in phase space, i.e., physical space and velocity space, using tensor product formulas. This error analysis is performed by Cottet and Koumoutsakos [12] as follows. If the new and old particle positions are denoted by $x_{\boldsymbol{i}}$ and $x_k$ and their charges are denoted by $q_{\boldsymbol{i}}$ and $q_k$, respectively, the new particle charge value through interpolation is

$$q_{\boldsymbol{i}} = \sum_k q_k u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right), \tag{4.1}$$

where $u$ is a interpolation function. The error in particle represented value, for example, the charge density, by replacing the old particle distribution with the new one is then

$$e(x) = \sum_k q_k \delta_{\Delta x}(x_k - x) - \sum_{\boldsymbol{i}} q_{\boldsymbol{i}} \delta_{\Delta x}(x_{\boldsymbol{i}} - x). \tag{4.2}$$

Substituting (4.1) to the above equation, we get

$$\begin{align}
e(x) &= \sum_k \left[ q_k \delta_{\Delta x}(x_k - x) - \left( \sum_{\boldsymbol{i}} q_k u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right)\right) \delta_{\Delta x}(x_{\boldsymbol{i}} - x)\right] \tag{4.3}\\
&= \sum_k q_k \left[ \delta_{\Delta x}(x_k - x) - \sum_{\boldsymbol{i}} u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) \delta_{\Delta x}(x_{\boldsymbol{i}} - x)\right] \tag{4.4}\\
&= \sum_k q_k \sum_{\boldsymbol{i}} \left[\delta_{\Delta x}(x_k - x) - \delta_{\Delta x}(x_{\boldsymbol{i}} - x)\right] u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) \tag{4.5}\\
&\leqq \sum_k q_k \sum_{\boldsymbol{i}} \sum_{\alpha} ((x_{\boldsymbol{i}} - x_k) \cdot \nabla \delta_{\Delta x})^{\alpha} u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right). \tag{4.6}
\end{align}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We see that if

$$\sum_{\boldsymbol{i}} (x_{\boldsymbol{i}} - x_k)^{\alpha} u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) = 0, \quad 1 \leqslant |\alpha| \leqslant m-1, \tag{4.7}$$

then

$$e(x) = O(h_x^m). \tag{4.8}$$

This is the discrete analog of moment conditions in Appendix A (equation (A.5)).

The fundamental interpolation analysis is due to Schoenberg [38]. He considers the interpolation function in the form

$$\tilde{\psi}(x, h) = \sum_{i=-\infty}^{\infty} \psi(x_i) u\left(\frac{x_i - x}{h_x}\right), \tag{4.9}$$

where $\psi$ is the interpolated function and $u$ is the interpolating kernel. If $\tilde{\psi}(x_i) = \psi(x_i)$, $u$ is called ordinary interpolation, whereas if $\tilde{\psi}(x_i) \neq \psi(x_i)$, $u$ is called smooth interpolation. In particle methods, a smooth interpolation formula, e.g., a B-spline, is often used, since ordinary interpolation functions usually don't have a high order of smoothness and so tend to introduce large errors in the interpolated value if the particle position has large fluctuation. However, all B-splines are limited to only second-order accuracy. For example, if a B-spline type of interpolation function such as $u_1$ (equation (3.9)) is used, the error will be $O(h_x^2)$. Assuming $h_x = O(h_v) = O(dt)$, where $dt$ is the PIC step size, $u_1$ only results in a first-order method overall since we lose one order of accuracy in the evolution step. To achieve a second-order method, the remapping error should be at least third order.

Monaghan [33] presented a systematic way to increase the accuracy of a interpolation function based on B-splines while maintaining the smoothness of the function. Those high-order smooth functions are called modified B-splines. He suggested creating a modified B-spline by

$$W(y) = \frac{1}{2}(3M + yM'), \tag{4.10}$$

where $M$ is a B-spline and $M'$ is its derivative. For example, we can use the B-spline of order four to construct a interpolation function with third-order accuracy, where $m = 3$. The modified B-spline $W_4$ is defined as (see Figure 1)

$$W_4(y) = \begin{cases} 1 - \frac{5|y|^2}{2} + \frac{3|y|^3}{2} & \text{if } 0 \leq |y| \leq 1, \\ \frac{1}{2}(2 - |y|)^2(1 - |y|) & \text{if } 1 \leq |y| \leq 2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

The first- and second-order derivatives of $W_4$ are continuous. In addition, it has interpolation error of $O(h_x^3)$. $W_4$ has been used extensively in SPH for interpolation [32, 34]. It is also widely used in both vortex methods and SPH for remapping [12, 27, 7].

High-order interpolation functions are not positivity preserving. This is easy to be proved by observing the moment conditions (equation (A.5)). From the physics point of view, an interpolation function without positivity preservation might create nonphysical negative charge. This should be avoided in simulation. We modify the result of this interpolation step locally to produce a nonnegative distribution function.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

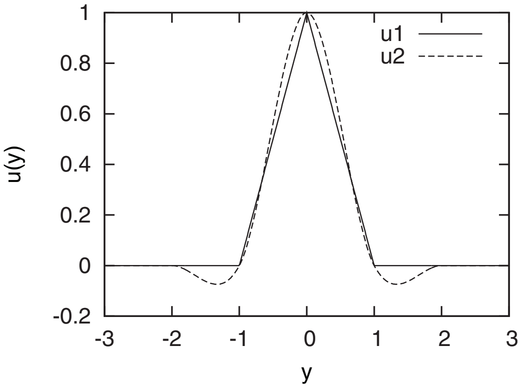


FIG. 1. *The linear and quadratic interpolation functions.* u1*: linear interpolation function, the B-spline of order two (equation* 3.9*).* u2*: quadratic interpolation function, the modified B-spline $W_4$ (equation* 4.11*).*

**4.1. Remapping with positivity preservation.** We design positivity-preserving algorithms based on the observation that the distribution function is conserved. If the value in a cell is less than zero (or exceeds unity), it must be the case that the net flux flowing out of (or in to) the cell is too large. This motivates a scheme based on flux-corrected transport.

**4.1.1. Global positivity preserving algorithm.** Flux-corrected transport (FCT) is a technique used to preserve positivity and monotonicity in Godunov methods for hyperbolic conservation laws. To apply this method to the remapping algorithm, we first represent the remapping operation as a flux difference

$$f_{\boldsymbol{i}}^{n+1} = f_{\boldsymbol{i}}^{n} + \nabla \cdot \boldsymbol{F}. \tag{4.12}$$

We can think of $f_{\boldsymbol{i}}^{n+1}$ as the distribution function after remapping and $f_{\boldsymbol{i}}^{n}$ as the distribution function before remapping. It is easy to formulate $f$ by

$$f_{\boldsymbol{i}}^{n+1} = \sum_k \frac{q_k}{h_x h_v} u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) u\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right) \tag{4.13}$$

and

$$f_{\boldsymbol{i}}^{n} = \sum_k \frac{q_k}{h_x h_v} H\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) H\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right), \tag{4.14}$$

where $H$ is the hat function

$$H(y) = \begin{cases} 1 & \text{if } 0 \le |y| \le \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

The flux $\boldsymbol{F}$ can be calculated by solving the following well-defined Poisson system:

$$\begin{aligned} \nabla \cdot \boldsymbol{F} &= f_{\boldsymbol{i}}^{n} - f_{\boldsymbol{i}}^{n+1} \\ &= \sum_k \frac{q_k}{h_x h_v} H\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) H\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right) \\ &\quad - \sum_k \frac{q_k}{h_x h_v} u\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) u\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right) \end{aligned} \tag{4.16}$$

with boundary conditions $\boldsymbol{F}_{x=0} = \boldsymbol{F}_{x=L}$ and $\nabla \cdot \boldsymbol{F}_{|v|=v_{\max}} = 0$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We define a low-order flux $\boldsymbol{F}^{\text{lo}}$ and a high-order flux $\boldsymbol{F}^{hi}$ as using a low-order interpolation function $u^{\text{lo}}$ (e.g., $M_2$) and a high-order interpolation function $u^{\text{hi}}$ (e.g., $W_4$), respectively. The flux in each direction can be calculated by

$$F^{\text{lo}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} = \frac{\varphi^{\text{lo}}_{\boldsymbol{i}+\boldsymbol{e}_d} - \varphi^{\text{lo}}_{\boldsymbol{i}}}{h_d}, \quad F^{\text{hi}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} = \frac{\varphi^{\text{hi}}_{\boldsymbol{i}+\boldsymbol{e}_d} - \varphi^{\text{hi}}_{\boldsymbol{i}}}{h_d}, \tag{4.17}$$

where $d = 0, N-1$ is the dimension in phase space. $\varphi^{\text{lo}}$ and $\varphi^{\text{hi}}$ are the potentials resulting from solving the Poisson equation using a high-order and a low-order interpolating kernel, respectively. For example,

$$\begin{aligned}\varphi^{\text{lo}}_{\boldsymbol{i}} = \Delta^{-1}_{\boldsymbol{h}} \Bigg[ &\sum_k \frac{q_k}{h_x h_v} H\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) H\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right) \\ &- \sum_k \frac{q_k}{h_x h_v} u^{\text{lo}}\left(\frac{x_{\boldsymbol{i}} - x_k}{h_x}\right) u^{\text{lo}}\left(\frac{v_{\boldsymbol{i}} - v_k}{h_v}\right)\Bigg].\end{aligned} \tag{4.18}$$

We then use the multidimensional FCT algorithms by Zalesak [45] with a minor modification. First, we do not enforce the maximum limiting. Second, in estimating the extra capacity of a cell, $Q_{\boldsymbol{i}}$ (equation (4.23)), we use use global minimum, 0, instead of local minimum. The algorithm proceeds as follows:

1. Compute $F^{\text{lo}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}}$ and $F^{\text{hi}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}}$ by a low-order and a high-order interpolation scheme, respectively.

