A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas

Bei Wang[1] Greg Miller[2] Phil Colella[3]

[1] Princeton Institute of Computational Science and Engineering
Princeton University

[2] Department of Chemical Engineering and Materials Science
University of California at Davis

[3] Applied Numerical Algorithm Group
Lawrence Berkeley National Laboratory

Mathematical and Computer Science Approaches to High Energy Density Physics Workshop, IPAM

May 10, 2011

Outline

The Vlasov-Poisson equation

We consider a collisionless electrostatic plasma with two species, electron and ion, described by the Vlasov-Poisson equation

Vlasov equation

$$\frac{\partial f_e}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_e + \mathbf{F} \cdot \nabla_{\mathbf{v}} f_e = 0$$

$$\mathbf{F} = -\frac{q_e}{m_e}\mathbf{E}$$

Poisson equation

$$\rho = 1 - q_e \int_{\mathbb{R}^n} f_e d\mathbf{v}$$

$$\Delta \phi = -\rho \quad \mathbf{E} = -\nabla \phi$$

$f_e(\mathbf{x}, \mathbf{v}, t)$: the electron distribution function in phase space
$\mathbf{F}(\mathbf{x}, t)$: Lorentz force (electrostatic case)
$\mathbf{E}(\mathbf{x}, t)$: the self-consistent electrostatic field

Lagrangian description of the system

The distribution function f is conserved along the characteristics. At time s and t

$$f(\mathbf{X}(t), \mathbf{V}(t), t) = f(\mathbf{X}(t_0), \mathbf{V}(t_0), t_0)$$

where $(\mathbf{X}(t), \mathbf{V}(t))$ is the characteristics of the Vlasov equation:

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \mathbf{V}(t) \\ \frac{d\mathbf{V}}{dt} &= \mathbf{F}(t) \end{aligned}$$

where $\mathbf{X}(t_0) = \mathbf{x_0}$ and $\mathbf{V}(t_0) = \mathbf{v_0}$

Review of numerical methods: grid methods

1. Grid-based methods include spectral methods (Knorr 63, Armstrong 69, Flimas-Farrell 94), semi-Lagrangian methods (Cheng-Knorr 76, Sonnendrucker 99, Nakamura-Yabe 99), finite volume methods (Fijalkow 99, Filbet 01, Colella 11) and finite element methods (Zaki 88, Kilimas-Farrell 94).
2. They have drawn much attention in the past decade thanks to increasing processing power.
   - Advantage: Smooth representation of f
   - Disadvantage: High dimensions (up to 6) $\longrightarrow$ high computational cost (specifically memory)

Review of numerical methods: particle methods

Particle methods,e.g.,the PIC method,are widely used and are preferred for high dimensions.

Advantages:

- Naturally adaptive, since particles only occupy spaces where the distribution function is not zero
- Simpler to implement, in particular in high dimensions

Disadvantages:

- particle noise $\longrightarrow$ difficulties to get precise results in some cases, for example, in simulating the problems with large dynamic ranges

Particle methods

- In particle methods, we approximate the distribution function by a collection of finite-size particles

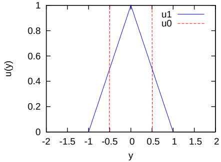
$$f(x,v,t) \approx \sum_k q_k \delta_{\varepsilon_x}(x - \tilde{X}_k(t)) \delta_{\varepsilon_v}(v - \tilde{V}_k(t))$$

$$q_k = f(x_i, v_i, 0) h_x h_v$$

where $\tilde{X}_k(0) = x_i$, $\tilde{V}_k(t) = v_i$.

$$\int_{-\infty}^{\infty} \delta_{\varepsilon_x}(y) dy = 1$$

$$\delta_{\varepsilon_x}(y) = \frac{1}{\varepsilon_x} u(\frac{y}{\varepsilon_x})$$



- At each time step, particles are transported along trajectories described by the equation of motion

$$\begin{aligned} \frac{dq_k}{dt} &= 0 \\ \frac{d\tilde{X}_k}{dt} &= \tilde{V}_k(t) \\ \frac{d\tilde{V}_k}{dt} &= \tilde{F}_k(t) \end{aligned}$$

where $\tilde{X}_k(0) = x_k$ and $\tilde{V}_k(0) = v_k$.

