Weighted Particle Method

Yoann Le Hénaff

PhD student, Rennes University, France

Dynamical systems, PDEs and Numerical Analysis
28 Sept. 2023

Content

Content

1D-1D Vlasov-Poisson equation

$$\left|\begin{array}{ll} \partial_t f(t,x,v) + v\partial_x f(t,x,v) + E(t,x)\partial_v f(t,x,v) = 0, & \text{(1.1a)} \\ E = \partial_x \Phi, \quad \partial_x^2 \Phi(t,x) = \rho(t,x), & \text{(1.1b)} \\ f(0,x,v) = f_0(x,v), & \text{(1.1c)} \end{array}\right.$$

where $t \geq 0$, $v \in \mathbb{R}$, $x \in \mathbb{T}$ and *charge density*:

$$\rho(t,x) := \int_{\mathbb{R}} \left(f(t,x,v) - \frac{1}{|\mathbb{T}|} \int_{\mathbb{T}} f(t,y,v) dy \right) dv.$$

Some properties of VP

- Transport equation $\rightarrow f$ constant along characteristics
- $(x,v) \mapsto (X(s;t,x,v), V(s;t,x,v))$ measure-preserving
- Conservation of $\mathbb{L}^p$ norms
- Conservation of energy $\mathcal{E}$ and momentum $\mathcal{M}$

$$\mathcal{E}(t) := \int |v|^2 f(t,x,v)dxdv + \int |\partial_x \Phi(t,x)|^2 \, dx = const.,$$

$$\mathcal{M}(t) = \int v f(t,x,v)dxdv = const.$$

Content

Usual numerical schemes

Splitting:

$$\partial_t f(t,x,v) + v\partial_x f(t,x,v) = 0$$
$$\implies f(t,x,v) = f(0, x - tv, v)$$

$$\partial_t f(t,x,v) + E(t,x)\partial_v f(t,x,v) = 0$$
$$\implies f(t,x,v) \approx f(0, x, v - tE(0,x))$$

Grid-based methods:

- Eulerian
- (Forward/Backward) Semi-Lagrangian

Particle methods:

- (Particle/Cloud)-In-Cell

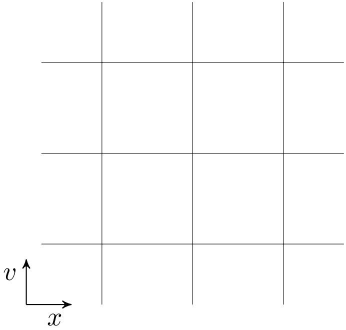
Grid-based methods

Most common: Backwards Semi-Lagrangian.

$$f(t^{n+1}, x_i, v_j) \approx f(t^n, \tilde{x}_i, \tilde{v}_j).$$

- sequence of 1D steps, even in n-D
- relies on a grid (costly for high-dimensional problems)
- interpolation introduces another source of approximations
- precision cannot be finer than grid

Lie splitting:



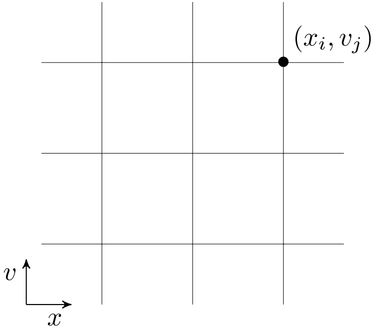
Grid-based methods

Most common: Backwards Semi-Lagrangian.

$$f(t^{n+1}, x_i, v_j) \approx f(t^n, \tilde{x}_i, \tilde{v}_j).$$

- sequence of 1D steps, even in n-D
- relies on a grid (costly for high-dimensional problems)
- interpolation introduces another source of approximations
- precision cannot be finer than grid

Lie splitting:



Grid-based methods

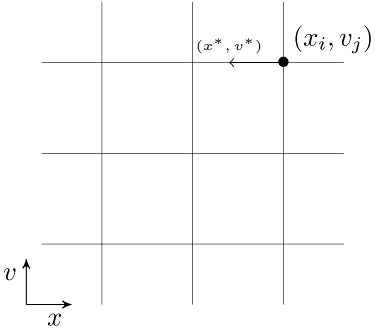
Most common: Backwards Semi-Lagrangian.