2. Define the antidiffusive flux as

$$A_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} = F^{\text{lo}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} - F^{\text{hi}}_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}}. \tag{4.19}$$

3. Limit the antidiffusive flux by

$$A^C_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} = C_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} A_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}}, \tag{4.20}$$

where

$$C_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} = \begin{cases} \min(1, R_{\boldsymbol{i}}), & A_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} \geq 0, \\ \min(1, R_{\boldsymbol{i}+\boldsymbol{e}_d}), & A_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} < 0, \end{cases} \tag{4.21}$$

and

$$R_{\boldsymbol{i}} = \begin{cases} \min(1, \frac{Q_{\boldsymbol{i}}}{P_{\boldsymbol{i}}}), & P_{\boldsymbol{i}} > 0, \\ 0, & P_{\boldsymbol{i}} = 0. \end{cases} \tag{4.22}$$

$Q_{\boldsymbol{i}}$ is the extra capacity of cell $\boldsymbol{i}$

$$Q_{\boldsymbol{i}} = \max(0, f^{\text{lo}}_{\boldsymbol{i}}), \tag{4.23}$$

and $P_{\boldsymbol{i}}$ is the sum of all antidiffusive fluxes away from the grid cell $\boldsymbol{i}$ defined as

$$P_{\boldsymbol{i}} = \sum_{d=0}^{1} \left( \max(0, A_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}}) - \min(0, A_{\boldsymbol{i}-\frac{\boldsymbol{e}_d}{2}}) \right). \tag{4.24}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

4. Compute the limited value as

$$f_{\boldsymbol{i}}^{\text{hi}} = f_{\boldsymbol{i}}^{\text{lo}} - \sum_{d=0}^{1} \left( A^C_{\boldsymbol{i}+\frac{\boldsymbol{e}_d}{2}} - A^C_{\boldsymbol{i}-\frac{\boldsymbol{e}_d}{2}} \right). \tag{4.25}$$

This FCT algorithm enforces positivity of the distribution function using a globally determined function $\boldsymbol{F}$ in $\mathbb{R}^{2N}$. The Vlasov equation is in phase space, where $2N = 2, 4, 6$. For $N = 1$, this poses no particular difficulty. With $N = 2, 3$, this approach is unattractive since it requires us to solve the Poisson equation in four dimensions and six dimensions. However, this algorithm will be very useful to enforce positivity in high-order remapping for other particle methods, such as vortex methods and SPH. Meanwhile, we develop a second approach which uses only local information.

**4.1.2. Local positivity preserving algorithm.** Our local algorithm is based on the mass redistribution idea of Chern and Colella [10], first applied to enforce positivity preservation by Hilditch and Colella [23]. In the algorithm, we redistribute the undershoot of cell $\boldsymbol{i}$

$$\delta f_{\boldsymbol{i}} = \min(0, f_{\boldsymbol{i}}^n) \tag{4.26}$$

to its neighboring cells $\boldsymbol{i} + \boldsymbol{\ell}$ in proportion to their capacity $\xi$

$$\xi_{\boldsymbol{i}+\boldsymbol{\ell}} = \max(0, f_{\boldsymbol{i}+\boldsymbol{\ell}}^n). \tag{4.27}$$

The distribution function is conserved, which fixes the constant of proportionality

$$f_{\boldsymbol{i}+\boldsymbol{\ell}}^{n+1} = f_{\boldsymbol{i}+\boldsymbol{\ell}}^n + \frac{\xi_{\boldsymbol{i}+\boldsymbol{\ell}}}{\sum_{\boldsymbol{k}\neq 0}^{\text{neighbors}} \xi_{\boldsymbol{i}+\boldsymbol{k}}} \delta f_{\boldsymbol{i}} \tag{4.28}$$

for $\boldsymbol{\ell} \neq 0$ such that cell $\boldsymbol{i} + \boldsymbol{\ell}$ is a neighbor of cell $\boldsymbol{i}$. Superscript $n$ and $n + 1$ denote the interpolated value before and after redistribution, respectively.

The drawback of this approach is that positivity is not guaranteed in a single pass. One might have to apply the method iteratively. In practice, however, we find a few iterations to be sufficient.

**4.2. Remapping on a hierarchy of grids.** Remapping on a uniform grid tends to create a lot of particles with very small strength on the tail of the distribution function. The situation becomes worse with each successive remapping step. We solve this problem using mesh refinement, where particle charges are interpolated on a hierarchy of block-structured grids. The main body of the distribution function is covered by finer grids, while the tail of the function is covered by coarser grids. Care must be taken in designing the remapping scheme on coarse-fine interfaces to make sure that the total charge is conserved and the overall accuracy of the field is maintained.

Before explaining the algorithm, we introduce the definition of a composite grid. We define a hierarchy of cell-centered grids $\Omega_\ell$, where $0 \leq \ell \leq \ell_{\max}$. $\Omega_{\ell=0}$ is the coarsest grid that covers the whole problem domain. The finer grids $\Omega_{\ell>0}$ are constructed as a union of cell-centered rectangles. The mesh spacing of each level is $\boldsymbol{h}_\ell = \boldsymbol{h}_{\ell-1}/\boldsymbol{r}_{\ell-1}$, where $\boldsymbol{r}_{\ell-1}$ is the refinement ratio of level $\ell - 1$. The composite grid consists of valid grids at all levels, where a valid grid is defined as a region not overlain by a finer grid. That is,

$$\Omega_c = \cup_{\ell=0}^{\ell_{\max}} \left( \Omega_\ell \setminus P_{\ell+1}^{\ell}(\Omega_{\ell+1}) \right), \tag{4.29}$$

where $P_{\ell+1}^{\ell}$ is the operator projecting from level $\ell + 1$ to level $\ell$.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

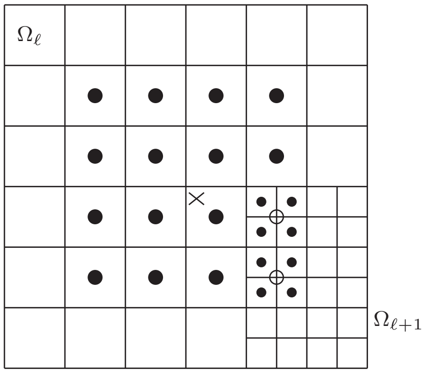


FIG. 2. *Particle is at the coarser level side. Cross signs denote the particle locations. The valid and the invalid deposited cells are denoted by filled circles and open circles, respectively. The refinement ratio is $\boldsymbol{r}_0 = (2, 2)$ in the plot.*

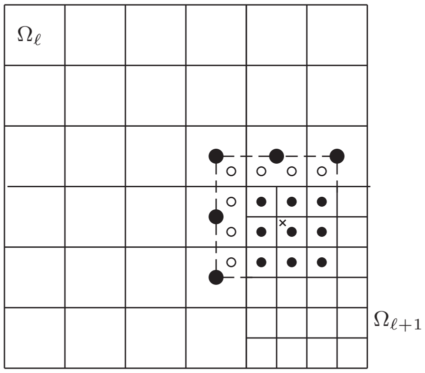


FIG. 3. *Particle is at the finer level side. Cross signs denote the particle locations. The valid and the invalid deposited cells are denoted by filled circles and open circles, respectively. The refinement ratio is $\boldsymbol{r}_0 = (2, 2)$ in the plot.*

At the beginning, a set of particles is loaded on the composite grid. If a particle is on a valid cell which is far enough away from a coarse-fine interface, it is deposited on the grid as in (4.11), where the interpolation stencil size is $\boldsymbol{h} = \boldsymbol{h}_\ell$. If a particle is near a coarse-fine interface and the interpolation stencil intersects the interface, special care must be taken. First, we deposit the charge on the surrounding cells as in the usual case. There are two cases depending on where the particle is located. If the particle is on the coarser side and the deposited cell is not a valid cell (open circles in Figure 2), the deposited charge will be transferred to the finer level cell through interpolation. On the other hand, if the particle is on the finer side and the deposited cell is outside the finer grid $\Omega_{\ell+1}$ (open circles in Figure 3), the charge will be transferred to the coarser level through projection. We lose one order of accuracy in interpolating the coarser level charge into the finer level. However, since the coarse-fine interface is in co-dimension one, the expected accuracy in the field, e.g., second order, will be observed in $L_\infty$ [30], [28, p. 19], [25, p. 17]. Our current implementation doesn't have time-dependent adaptivity. This feature can be incorporated by selecting some refinement criterion, for example, each particle on the grid carries approximately the same weight.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

**5. Collision model.** The phase-space grids constructed at remapping time provide an opportunity to introduce a Fokker–Planck collisional term to the Vlasov–Poisson equation. The coupled system describes the kinetic plasma system involving weak collisions.

We adopt the simplified Fokker–Planck collision model suggested by Rathmann and Denavit [36]. That is,

$$\left(\frac{\partial f}{\partial t}\right)_c = \frac{\partial}{\partial v}\left[\nu v f + D\frac{\partial}{\partial v}(\nu f)\right], \tag{5.1}$$

where

$$\nu = C/(2\langle v^2\rangle + v^2)^{3/2} \tag{5.2}$$

is the velocity-dependent collision frequency parameter, and

$$D = \langle v^2\nu\rangle/\langle\nu\rangle. \tag{5.3}$$

The brackets $\langle\dots\rangle$ denote a weighted average quantity, where $\langle g\rangle = \frac{\int gf dv}{\int f dv}$. The first term on the right-hand side of (5.1) represents a friction force due to collisions, while the second term represents diffusion. The diffusion parameter $D$ is chosen to conserve total energy [36].

The above system is a nonlinear convection-diffusion equation. We solve it on phase-space grids at remapping time with a second-order accurate finite-volume method. As a diffusion-dominated problem, the convection term can be represented by a central difference discretization. It is crucial to extend the domain in velocity space such that homogeneous Neumann boundary conditions can be applied. In implementing the finite-volume method for the equation on a hierarchy of grids, we discretize the system on the composite grid (4.29) where both Dirichlet and Neumann boundary conditions are enforced at coarse-fine interfaces [29].