The PIC method

- Charge assignment:

$$\tilde{\rho}(x_j, t^n) = \sum_k \frac{q_k}{\varepsilon_x} u_1(\frac{x_j - \tilde{X}_k(t^n)}{\varepsilon_x})$$

  where j is the grid index.

- Field solver: i.e., FFTs or multigrid methods

$$\frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{{\varepsilon_x}^2} = \tilde{\rho}_j$$

$$E_j = \frac{\phi_{j-1} - \phi_{j+1}}{2\varepsilon_x}$$

- Force interpolation:

$$\tilde{E}(\tilde{X}_k, t^n) = \sum_j E_j u_1(\frac{x_j - \tilde{X}_k(t^n)}{\varepsilon_x})$$

What is particle noise?

Particle noise: the numerical error introduced when evaluating the moments of the distribution function using particles in phase space and **the particle disorder induced by the numerical error**
Error analysis:

- Monte Carlo estimate (Aydemir 93)

$$error \propto \frac{\sigma}{\sqrt{N}}$$

  where σ is the standard deviation depends on the particle sampling and the distribution function.

- The approach in vortex methods: consistency error + stability error (Cottet-Raviart 84):

$$error \propto \text{ consistency error } \times (\exp(at) - 1)$$

  where a is a physical parameter.

Error Analysis of the PIC method for the VP system

The charge density error is

$$|\rho(x,t)-\tilde{\rho}(x,t)| = |\rho(x,t)-\sum_k q_k\delta_{\varepsilon_x}(x-\tilde{X}_k(t))|$$

$$\leq \underbrace{\left|\rho(x,t)-\int_{\mathbb{R}}\rho(y,t)\delta_{\varepsilon_x}(x-y)dy\right|}_{\textbf{moment error}:e_m(x,t)\propto\varepsilon_x^2}$$

$$+\underbrace{\left|\int_{\mathbb{R}}\rho(y,t)\delta_{\varepsilon_x}(x-y)dy-\sum_k q_k\delta_{\varepsilon_x}(x-X_k(t))\right|}_{\textbf{discretization error}:e_d(x,t)\propto{\varepsilon_x}^2(\frac{h_x}{\varepsilon_x})^2}$$

$$+\underbrace{\left|\sum_k q_k\delta_{\varepsilon_x}(x-X_k(t))-\sum_k q_k\delta_{\varepsilon_x}(x-\tilde{X}_k(t))\right|}_{\textbf{stability error}:e_s(x,t)\propto\frac{1}{\varepsilon_x}\max_k|X_k-\tilde{X}_k|}$$

$$|E(x,t)-\tilde{E}(x,t)| \propto \left({\varepsilon_x}^2+{\varepsilon_x}^2\left(\frac{h_x}{\varepsilon_x}\right)^2+\max_k|\tilde{X}_k(t)-X_k(t)|\right)$$

By following Cottet and Raviart (84):

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

and

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

By using a variation on Gronwall's inequality, we get

$$\max_k(|X_k(t) - \tilde{X}_k(t)| + |V_k(t) - \tilde{V}_k(t)| + \left\|E - \tilde{E}(\cdot,t)\right\|_{L^\infty})$$
$$\leq C_1 \left[\underbrace{{\varepsilon_x}^2 + {\varepsilon_x}^2 (\frac{h_x}{\varepsilon_x})^2}_{e_c(x,t)} + \underbrace{\left({\varepsilon_x}^2 + {\varepsilon_x}^2 (\frac{h_x}{\varepsilon_x})^2\right)(\exp(at) - 1)}_{e_s(x,t)} \right]$$

where a is $\max(1, \left\|\frac{\partial E}{\partial x}(\cdot,t)\right\|_{L^\infty})$, but not ε_x and h_x.

Requirements for Convergence

- Particle overlapping: $\frac{h_x}{\varepsilon_x} \leq 1$
- Particle regularization: control the exponential-like term

ref: Cottet-Raviart 84 and Wang,Miller,Colella 11

Options if we want to reduce particle noise

- **Perturbative methods, such as the δf method** (Kotschenreuther 88, Dimits-Lee 93, Parker-Lee 93): discretize the perturbation with respect to a (local) Maxwellian in velocity space using particles

  $\Rightarrow$ reduce σ (error $\propto \frac{\sigma}{\sqrt{N}}$)