$$f(t^{n+1}, x_i, v_j) \approx f(t^n, \tilde{x}_i, \tilde{v}_j).$$

- sequence of 1D steps, even in n-D
- relies on a grid (costly for high-dimensional problems)
- interpolation introduces another source of approximations
- precision cannot be finer than grid

Lie splitting:

$$x^* := x_i + (t^{n+1} - t^n)v_j, \qquad v^* := v_j$$



Grid-based methods

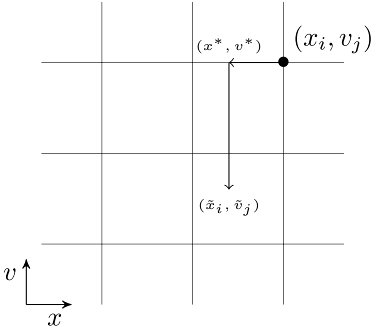
Most common: Backwards Semi-Lagrangian.

$$f(t^{n+1}, x_i, v_j) \approx f(t^n, \tilde{x}_i, \tilde{v}_j).$$

- sequence of 1D steps, even in n-D
- relies on a grid (costly for high-dimensional problems)
- interpolation introduces another source of approximations
- precision cannot be finer than grid

Lie splitting:

$$x^* := x_i + (t^{n+1} - t^n)v_j, \qquad v^* := v_j$$
$$\tilde{x}_i := x^*, \qquad \tilde{v}_j = v^* + (t^{n+1} - t^n)\partial_x \Phi[f^*]$$



Grid-based methods

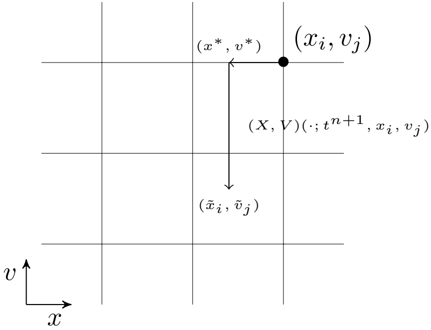
Most common: Backwards Semi-Lagrangian.

$$f(t^{n+1}, x_i, v_j) \approx f(t^n, \tilde{x}_i, \tilde{v}_j).$$

- sequence of 1D steps, even in n-D
- relies on a grid (costly for high-dimensional problems)
- interpolation introduces another source of approximations
- precision cannot be finer than grid

Lie splitting:

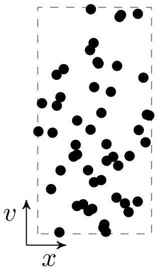
$$x^* := x_i + (t^{n+1} - t^n) v_j, \qquad v^* := v_j$$
$$\tilde{x}_i := x^*, \qquad \tilde{v}_j = v^* + (t^{n+1} - t^n)\partial_x \Phi[f^*]$$



Particle methods

Most common: Particle-In-Cell

$$f(t,x,v) \approx \sum_{p=1}^{P} w_p \delta_{X_p(t)}(x)\delta_{V_p(t)}(v).$$



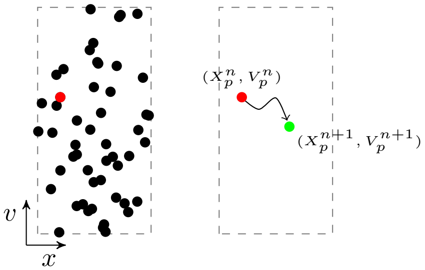
- cheap in high-dimensional settings (Monte-Carlo integrations)
- "noisy" numerical results $\rightarrow$ high number of particles

Particle methods

Most common: Particle-In-Cell

$$f(t,x,v) \approx \sum_{p=1}^{P} w_p \delta_{X_p(t)}(x)\delta_{V_p(t)}(v).$$

- cheap in high-dimensional settings (Monte-Carlo integrations)
- "noisy" numerical results → high number of particles