As a time integrator for (5.1), we use the $L_0$-stable scheme of Twizell, Gumel, and Arigu [41]. In addition, as an implicit integrator, the time step will not be restricted by the CFL condition. This feature is very important since otherwise the remapping time will be restricted by the CFL condition. The collisional part is coupled to the Vlasov–Poisson system with Strang's splitting, a second-order operator splitting method [40].

We summarize the procedure of the PIC method with remapping for the Vlasov–Poisson equation along with the Fokker–Planck collision term in a single step below. The flow chart is displayed in Figure 4.

Step 1. Solve the Vlasov–Poisson equation with the PIC method from $t^N$ to $t^{N+1/2}$ followed by a positive adaptive mesh refinement (AMR) remapping at time

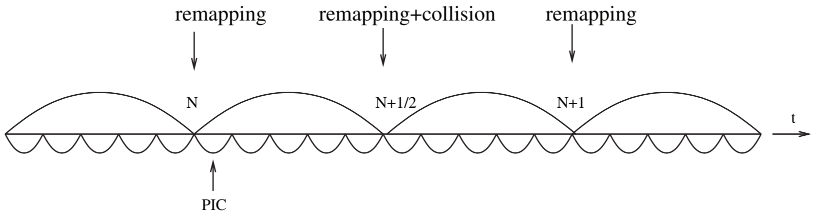


FIG. 4. *The flow chart of the algorithm.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$t^{N+1/2}$. The evolution of the distribution function from $t^N$ to $t^{N+1/2}$ involves several PIC integration steps (e.g., $K = 5$; see section 6).

Step 2. Solve the Fokker–Planck equation with a second-order finite-volume method from $t^N$ to $t^{N+1}$ on the phase-space remapping grids at time $t^{N+1/2}$.

Step 3. Solve the Vlasov–Poisson equation with the PIC method from $t^{N+1/2}$ to $t^{N+1}$ following by a positive AMR remapping at time $t^{N+1}$.

**6. Numerical results.** In this section, we first demonstrate the method for the Vlasov–Poisson system on one-dimensional plasma problems, including linear Landau damping and the two-stream instability. We then demonstrate the method for the Vlasov–Poisson–Fokker–Plank system on the two-stream instability problem. In implementing remapping on a hierarchy of locally refined grids, we refine the grid in velocity space only, since small structures occur in velocity space, but not in physical space in the tests below.

The following notation is used to denote the numerical parameters in this chapter. The ratio of the initial particle mesh spacing in physical space and the solver mesh spacing is $r_h = \frac{h_x}{\Delta x}$. Remapping is applied periodically with parameter $K = \frac{\Delta t}{dt}$, where $\Delta t$ and $dt$ are remapping step size and PIC integration step size, respectively. As suggested by the error analysis, remapping is expected to be applied when the exponential term exceeds some threshold. In the current research, we set the remapping frequency to be a constant number $K = 5$. That means we apply remapping at every five PIC steps. We can also choose other remapping frequency numbers. However, our numerical study suggests that $K = 5$ or $K = 10$ is an optimal number for the current problems.

We use Richardson extrapolation for the error estimates. If $\tilde{E}^h$ is the electric field computed with the initial phase-space discretization $(h_x, h_v)$ and PIC integration step $dt$, whereas $\tilde{E}^{2h}$ is the field computed with $(2h_x, 2h_v)$ and $2dt$, the relative solution error is defined as

$$e^h = |\tilde{E}^h - \tilde{E}^{2h}|. \tag{6.1}$$

$q$ is the order of the method and is calculated by

$$q = \log_2\left(\frac{||e^{2h}||}{||e^h||}\right). \tag{6.2}$$

### 6.1. Vlasov–Poisson system.

**6.1.1. Linear Landau damping.** The initial distribution for linear Landau damping is

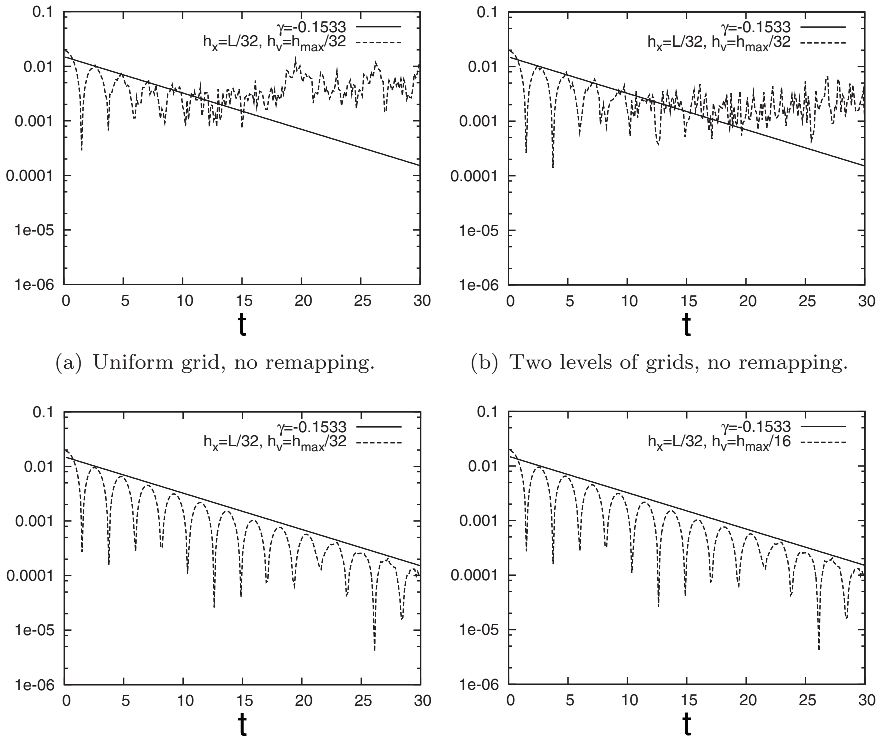
$$f(x, v, t = 0) = \frac{1}{\sqrt{2\pi}} \exp(-v^2/2)(1 + \alpha\cos(kx)) \\ (x, v) \in [0, L = 2\pi/k] \times [-v_{\max}, v_{\max}], \tag{6.3}$$

where $\alpha = 0.01$ is the intensity of the perturbation, $k = 0.5$ is the first mode of the electric field, and $v_{\max} = 10$. In initializing particles, we set a threshold for the particle charge intensity. We ignore the particles which have weight less than $1.0 \times 10^{-9}$.

In the test, we are interested in the evolution of the electric field. This problem has been studied extensively [9, 42, 35, 18, 5]. According to Landau's theory, the electric field is expected to decrease exponentially with damping rate $\gamma = 0.153$ and frequency $\omega = 1.416$. We initialize the problem first on a uniform grid, $h_x = L/32$, $h_v = v_{\max}/32$,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Uniform grid, no remapping.

(b) Two levels of grids, no remapping.

(c) Uniform grid, with remapping.

(d) Two levels of grids, with remapping.

| case | total number of particles |
|---|---|
| (a) | 1600 |
| (b) | 2624 |
| (c) | 2048 |
| (d) | 1539 |

Fig. 5. *Comparison of the amplitude of the electric field (or $L_\infty$) in logarithm scale for linear Landau damping solved by the PIC method with different remapping grids. Scales $(h_x, h_v)$ denote the particle grid mesh spacing at the base level. In the upper two plots, we start by putting particles on a uniform grid and two levels of grids, respectively, and we don't apply remapping during the whole evolution. In the bottom two plots, we start by putting particles on a uniform grid and two levels of grids, respectively, but apply periodic remapping. The remapping is done on a uniform grid and two levels of grids, respectively, as in the starting layout. The table shows the maximum cardinality of particles during the simulation in the different cases above.* (a) *The standard PIC method fails to solve the Landau damping problem,* (b) *increasing the total number of particles in the standard PIC method improves the results,* (c) *the PIC method with remapping succeeds in tracking the exponential decay of the electric field, and* (d) *mesh refinement reduces the total number of particles without destroying the simulation.*

without remapping. The PIC time step is $dt = 1/8$. We set the ratio of the particle mesh spacing to the solver mesh spacing to be $r_h = 1/2$. Figure 5(a) shows that PIC without remapping fails to track the exponential damping. We then solve the same problem after increasing the total number of particles through initializing on two levels of grids, with another finer level covering subdomain $v \in (-5, 5)$. Figure 5(b) shows that increasing the total number of particles in the standard PIC method improves the result. The exponential decay is successfully captured during the early time of the simulation. However, it fails to track the damping rate after $t = 12$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

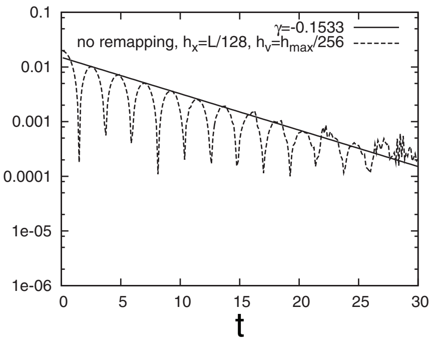


Fig. 6. *The evolution of the amplitude of the electric field in logarithm scale for the linear Landau damping problem solved by the standard PIC method without remapping. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level. We have two levels of grids in the plot. The total number of particles is* 60,416. *The CPU running time is* 141.6 *seconds. Compared to Figure* 5(d)*, we see the standard PIC method is much more expensive.*

We then solve the problem with uniform initialization as in Figure 5(a), but with remapping at a rate $K = 5$. Figure 5(c) shows that PIC with remapping gives a good approximation (Figure 5(c)) compared to the exact damping rate. We show only the results up to $t = 30$ since the simulation will be influenced by the recurrence effect at later times. Figure 5(d) shows the decay of the electric energy solved by initializing and remapping particles on two levels of grids. Compared to Figure 5(c), we start from a coarser mesh in velocity space, but we add a finer level in velocity space on subdomain $v \in [-5, 5]$. The exponential decay is successfully captured with fewer particles than in Figure 5(c). The table shows the maximum number of particles used during the simulation in each case above.