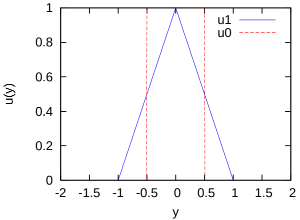
- **Our approach: remapping**: remap the distorted charge distribution on regularized grid(s) in phase space and then create a new set of particle charges from the grids with regularized distribution

  $\Rightarrow$ reduce exponential term

Conservative remapping on phase space (x, v)

Particle charges are remapped (interpolated) to a grid in phase space

$$q_{\mathbf{i}} = q_k u(\frac{x_k - x_{i_x}}{h_x}) u(\frac{v_k - v_{i_v}}{h_v})$$



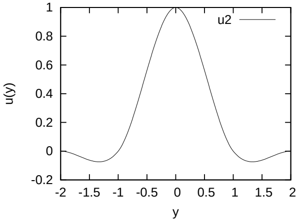
Total charge and momentum are conserved if the interpolation function u satisfies

$$\sum_{i\in\mathbb{Z}} u(\frac{x-x_i}{h}) = 1$$
$$\sum_{i\in\mathbb{Z}} x_i u(\frac{x-x_i}{h}) = x$$

High order interpolation

A modified B-spline by Monaghan (85)

$$u_2(y) = \begin{cases} 1 - \frac{5|y|^2}{2} + \frac{3|y|^3}{2} & 0 \leq |y| \leq 1 \\ \frac{1}{2}(2-|y|)^2(1-|y|) & 1 < |y| \leq 2 \\ 0 & \text{otherwise} \end{cases}$$



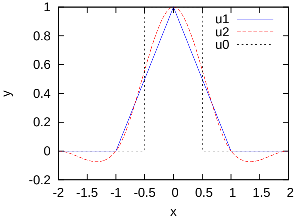
u_2 can approximate a quadratic function exactly (error is $\mathcal{O}(h^3)$). Moreover, its first and second order derivatives are continuous.

Positivity for high order interpolation functions

Algorithm: Global mass redistribution based on flux corrected transport (Zalesak 78)
Treat interpolation as advection

$$f_{\mathbf{i}}^{n+1} = f_{\mathbf{i}}^{n} + \bigtriangledown \cdot \mathbf{F}$$

$$f_{\mathbf{i}}^{n+1} = \sum_k q_k \frac{1}{h_x h_v} u_{1|2}\left(\left|\frac{\mathbf{x_i} - \mathbf{x}_k}{h}\right|\right) \quad f_{\mathbf{i}}^{n} = \sum_k q_k \frac{1}{h_x h_v} u_0\left(\left|\frac{\mathbf{x_i} - \mathbf{x}_k}{h}\right|\right)$$



Obtain the flux by solving a Poisson equation $\bigtriangledown \cdot \mathbf{F} = f_{\mathbf{i}}^{n+1} - f_{\mathbf{i}}^{n}$
Define a low order flux $\mathbf{F}^{\mathrm{lo}}$ and a high order flux $\mathbf{F}^{\mathrm{hi}}$ as being using a low order $u1$ and a high order $u2$ interpolation.
Correct the low interpolation value by using the idea of FCT

Positivity for high order interpolation functions

Algorithm: Local mass redistribution by Chern-Colella (87)

We redistribute the undershoot of cell **i**

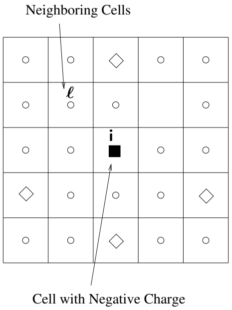
$$\delta f_{\mathbf{i}} = \min(0, f_{\mathbf{i}}^n)$$

to neighboring cells in proportion to their capacity ξ

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

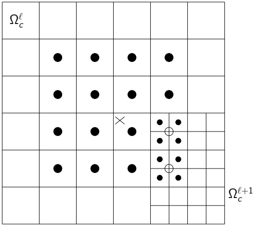
The distribution function is conserved, which fixes the constant of proportionality

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

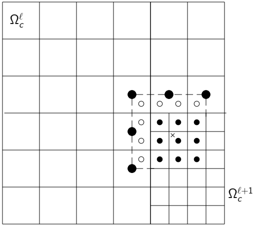


Mesh refinement

Motivation: Remapping creates too many small strength particles at the edge of the distribution function.
Algorithm: Interpolate as in uniform grid first, then transfer the charge from invalid cells to valid cells