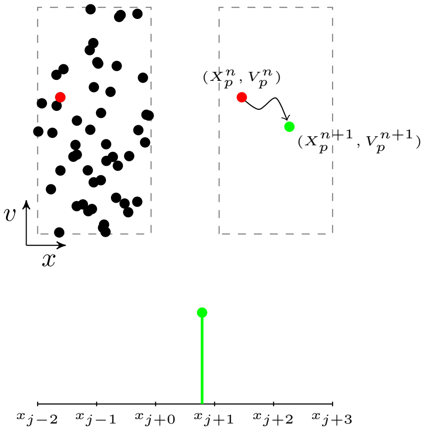


Particle methods

Most common: Particle-In-Cell

$$f(t,x,v) \approx \sum_{p=1}^{P} w_p \delta_{X_p(t)}(x)\delta_{V_p(t)}(v).$$

- cheap in high-dimensional settings (Monte-Carlo integrations)
- "noisy" numerical results $\to$ high number of particles
- the "deposition" step is difficult to analyze properly

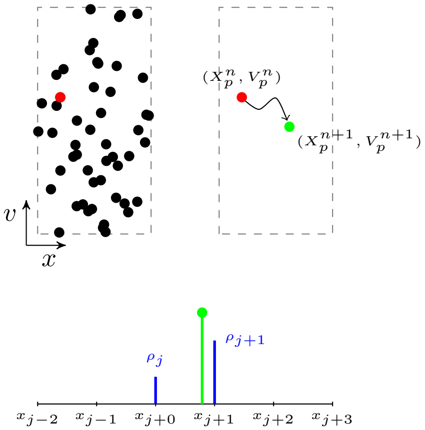


Particle methods

Most common: Particle-In-Cell

$$f(t,x,v) \approx \sum_{p=1}^{P} w_p \delta_{X_p(t)}(x)\delta_{V_p(t)}(v).$$

- cheap in high-dimensional settings (Monte-Carlo integrations)
- “noisy” numerical results → high number of particles
- the “deposition” step is difficult to analyze properly



Content

Weighted Particle Method

Initially introduced and used by Barré et al[1], later proved to converge[2].
Main idea: solve the Poisson equation explicitly in Fourier.

$$E(t,x) = \frac{1}{|\mathbb{T}|} \sum_{k \in \mathbb{Z}^*} \frac{1}{2\pi \left|\frac{k}{L}\right|^2} \frac{k}{L} \left[\sin\left(2\pi k \cdot \frac{x}{L}\right) C_k(t) - \cos\left(2\pi k \cdot \frac{x}{L}\right) S_k(t) \right],$$

where

$$C_k(t) := \int_{\mathbb{T}\times\mathbb{R}} \cos\left(2\pi k \frac{y}{L}\right) f(t,y,v) dy dv,$$

$$S_k(t) := \int_{\mathbb{T}\times\mathbb{R}} \sin\left(2\pi k \frac{y}{L}\right) f(t,y,v) dy dv.$$

[1] Julien Barré, Alain Olivetti, and Yoshiyuki Y Yamaguchi. "Algebraic Damping in the One-Dimensional Vlasov Equation". In: *Journal of Physics A: Mathematical and Theoretical* 44.40 (Oct. 2011), p. 405502. ISSN: 1751-8113, 1751-8121. DOI: 10.1088/1751-8113/44/40/405502.

[2] YLH. "Grid-free Weighted Particle method applied to the Vlasov-Poisson equation". en. In: 2022, p. 40. URL: https://hal.science/hal-03736227.

Measure-preserving property:

$$\begin{aligned} C_k(t) &= \int_{\mathbb{T}\times\mathbb{R}} \cos\left(2\pi k \frac{y}{L}\right) f(t,y,v)dydv \\ &= \int_{\mathbb{T}\times\mathbb{R}} \cos\left(2\pi k \frac{X(t;0,x,u)}{L}\right) f_0(x,u)dxdu, \end{aligned}$$

and

$$\begin{aligned} S_k(t) &= \int_{\mathbb{T}\times\mathbb{R}} \sin\left(2\pi k \frac{y}{L}\right) f(t,y,v)dydv \\ &= \int_{\mathbb{T}\times\mathbb{R}} \sin\left(2\pi k \frac{X(t;0,x,u)}{L}\right) f_0(x,u)dxdu. \end{aligned}$$

$\rightarrow$ quadrature in (x,u) $\rightarrow$ time-integration for each (x_i,u_j).