To further demonstrate the developed method, we also show the evolution of the electric field using the standard PIC method with a large number of particles in Figure 6. We hope that the standard PIC can give us similar decay plot as in the remapped PIC method (Figure 5(d)) by increasing the total number of particles. The total number of particles and the CPU running time in Figure 6 show that the standard PIC is much more expensive than the remapped PIC method.

Finally, we compare the convergence rates of the electric field errors without and with remapping. The particles in the simulation are initialized on two levels of grids with the finer grid covering velocity space subdomain $v \in (-5, 5)$. The particle and the solver mesh spacing ratio is set to be $r_h = 1/2$ as before. Figure 7 shows the $L_\infty$ norm of the errors at the case without remapping in three different resolutions with the corresponding convergence rates. Comparing Figure 8 with Figure 7, we see that remapping significantly reduces the $L_\infty$ error of the electric field. Second-order convergence rates are obtained after applying remapping.

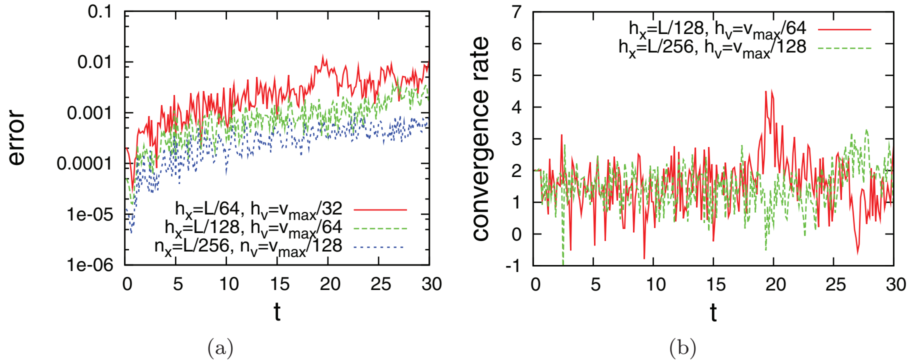
**6.1.2. The two-stream instability.** We consider the two-stream instability with the initial distribution

$$f(0, x, v) = \frac{1}{\sqrt{2\pi}} v^2 \exp(-v^2/2)(1 + \alpha \cos(kx)), \quad (x, v) \in [0, L = 2\pi/k] \times [-v_{\max}, v_{\max}], \tag{6.4}$$

where $\alpha = 0.01$, $k = 0.5$, and $v_{\max} = 10$. In initializing particles, we ignore the particles which have strength less than $1.0 \times 10^{-9}$ as before.

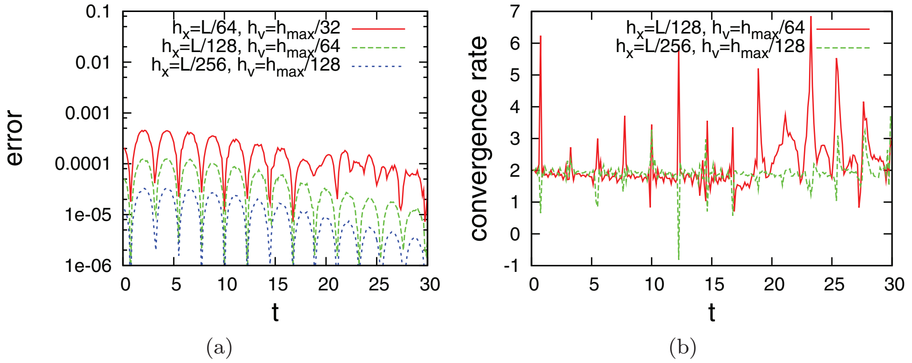
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) (b)

FIG. 7. *Error and convergence rate plots for linear Landau damping problem without remapping. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level.* (a) *The* $L_\infty$ *norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on* (a).



(a) (b)

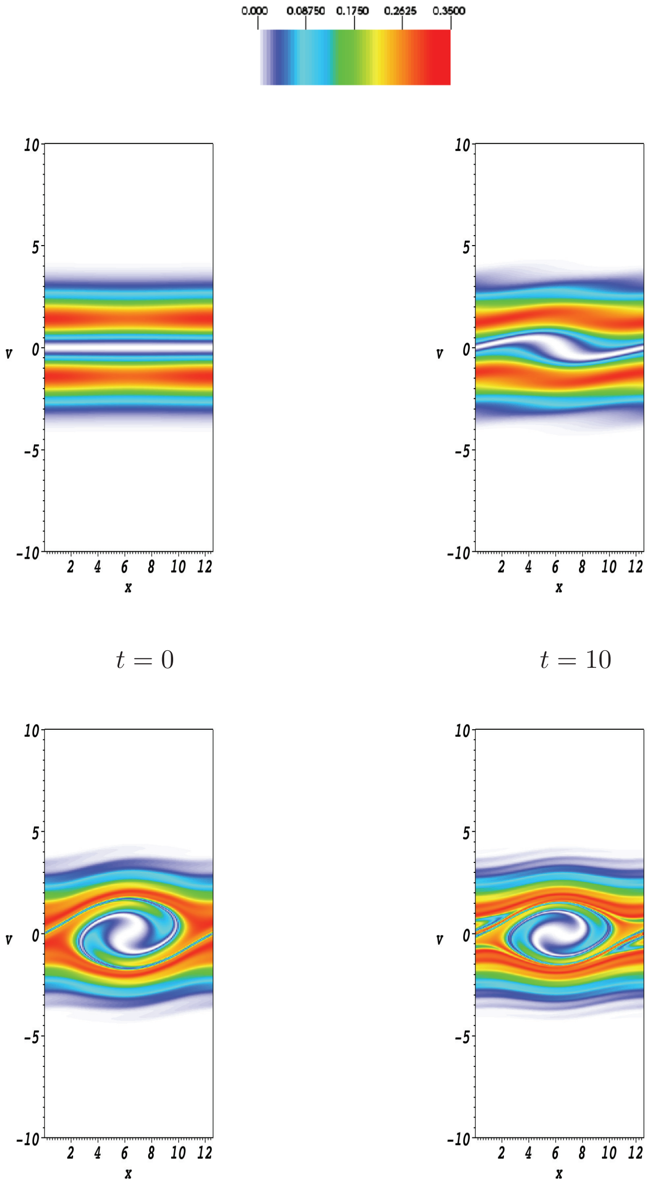
FIG. 8. *Error and convergence rate plots for linear Landau damping problem with remapping. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level.* (a) *The* $L_\infty$ *norm of the electric field errors on three different resolutions. Compared to the case without remapping in Figure* 7, *we see the errors are significantly reduced.* (b) *The convergence rates for the errors on* (a).

Figure 9 shows the evolution of the distribution function in phase space. The distribution function plots are obtained by reproducing the particle-based distribution function through a second-order interpolation on phase-space grids. In the simulation, the distribution function is initialized and remapped on a two-levels grid starting at $h_x = L/128$, $h_v = v_{\max}/256$. We refine velocity space on region $v \in [-5, 5]$ by a factor of 2. We choose $dt = 1/32$ for the PIC integration step size and $r_h = 1/2$ for the particle and solver mesh spacing ratio. At $t = 10$, a vortex forms. A hole structure associated with particle trapping is seen around $t = 20$. After $t = 20$, the vortex rotates periodically with $T \approx 17$. Comparing it with the other calculations [9, 42, 35, 18, 5], we see a very good agreement.

We investigate the method in two series of tests. As the first one, we compare the convergence rates of the electric field errors without and with remapping. The ratio of the particle and the solver mesh spacing is set to be $r_h = 1/2$. Figure 10 shows the $L_\infty$ norm of the errors in the case without remapping with three different resolutions. The corresponding convergence rates are shown at the bottom. Second-order convergence rates are lost early in the simulation, around $t = 20$. Compared to the results with

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



$t = 0$ $t = 10$

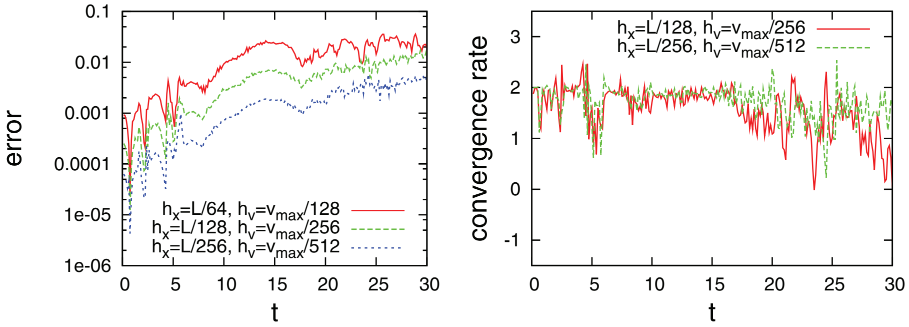
$t = 20$ $t = 30$

FIG. 9. *The distribution function $f(x, v, t)$ for the two-stream instability at times $t = 0, 10, 20, 30$, respectively. We initialize and remap the distribution function on two levels of grids, with base grid at $h_x = L/128$, $h_v = v_{\max}/256$. The grid is refined by a factor of 2 in $v$ space on subdomain $v \in [-5, 5]$. We see filamentation around $t = 30$.*

remapping in Figure 11, we see that remapping extends the second-order convergence rates to longer times, until $t = 28$. The loss of accuracy after $t = 28$ is caused by the development of filaments. In the second series of tests, we set $r_h = 1$. According to Cottet's convergence formula [13], the simulation will result only in a first-order method. However, Figures 12 and 13 show that the PIC method can be second order