Particle is at the coarser level side



Particle is at the finer level side

The valid deposit cells: filled circles   The invalid cell: open circles

Introduce collision term

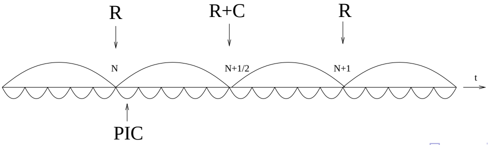
1. Remapping provides an opportunity to integrate collision models with a grid-based solver.
2. Example: simplified Fokker-Planck equation suggested by Rathmann and Denavit (75)

$$(\frac{\partial f}{\partial t})_c = \nabla_v \cdot [\nu v f + D \nabla_v (\nu f)]$$

where (ν, D) are constants.

- Discretize it using a finite volume discretization and a second order L_0 stable, implicit scheme
- Solve the matrix system using multigrid method
- Couple it with the Vlasov equation with operator splitting

$$\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} - E \cdot \frac{\partial f}{\partial v} = (\frac{\partial f}{\partial t})_c$$



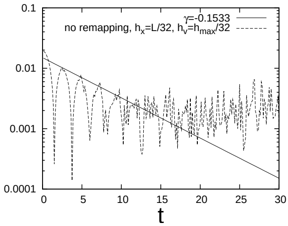
1D Vlasov-Poisson: linear Landau damping

The initial distribution of linear Landau damping is

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

where $\alpha = 0.01, k = 0.5, L = 2\pi/k$.

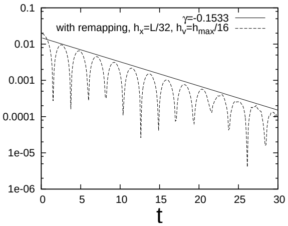
The evolution of the amplitude of the electric field: exponential decay with rate $\gamma = -0.1533$



w/o remapping

particle number: 960

CPU times: 3.99 seconds

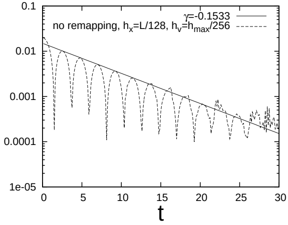


w/ remapping

particle number: 960-1,539

CPU times: 5.21 seconds

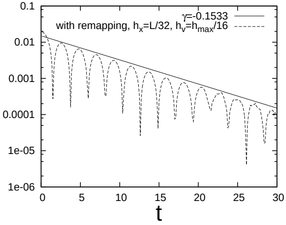
1D Vlasov-Poisson: linear Landau damping



w/o remapping

particle number: 60,416

CPU times: 141.60 seconds



w/ remapping

particle number: 1,539

CPU times: 5.21 seconds

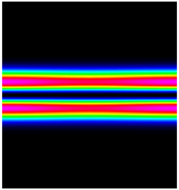
1D Vlasov-Poisson: the two stream instability

The initial distribution of the two stream instability is

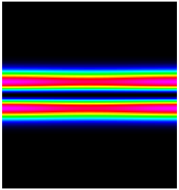
$$f_0(x, v) = \frac{1}{\sqrt{2\pi}} v^2 exp(-v^2/2)(1 + \alpha \cos(kx))$$

where $\alpha = 0.05$, $k = 0.5$, $L = 2\pi/k$.

The evolution of f(x,v,t)



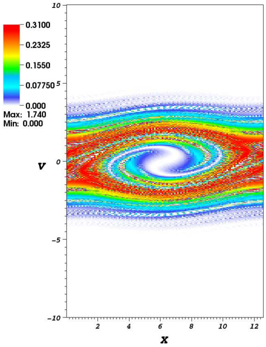
w/o remapping, particle number: 18,360



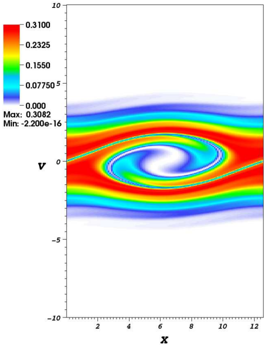
w/ remapping, particle number: 18,360-24,242

1D Vlasov-Poisson: the two stream instability

Comparison of $f(x, v, t)$ at the same instant of time $t = 20$



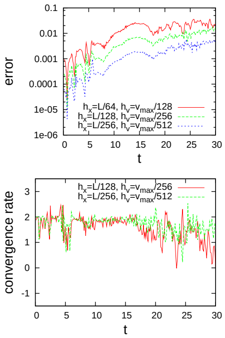
w/o remapping $\max_c$ = **1.740 vs.** $\max_e$ = **0.3**