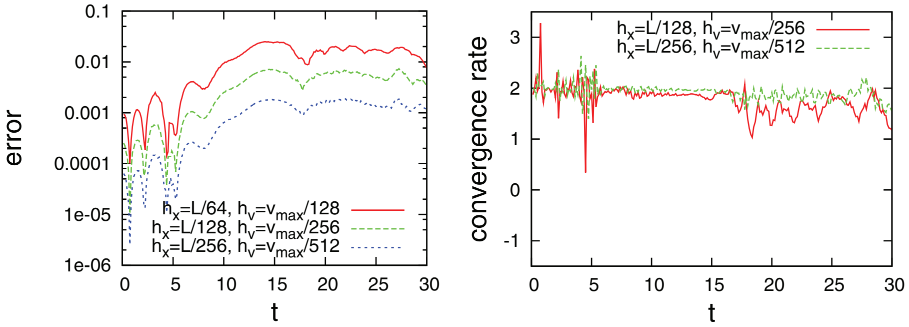
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) (b)

FIG. 10. *Error and convergence rate plots for the two-stream instability without remapping. We set $r_h = 1/2$. Scales $(h_x, h_v)$ denote the particle grid mesh spacing at the base level.* (a) *The $L_\infty$ norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on plot* (a). *Second-order convergence rates are lost around $t = 20$.*



(a) (b)

FIG. 11. *Error and convergence rate plots for the two-stream instability with remapping. We set $r_h = 1/2$. Scales $(h_x, h_v)$ denote the particle grid mesh spacing at the base level.* (a) *The $L_\infty$ norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on plot* (a). *Second-order convergence rates are observed until $t = 28$. The lost of accuracy after $t = 28$ is due to filamentation.*

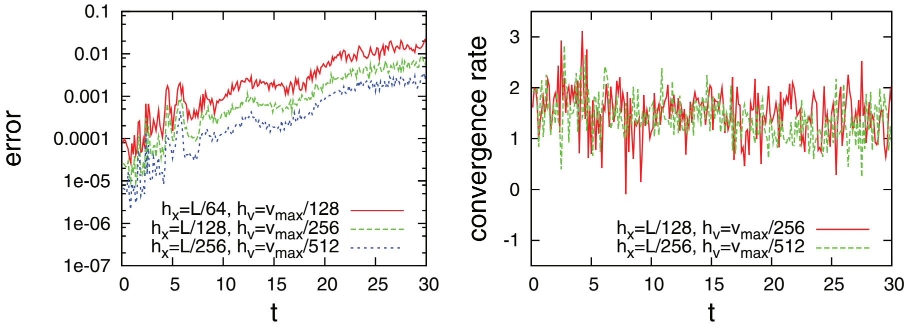
even with $r_h = 1$. This further demonstrate our error formula (3.12), which says the consistency error is second order as long as $r_h \leq 1$.

We also compare the distribution function at the same instant time $t = 20$ by both methods in Figure 14. For visualization purposes, in the case of the PIC method without remapping, we interpolate the particle-based distribution on phase-space grids. We see that the standard PIC method results in a very noisy solution in Figure 14(a). In addition, the maximum of the approximated distribution function has a large error compared with the analytic value, $f_{\max} = 0.3$. Figure 14(b) shows the distribution function solved by the PIC with remapping. Compared to the case without remapping, remapping obviously controls numerical noise and reduces the maximum error. We preserve the maximum of the distribution function by applying the mass redistribution algorithm as in positivity preservation.

Finally, we compare the evolution of the total number of particles in three cases. In the first case, we initialize and remap the problem on two levels of grids, with

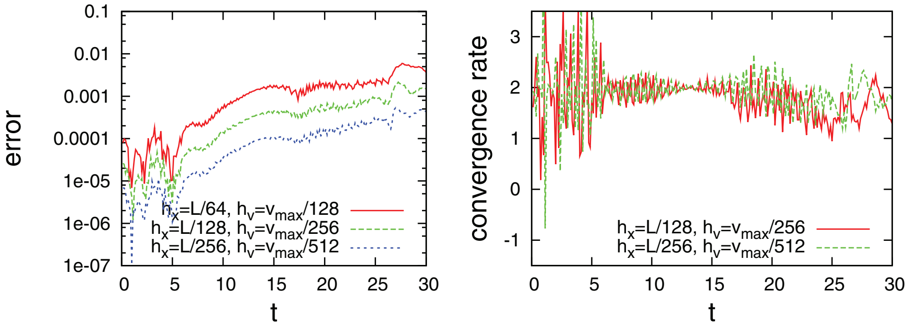
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) (b)

FIG. 12. *Error and convergence rate plots for the two-stream instability without remapping. We set* $r_h = 1$*. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level.* (a) *The* $L_\infty$ *norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on plot* (a).



(a) (b)

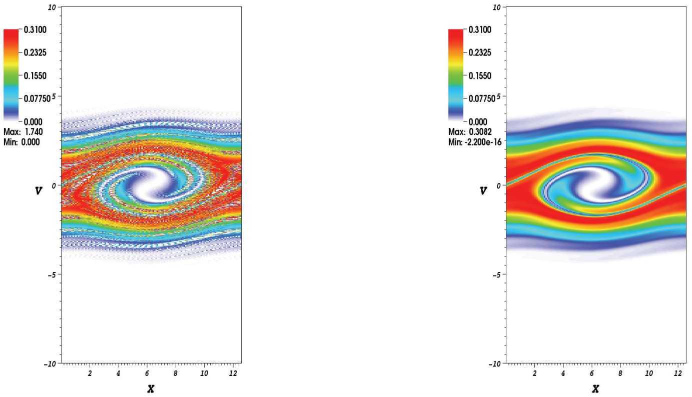
FIG. 13. *Error and convergence rate plots for the two-stream instability with remapping. We set* $r_h = 1$*. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level.* (a) *The* $L_\infty$ *norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on plot* (a).

refinement in velocity space by a factor of 2 on subdomain $v \in (-5, 5)$. The base grid mesh spacing is $h_x = L/64$, $h_v = v_{\max}/128$. In the second case, we initialize and remap the problem on a uniform grid with mesh spacing $h_x = L/64$, $h_v = v_{\max}/256$. In the last case, we initialize the problem on two levels of grids as in the first case, but with grid size doubled in velocity space in all levels. We do not apply remapping in this case. Figure 15 shows that remapping with mesh refinement reduces the total number of particles dramatically without losing accuracy. In the remapping cases, the total number of particles increases linearly during the early state of the evolution. The number is saturated as the evolution becomes stable since we ignore the particles when their strength is smaller than machine accuracy, e.g., $10^{-16}$.

**6.2. Vlasov–Poisson–Fokker–Planck system.** We investigate the method for solving Vlasov–Poisson-Fokker–Planck on the two-stream instability problem. The initial distribution function is defined in (6.4) as in the collisionless case. We solve the problem on two levels of grids starting at $h_x = L/128$, $h_v = v_{\max}/256$. The velocity

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

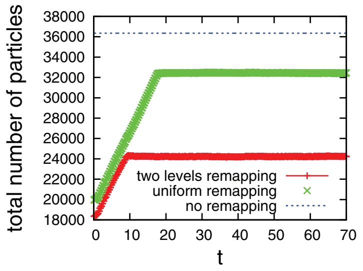
Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Without remapping at time $t = 20$, $\tilde{f}_{\max} = 1.591$ (vs. $f_{\max} = 0.3$), $\tilde{f}_{\min} = 0$.

(b) With remapping at time $t = 20$, $\tilde{f}_{\max} = 0.306$ (vs. $f_{\max} = 0.3$), $\tilde{f}_{\min} = 0$.

FIG. 14. *Comparision of the distribution function at the same instant of time $t = 20$* (a) *without and* (b) *with remapping for the two-stream instability. The grid-based distribution function is obtained by reproducing the particle-based distribution function through a second-order interpolation. We initialize the distribution function on two levels of grids, with base grid at $h_x = L/128$, $h_v = v_{\max}/256$. The grid is refined by a factor of 2 in $v$ space on subdomain $v \in [-5, 5]$. The standard PIC method results in a very noisy solution with large maximum error. If we apply remapping to the PIC method, both numerical noise and maximum error are significantly reduced. The analytic maximum of the distribution function in the two-stream instability is about $f_{\max} = 0.3$.*



| case | error |
|---|---|
| 1: two-levels remapping | 0.02292 |
| 2: uniform remapping | 0.02292 |
| 3: no remapping | 0.03546 |

FIG. 15. *Comparison of the total number of particles in three different remapping grids for the two-stream instability. Plus line (case 1): initialize and remap on two levels of grids. The base grid mesh spacing is $h_x = L/64$, $h_v = v_{\max}/128$. We refine the mesh in $v$ space by a factor of 2 on subdomain $v \in [-5, 5]$. Cross line (case 2): initialize and remap on a uniform grid with mesh spacing $h_x = L/64$, $h_v = v_{\max}/256$. The mesh spacing in case 2 is the same as the finer level mesh spacing in case 1. Dash line (case 3): initialize on two levels of grids, without remapping. The base grid mesh spacing is $h_x = L/64$, $h_v = v_{\max}/256$. We refine the mesh in $v$ space by a factor of 2 on subdomain $v \in [-5, 5]$ as in case 1. Compared to case 1, case 3 doubles the total number of initial particles. The table shows the $L_\infty$ norm error of the electric field in the above simulations at $t = 70$. We see that refinement on velocity space reduces the total number of particles dramatically without losing accuracy.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

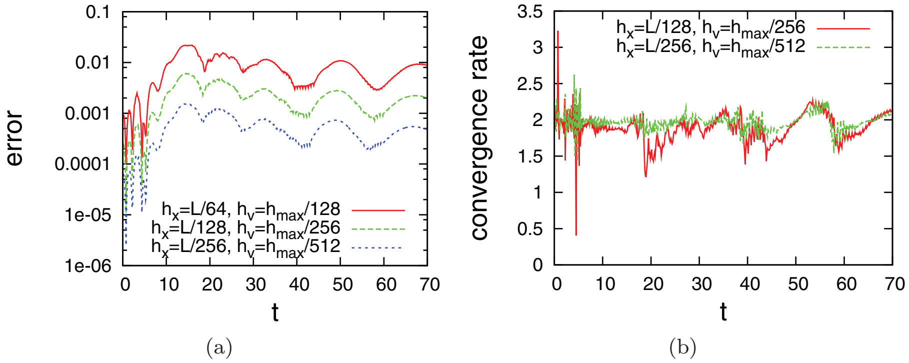


FIG. 16. *Error and convergence plots for the two-stream instability including collision on Figure 9. Scales* $(h_x, h_v)$ *denote the particle grid mesh spacing at the base level.* (a) *The* $L_\infty$ *norm of the electric field errors on three different resolutions.* (b) *The convergence rates for the errors on plot* (a). *The second-order convergence rates are maintained.*

space is refined by a factor of 2 on region $v \in [-5, 5]$. We choose $dt = 1/32$ for the PIC integration step size. The collision frequency in (5.2) is $C = 0.01$. Figure 16 shows the errors and the corresponding convergence rates for the simulation for the coupled system. Second-order convergence rates are observed over the whole simulation. Figure 17 shows that filaments are smoothed by the dissipative term which results in a well-resolved system.

## 7. Conclusion and future work.

We have presented an accurate and efficient particle method for solving kinetic plasma problems. The method is based on the PIC method with periodic reinitialization of the particle distribution through remapping on phase-space grids. The method achieves second-order accuracy through the use of a high-order remapping stencil. In addition, we provide two positivity preservation algorithms for high-order remapping. The phase-space grids used for remapping also offer an opportunity to include a collision term and a grid-based solver. We have demonstrated the method on a set of one-dimensional plasma problems, in both collisionless and weakly collisional limits. Second-order convergence rates on the electric field have been obtained for the two-stream instability under weakly collisional conditions.

The method can be naturally generalized to high dimensions. However, a parallel implementation of the method is necessary for performing higher-dimensional simulation. The current implementation is parallelized by using the block-structured grid and domain decomposition on physical space. It is not a linear scaled algorithm because of decomposition in physical space only. Domain decomposition based on phase space will be a great option to achieve scalability of the algorithm, which will be an extension of the current work.

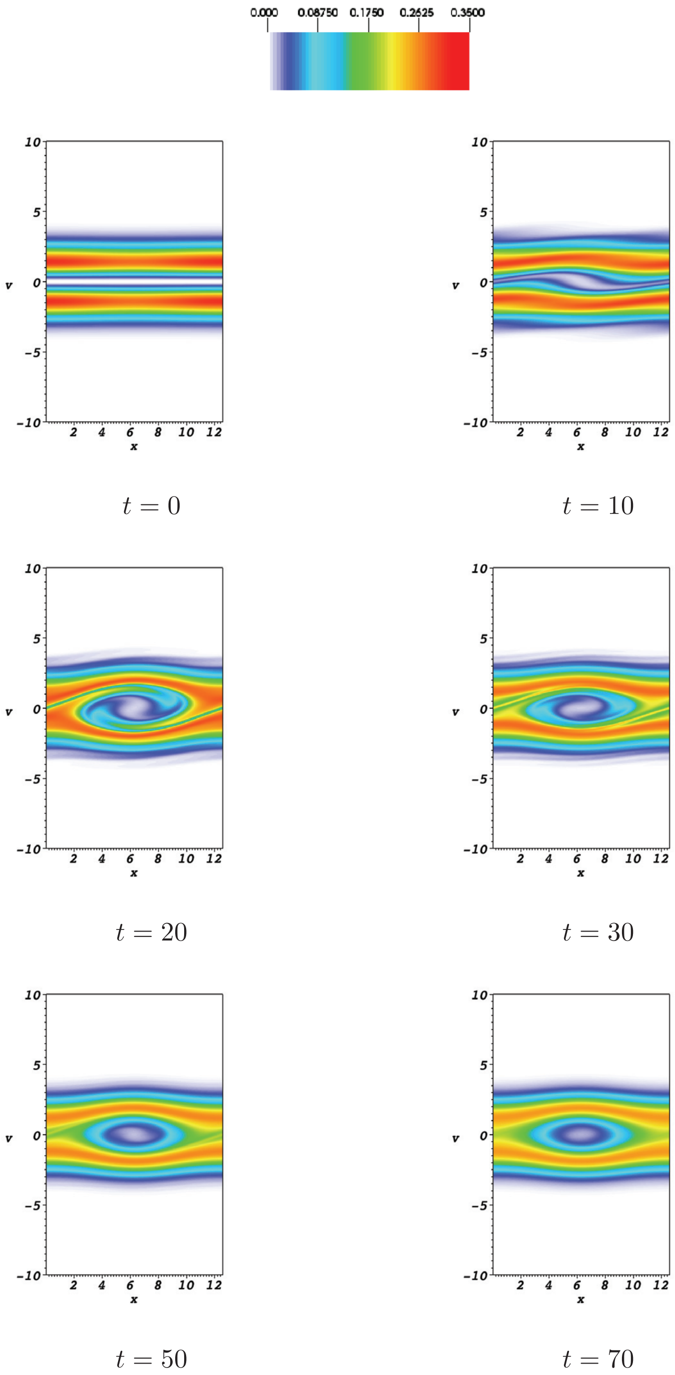
## Appendix A. The accuracy of particle methods for the one-dimensional Vlasov–Poisson system.

We start the analysis by reproducing the electron charge density function in physical space through a particle representation:

$$\tilde{\rho_e}(x,t) = \sum_k q_k \delta_{\Delta x}(x - \tilde{X}_k(t)), \quad 0 \le t \le T. \tag{A.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



$t = 0$ $t = 10$

$t = 20$ $t = 30$

$t = 50$ $t = 70$

FIG. 17. *The distribution function $f(x, v, t)$ for the two-stream instability including collision at times $t = 0, 10, 20, 30, 50, 70$, respectively. Compare with the collisionless case 9. Filaments are smoothed by the collision term.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The error in charge density is

$$
\begin{aligned}
e_t(x,t) &= |\rho(x,t) - (1 - \tilde{\rho}_e(x,t))| \\
&= |\rho_e(x,t) - \sum_k q_k \delta_{\Delta x}(x - \tilde{X}_k(t))| \\
&\le \underbrace{\left| \rho_e(x,t) - \sum_k q_k \delta_{\Delta x}(x - X_k(t)) \right|}_{e_c(x,t)} \\
&\quad + \underbrace{\left| \sum_k q_k \delta_{\Delta x}(x - X_i(t)) - \sum_k q_k \delta_{\Delta x}(x - \tilde{X}_k(t)) \right|}_{e_s(x,t)},
\end{aligned}
\tag{A.2}
$$

where $\rho_e(x,t)$ is the exact electron charge density. We use $X_k$ and $\tilde{X}_k$ to denote the exact and the particle approximated trajectories, respectively. The term $e_c$ is the consistency error caused by representing the continuous system with discretized particles, while the term $e_s$ is the stability error introduced by approximating the system with the computed trajectories $(\tilde{X}_k(t), \tilde{V}_k(t))$ instead of the exact ones $(X_k(t), V_k(t))$.

**A.1. Consistency error of the charge density.** The consistency error measures the error associated with representing a smooth charge density by arbitrarily distributed particles. Since the error in replacing arbitrarily distributed particles with a set of uniformly distributed particles is of the same order as the consistency error (see (4.8), we can start our analysis by measuring the error associated with representing a smooth charge density on a rectangular grid. In addition, since there is no error in representing ion charge density, the charge density error is reduced to the particle representation of electron charge density. That is,

$$
\begin{aligned}
e_c(x,t) &= \left| \rho_e(x,t) - \sum_{\boldsymbol{i}} q_{\boldsymbol{i}} \delta_{\Delta x}(x - x_{\boldsymbol{i}}) \right| \\
&\le \underbrace{\left| \rho_e(x,t) - \int_{\mathbb{R}} \rho_e(y,t) \delta_{\Delta x}(x - y) dy \right|}_{e_m(x,t)} \\
&\quad + \underbrace{\left| \int_{\mathbb{R}} \rho_e(y,t) \delta_{\Delta x}(x - y) dy - \sum_{\boldsymbol{i}} q_{\boldsymbol{i}} \delta_{\Delta x}(x - x_{\boldsymbol{i}}) \right|}_{e_d(x,t)},
\end{aligned}
\tag{A.3}
$$

where $e_m(x,t)$ is the moment error and $e_d(x,t)$ is the discretization error. Particles are located at the cell center of the rectangular grid. This replacement enables the analysis to be based on the well-developed interpolation theorem for data on a grid.