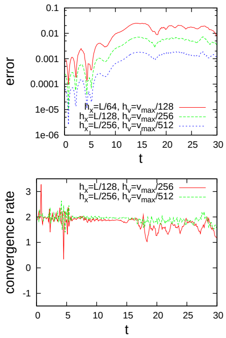
w/ remapping $\max_c$ = **0.3082 vs.** $\max_e$ = **0.3**

1D Vlasov-Poisson: the two stream instability

E field Errors and convergence rates



w/o remapping

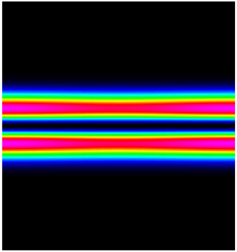


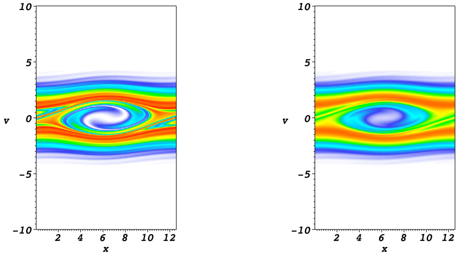
w/ remapping

1D Vlasov-Poisson-Fokker-Planck: the two stream instability

Simulation w/o (left) and w/ (right) collision at t=30

The evolution of f(x,v,t)





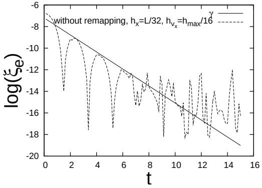
2D Vlasov-Poisson:linear Landau damping

The initial distribution of linear Landau damping in 2D is

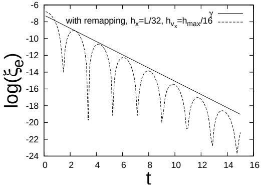
$$f_0(x, y, v_x, v_y) = \frac{1}{2\pi} \exp(-(v_x^2 + v_y^2)/2)(1 + \alpha \cos(kx)\cos(ky))$$

where $\alpha = 0.05$, $k = 0.5$, $L = 2\pi/k$.

The evolution of the electric energy ξ_e: exponential decay with constant rate γ



w/o remapping, particle number: 1,228,800



w/ remapping, particle number: 1,228,800-1,835,008

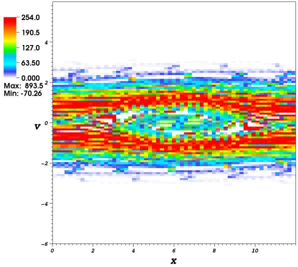
2D Vlasov-Poisson: the two stream instability

The initial distribution for the two stream instability in 2D is

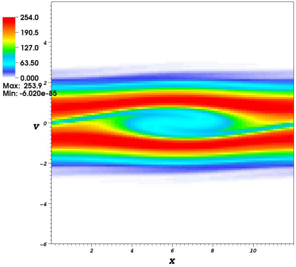
$$f_0(x, y, v_x, v_y) = \frac{1}{12\pi} \exp(-(v_x^2 + v_y^2)/2)(1 + \alpha \cos(k_x x))(1 + 5v_x^2)$$

where $\alpha = 0.05$, $k_x = 0.5$, $L = 2\pi/k$.

Comparison of projected distribution function in (x, v_x) at time $t = 20$



w/o remapping, particle number: 14,408,192



w/ remapping particle number: 14,408,192-22,184,217

Code implementation

The parallel and multidimensional Vlasov-Poisson solver is implemented using Chombo framework, a C/C++ and FORTRAN library for the solutions of partial differential equations on a hierarchy of block-structured grid with finite difference methods developed in the ANAG group of LBNL.

- The physical domain is decomposed into patches
- Each patch is assigned to a processor
- Particles are assigned to a processor according to their physical position
- Particles are transferred between patches using MPI

Continuous work in ANAG: scalable multidimensional particle in cell code with remapping

Future Work

- Adaptivity on creating a hierarchy of grids for remapping
- Apply the method to magnetic fusion plasmas, e.g., gyrokinetic particle in cell method
- Parallel scalability
- GPU acceleration of remapping

Thank you!

Questions?