Moment error $e_m$ is a consequence of approximating the exact delta function $\delta$ with a discrete delta function $\delta_{\Delta x}$. We estimate it by using a Taylor expansion. That

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

is,

$$
\begin{aligned}
e_m(x,t) &= \left|\rho_e(x,t) - \int_{\mathbb{R}} \rho_e(y,t)\delta_{\Delta x}(x-y)dy\right| \\
&= \left|\rho_e(x,t) - \int_{\mathbb{R}} \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{\ell!}\frac{\partial \rho_e(x,t)^\ell}{\partial x^\ell}(x-y)^\ell \delta_{\Delta x}(x-y)dy\right| \\
&= \left|\rho_e(x,t) - \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{\ell!}\frac{\partial \rho_e(x,t)^\ell}{\partial x^\ell}\int_{\mathbb{R}} (x-y)^\ell \delta_{\Delta x}(x-y)dy\right|.
\end{aligned}
\tag{A.4}
$$

We assume that the moment conditions satisfy

$$
\int_{\mathbb{R}} (x-y)^\ell \delta_{\Delta x}(x-y)dy = \begin{cases} 1, & \ell = 0, \\ 0, & \ell < m \leq p, \end{cases}
\tag{A.5}
$$

where $p$ is the order of the differentiability of the charge density $\rho_e(x,t)$, and $m$ is the moment with the largest contribution to $e_m$. We further assume that $\int_{\mathbb{R}} |\delta_{\Delta x}(x-y)dy|$ is bounded. Then we get

$$
e_m(x,t) \leq c_0(T)\Delta x^m,
\tag{A.6}
$$

where $c_0(T) > 0$ is independent of $\Delta x$. Here and in what follows, all constants will be positive and independent of $\Delta x$ and $h$. In the case of the PIC method, since $u_2$ (equation (3.9)) is used as the interpolation function, $m = 2$.

In general, the discretization error is assessed by numerical quadrature rules as in [1]. But in some cases, particularly when B-splines ($M_n$) are used as the interpolation kernel, the smooth interpolation formula of Schoenberg [38] gives a more accurate estimate. We follow Monaghan [34] to derive the smooth interpolation formula below. A discretization error estimate based on the smoothed interpolation formula is presented at the end.

The error associated with approximating the convolution integral by a sum can be represented as

$$
\begin{aligned}
&\sum_{p=-\infty}^{\infty} \rho_e(ph_x,t)\delta_{\Delta x}(x-ph_x) - \int_{-\infty}^{\infty} \rho_e(uh_x,t)\delta_{\Delta x}(x-uh_x)du \\
&\qquad = 2\sum_{r=1}^{\infty}\int_{-\infty}^{\infty} \rho_e(uh_x,t)\delta_{\Delta x}(x-uh_x)\cos(2\pi ru)du,
\end{aligned}
\tag{A.7}
$$

where $p \in \mathbb{Z}$, $u \in \mathbb{R}$, and $\delta_{\Delta x}$ is a discrete delta function as given in (3.4). This formula is usually known as the Poisson summation formula [16].

Now let us expand the right-hand side of the function around $uh_x = x$ and take the dominant contribution term $r = 1$. It becomes

$$
\begin{aligned}
e_{r=1} &= \left|2\int_{-\infty}^{\infty} \rho_e(uh_x,t)\delta_{\Delta x}(x-uh_x)\cos(2\pi u)du\right| \\
&= \left|\frac{2}{h_x}\sum_{a=0}^{\infty}\frac{\rho_e^a(x,t)}{a!}\int_{-\infty}^{\infty} w^a \delta_{\Delta x}(w)\cos\left(\frac{2\pi(x+w)}{h_x}\right)dw\right|,
\end{aligned}
\tag{A.8}
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $w = uh_x - x$. The integral in (A.8) has the form

$$\begin{aligned} I_a(x,\Delta x) &= \int_{-\infty}^{\infty} w^a \delta_{\Delta x}(w) \cos\left(\frac{2\pi(x+w)}{h_x}\right) dw \\ &= \cos\left(\frac{2\pi x}{h_x}\right) \int_{-\infty}^{\infty} w^a \delta_{\Delta x}(w) \cos\left(\frac{2\pi w}{h_x}\right) dw \\ &\quad + \sin\left(\frac{2\pi x}{h_x}\right) \int_{-\infty}^{\infty} w^a \delta_{\Delta x}(w) \sin\left(\frac{2\pi w}{h_x}\right) dw. \end{aligned} \tag{A.9}$$

It is easy to observe that

$$I_a(x,\Delta x) = \begin{cases} \cos\left(\frac{2\pi x}{h}\right) \frac{(-1)^{a/2}}{(2\pi^a)} \left[\frac{d^a}{dt^a} g(z)\right]_{z=1/h_x}, & a \text{ is even}, \\ \sin\left(\frac{2\pi x}{h}\right) \frac{(-1)^{(a-1)/2}}{(2\pi^a)} \left[\frac{d^a}{dt^a} g(z)\right]_{z=1/h_x}, & a \text{ is odd}, \end{cases} \tag{A.10}$$

where

$$g(z) = \int_{-\infty}^{\infty} \delta_{\Delta x}(w) \cos(2\pi z w)\, dw \tag{A.11}$$

is the Fourier cosine transformation of the discrete delta function.

With the above smooth interpolation formula, the discretization error in (A.3) is

$$\begin{aligned} e_d(x,t) &= \left| \int_{\mathbb{R}} \rho_e(y,t) \delta_{\Delta x}(x-y) dy - \sum_{\boldsymbol{i}} q_{\boldsymbol{i}} \delta_{\Delta x}(x - x_{\boldsymbol{i}}) \right| \\ &\leq c_1(T) \left[\frac{d^n}{dt^n} g(z)\right]_{z=1/h_x}, \end{aligned} \tag{A.12}$$

where $e_d$ is bounded by the $n$th-order derivative of the Fourier transformation of the discrete delta functions at points $th = \pm 1, \pm 2, \ldots$ with

$$\left[\tfrac{d^a}{dt^a} g(z)\right]_{z=1/h_x} = 0, \quad \text{when } a < n\ . \tag{A.13}$$

For example, the Fourier cosine transformation of a B-spline $(M_n)$ of order $n$ [38] has the form

$$\left[\frac{\sin(\pi z \Delta x)}{\pi z \Delta x}\right]^n. \tag{A.14}$$

Since

$$\left[\frac{d^a}{dt^a} \hat{M}_n(z)\right]_{z=1/h_x} = \begin{cases} 0, & a < n, \\ \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n, & a = n, \end{cases} \tag{A.15}$$

where $\hat{M}_n(z) = \int_{-\infty}^{\infty} M_n(w,\Delta x) \cos(2\pi z w) dw$, when $M_n$ is taken as the discrete delta function, the discretization error is bounded by

$$e_d(x,t) \leq c_2(T) \left(\Delta x^n \left(\frac{h_x}{\Delta x}\right)^n\right). \tag{A.16}$$

The ratio $\frac{h_x}{\Delta x}$ will be shown to be important for convergence. In the case of the PIC method, $n = 2$ for the B-spline of order 2.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Combining the moment error (A.6) and the discretization error (A.16), we obtain the consistency estimate using B-spline type discrete delta functions,

$$e_c(x,t) \le c_3(T)\left(\Delta x^m + \Delta x^n\left(\frac{h_x}{\Delta x}\right)^n\right). \tag{A.17}$$

It worth mentioning that B-spline type discrete delta functions only lead to a second-order method. This is because the moment order $m \le 2$ for all B-splines, although the discretization order $n$ can be increased arbitrarily. It is easy to see from (A.5) that to cancel the quadratic moment, the discrete delta function needs to have some negative region. However, all B-spline functions are nonnegative.

**A.2. Convergence.** The remaining step is to estimate $\max_k |\tilde{X}_k(t) - X_k(t)|$ to determine the convergence of the PIC method. We use the exact same approach as in proving (4.23) of [13].

By using (3.2) and the exact trajectory function we have

$$(\tilde{X}_k - X_k)(t) = \int_0^t (\tilde{V}_k - V_k)(t')dt' \tag{A.18}$$

and

(A.19)

$$\begin{aligned}(\tilde{V}_k - V_k)(t) &= -\int_0^t (\tilde{E}(\tilde{X}_k(t'),t') - E(X_k(t'),t'))dt' \\ &= -\int_0^t ((\tilde{E} - E)(\tilde{X}_k(t'),t') + (E(\tilde{X}_k(t'),t') - E(X_k(t'),t')))dt'.\end{aligned}$$

The first term $(\tilde{E} - E)(\cdot,t)$ is the electric field error through particle approximation. It is easily shown that

$$\begin{aligned}(\tilde{E} - E)(x,t) &= \left|E(x,t) - \left(\int_0^L K(x,y)dy - \sum_k q_k K_{\Delta x}(x - \tilde{X}_k(t))\right)\right| \\ &\le \underbrace{\left|E(x,t) - \left(\int_0^L K(x,y)dy - \sum_k q_k K_{\Delta x}(x - X_k(t))\right)\right|}_{e_c^E(x,t)} \\ &\quad + \underbrace{\left|\sum_k q_k K_{\Delta x}(x - X_k(t)) - \sum_k q_k K_{\Delta x}(x - \tilde{X}_k(t))\right|}_{e_s^E(x,t)},\end{aligned} \tag{A.20}$$

where $K_{\Delta x}(x,\cdot) = K(x,\cdot) * \delta_{\Delta x}$.

We can estimate the consistency error, $e_c^E$, by regarding it as the field induced by the consistency error of the charge density (equation (A.17)). Since the discrete Laplacian operator is a stable operator, by using the same analysis as in [25, 30], we get

$$e_c^E(x,t) \le c_5(T)\left(\Delta x^m + \Delta x^n\left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2\right). \tag{A.21}$$

The last term $\Delta x^2$ is the error associated with using a second-order Poisson solver.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The bound of $e_s^E(x,t)$ is given in Lemma 10 of [13]. Assume that $f(x,v,0)$ satisfies the initial condition (2.4), and

$$\max_k |\tilde{X}_k(t) - X_k(t)| \le L - 2\epsilon,$$

there exists a constant $c_6(T) > 0$ independent of $h$ such that for $0 \le t \le T$

$$e_s^E(x,t) \le c_6(T)\left(1 + \frac{h}{\Delta x}\right) \max_k |\tilde{X}_k(t) - X_k(t)|, \tag{A.22}$$

where $h = \sqrt{h_x^2 + h_v^2}$.

Combining (A.21) and (A.22), we get

$$|(\tilde{E} - E)(x,t)| \ \le \ c_7(T)\left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2 + \max_k |\tilde{X}_k(t) - X_k(t)|\right). \tag{A.23}$$

Substituting (A.23) into (A.19) we have

$$\begin{aligned}
&|(\tilde{X}_k - X_k)(t)| + |(\tilde{V}_k - V_k)(t)| \\
&\le \int_0^t c_7(T)\left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2 + \max_k \left|\left(\tilde{X}_k - X_k\right)(t')\right|\right) dt' \\
&\quad + \int_0^t \left\|\frac{\partial E}{\partial x}(\cdot,t')\right\|_{L^\infty(\mathbb{R})} \left|\left(\tilde{X}_k - X_k\right)(t')\right| dt' + \int_0^t \left|\left(\tilde{V}_k - V_k\right)(t')\right| dt' \\
&< \int_0^t c_7(T)\left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2\right) dt' \\
&\quad + \int_0^t \max\left(1, \ \left\|\frac{\partial E}{\partial x}(\cdot,t')\right\|_{L^\infty(\mathbb{R})}\right)\left(\left|\left(\tilde{X}_k - X_k\right)(t')\right| + \left|\left(\tilde{V}_k - V_k\right)(t')\right|\right) dt'.
\end{aligned} \tag{A.24}$$

Setting $e(t) = \max_k(|(\tilde{X}_k - X_k)(t)| + |(\tilde{V}_k - V_k)(t)|)$, we obtain

$$e(t) \le c_8(T)\left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2\right) t + a(T)\int_0^t e(t')dt'. \tag{A.25}$$

This is a variation on Gronwall's inequality.

By solving (A.25), the trajectory error is bounded as

$$e(t) \le \frac{c_8}{a}\left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2\right)(\exp(at) - 1). \tag{A.26}$$

The above error bound is the basic convergence result for the PIC method.

By substituting (A.26) into (A.23), the error in the electric field is represented as

$$\begin{aligned}
|E(x,t) - \tilde{E}(x,t)| \le C'(T)\Bigg(&\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n \\
&+ \Delta x^2 + \left(\Delta x^m + \Delta x^n \left(\frac{h_x}{\Delta x}\right)^n + \Delta x^2\right)(\exp(at) - 1)\Bigg),
\end{aligned}$$

where $C'(T)$ are independent of $\Delta x$ and $h_x$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

## REFERENCES

[1] C. A. Anderson and C. Greengard, *On vortex methods*, SIAM J. Numer. Anal., 22 (1985), pp. 413–440.

[2] T. Armstrong and D. Montgomery, *Numerical study of weakly unstable electron plasma oscillation*, Phys. Fluids, 12 (1969), pp. 2094–2098.

[3] J. T. Beale and A. Majda, *Vortex methods* I: *Convergence in three dimensions*, Math. Comp., 39 (1982), pp. 1–27.

[4] J. T. Beale and A. Majda, *Vortex methods* II: *High order accuracy in two and three dimensions*, Math. Comp., 39 (1982), pp. 29–52.

[5] N. Besse and E. Sonnendrucker, *Semi-Lagrangian schemes for the Vlasov equation on an unstructured mesh of phase space*, J. Comput. Phys., 191 (2003), pp. 341–376.

[6] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, Institute of Physics Publishing, Bristol, UK, 1991.

[7] A. K. Chaniotis, D. Poulikakos, and P. Koumoutsakos, *Remeshed smoothed particle hydrodynamics for the simution of viscous and heat conducting flows*, J. Comput. Phys., 182 (2002), pp. 67–90.

[8] Y. Chen, S. E. Parker, G. Rewoldt, S. Ku, G. Y. Park, and C.-S. Chang, *Coarse-graining the eletron distribution in turbulence simulations of tokamak plasmas*, Phys. Plasmas, 15 (2008).

[9] C. Z. Cheng and G. Knorr, *The integration of the Vlasov equation in configuration space*, J. Comput. Phys., 22 (1976), pp. 2330–2351.

[10] I.-L. Chern and P. Colella, *A Conservative Front Tracking Method for Hyperbolic Conservation Laws*, Technical report UCRL-97200, Lawrence Livermore National Laboratory, Livermore, CA, 1987.

[11] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, *High-order finite-volume methods in mapped coordinates*, J. Comput. Phys., 230 (2011), pp. 2952–2976.

[12] G.-H. Cottet and P. D. Koumoutsakos, *Vortex Methods: Theory and Practice*, Cambridge University Press, Cambridge, UK, 2000.

[13] G.-H. Cottet and P. A. Raviart, *Particle methods for one-dimensional Vlasov-Poisson equations*, SIAM J. Numer. Anal., 21 (1984), pp. 52–76.

[14] N. Crouseilles, T. Respaud, and E. Sonnendrucker, *A forward semi-Lagrangian method for the numerical solution of the Vlasov equation*, Comput. Phys. Comm., 180 (2009), pp. 1730–1745.

[15] J. Denavit, *Numerical simulation of plasma with periodic smoothing in phase space*, J. Comput. Phys., 9 (1972), pp. 75–98.

[16] H. Dym and H. P. McKean, *Fourier Series and Integrals*, Academic Press, New York, 1972.

[17] E. Fijalkow, *A numerical solution to the Vlasov equation*, Comput. Phys. Comm., 116 (1999), pp. 319–328.

[18] F. Filbet, E. Sonnendrucker, and P. Bertrand, *Conservative numerical schemes for the Vlasov equation*, J. Comput. Phys., 172 (2001), pp. 166–187.

[19] F. Filbet and E. Sonnendrucker, *Comparision of Eulerian Vlasov solvers*, Comput. Phys. Comm., 150 (2003), pp. 247–266.

[20] K. Ganguly and H. D. Victory, Jr., *On the convergence of particle methods for multidimensional Vlasov-Poisson systems*, SIAM J. Numer. Anal., 26 (1989), pp. 249–288.

[21] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, Ph.D. thesis, Yale University, 1987.

[22] O. Hald, *The convergence of vortex methods,* II, SIAM J. Numer. Anal., 16 (1979), pp. 726–755.

[23] J. Hilditch and P. Colella, *A projection method for low Mach number fast chemistry reacting flow*, in Proceedings of the AIAA Aerospace Sciences Meeeting, Reno, NV, 1997.

[24] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, McGraw-Hill, New York, 1981.

[25] H. Johansen, *Cartesian Grid Embedded Boundary Finite Difference Method for Elliptic and Parabolic Partial Differential Equations on Irregular Domains*, Ph.D. thesis, University of California, Berkeley, CA, 1997.

[26] A. J. Klimas, *Numerical method based on the Fourier-Fourier transform approach for modeling* 1*-d electron plasma evolution*, J. Comput. Phys., 50 (1983), pp. 270–306.

[27] P. D. Koumoutsakos, *Inviscid axisymmetrization of elliptic vortex*, J. Comput. Phys., 138 (1997), pp. 821–857.

[28] D. Martin, *An Adaptive Cell-Centered Projection Method for the Incompressible Euler Equations*, Ph.D. thesis, Uiversity of California, Berkeley, CA, 1998.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/16 to 136.152.142.101. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[29] D. Martin and K. Cartwright, *Solving Poisson's Equation Using Adaptive Mesh Refinement*, Electronics Research Laboratory, report UCB/ERL M96/66, University of California, Berkeley, CA, 1996.

[30] P. McCorquodale, P. Colella, D. Grote, and J.-L. Vay, *A node-centered local refinement algorithm for Poisson's equation in complex geometries*, J. Comput. Phys., 201 (2004), pp. 34–60.

[31] B. Merriman, *Particle approximation*, Vortex Dynamics and Vortex Methods, Lecture Notes in Math. 28, Springer, New York, 1991, pp. 481–546.

[32] J. J. Monaghan, *Why particle methods work*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 422–433.

[33] J. J. Monaghan, *Extrapolating B splines for interpolation*, J. Comput. Phys., 60 (1985), pp. 253–262.

[34] J. J. Monaghan, *Particle methods for hydrodynamics*, Comput. Phys. Rep., 3 (1985), pp. 71–124.

[35] T. Nakamura and T. Yabe, *Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space*, Comput. Phys. Comm., 120 (1999), pp. 122–154.

[36] C. E. Rathmann and J. Denavit, *Simulation of collisional effects in plasmas*, J. Comput. Phys., 18 (1975), pp. 165–187.

[37] P. A. Raviart, *An analysis of particle methods*, Numerical Methods in Fluid Dynamics, Lecture Notes in Math. 1127, Springer, New York, 1985, pp. 243–324.

[38] I. J. Schoenberg, *Cardinal Spline Interpolation*, SIAM, Philadelphia, 1973.

[39] E. Sonnendrucker, J. Roche, P. Bertrand, and A. Ghizzo, *The semi-Lagrangian method for the numerical resolution of Vlasov equations*, J. Comput. Phys., 149 (1998), pp. 201–220.

[40] G. Strang, *On the construction and comparision of differene schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.

[41] E. H. Twizell, A. B. Gumel, and M. A. Arigu, *Second-order $L_0$-stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math., 6 (1996), pp. 333–352.

[42] T. Utsumi, T. Kunugi, and J. Koga, *A numerical method for solving the one-dimensional Vlasov-Poisson equation in phase space*, Comput. Phys. Comm., 108 (1998), pp. 159–179.

[43] S. Vadlamani, S. E. Parker, Y. Chen, and C. Kim, *The particle-continuum method: An algorithmic unification of particle-in-cell and continuum methods*, Comput. Phys. Comm., 164 (2004), pp. 209–213.

[44] H. D. Victory, Jr., and E. J. Allen, *The convergence theory of particle-in-cell methods for multidimensional Vlasov-Poisson systems*, SIAM J. Numer. Anal., 28 (1991), pp. 1207–1241.

[45] S. T. Zalesak, *Fully multidimensional flux-corrected transport algorithms for fluids*, J. Comput. Phys., 31 (1978), pp. 335–362.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.