Weighted Dirac particles appear naturally!

Truncation to $|k| \leq K$, $K \in \mathbb{N}^*$

$$E^K(t,x) = \frac{1}{|\mathbb{T}|} \sum_{\substack{k \in \mathbb{Z}^* \\ |k| \leq K}} \frac{1}{2\pi \left|\frac{k}{L}\right|^2} \frac{k}{L} \left[\sin\left(2\pi k \cdot \frac{x}{L}\right) C_k^K(t) - \cos\left(2\pi k \cdot \frac{x}{L}\right) S_k^K(t)\right].$$

Truncation to $|k| \leq K$, $K \in \mathbb{N}^*$

$$E^K(t,x) = \frac{1}{|\mathbb{T}|} \sum_{\substack{k \in \mathbb{Z}^* \\ |k| \leq K}} \frac{1}{2\pi \left|\frac{k}{L}\right|^2} \frac{k}{L} \left[\sin\left(2\pi k \cdot \frac{x}{L}\right) C_k^K(t) - \cos\left(2\pi k \cdot \frac{x}{L}\right) S_k^K(t) \right].$$

$$\begin{cases} \dfrac{dX^K(t;s,x,v)}{dt} = V^K(t;s,x,v), & X^K(s;s,x,v) = x \\ \dfrac{dV^K(t;s,x,v)}{dt} = E^K(t, X^K(t;s,x,v)), & V^K(s;s,x,v) = v \end{cases}$$

Truncation to $|k| \leq K$, $K \in \mathbb{N}^*$

$$E^K(t,x) = \frac{1}{|\mathbb{T}|} \sum_{\substack{k \in \mathbb{Z}^* \\ |k| \leq K}} \frac{1}{2\pi \left|\frac{k}{L}\right|^2} \frac{k}{L} \left[\sin\left(2\pi k \cdot \frac{x}{L}\right) C_k^K(t) - \cos\left(2\pi k \cdot \frac{x}{L}\right) S_k^K(t)\right].$$

$$\begin{cases} \dfrac{dX^K(t;s,x,v)}{dt} = V^K(t;s,x,v), & X^K(s;s,x,v) = x \\ \dfrac{dV^K(t;s,x,v)}{dt} = E^K(t, X^K(t;s,x,v)), & V^K(s;s,x,v) = v \end{cases}$$

$$C_k^K(t) = \int_{\mathbb{T}^d \times \mathbb{R}^d} \cos\left(2\pi k \cdot \frac{X^K(t;0,y,v)}{L}\right) f_0(y,v) dy dv,$$

$$S_k^K(t) = \int_{\mathbb{T}^d \times \mathbb{R}^d} \sin\left(2\pi k \cdot \frac{X^K(t;0,y,v)}{L}\right) f_0(y,v) dy dv.$$

Truncation to $|k| \leq K$, $K \in \mathbb{N}^*$

$$E^K(t,x) = \frac{1}{|\mathbb{T}|} \sum_{\substack{k \in \mathbb{Z}^* \\ |k| \leq K}} \frac{1}{2\pi \left|\frac{k}{L}\right|^2} \frac{k}{L} \left[\sin\left(2\pi k \cdot \frac{x}{L}\right) C_k^K(t) - \cos\left(2\pi k \cdot \frac{x}{L}\right) S_k^K(t)\right].$$

$$\begin{cases} \dfrac{dX^K(t;s,x,v)}{dt} = V^K(t;s,x,v), & X^K(s;s,x,v) = x \\ \dfrac{dV^K(t;s,x,v)}{dt} = E^K(t, X^K(t;s,x,v)), & V^K(s;s,x,v) = v \end{cases}$$

$$C_k^K(t) = \int_{\mathbb{T}^d \times \mathbb{R}^d} \cos\left(2\pi k \cdot \frac{X^K(t;0,y,v)}{L}\right) f_0(y,v) dy dv,$$

$$S_k^K(t) = \int_{\mathbb{T}^d \times \mathbb{R}^d} \sin\left(2\pi k \cdot \frac{X^K(t;0,y,v)}{L}\right) f_0(y,v) dy dv.$$

Then, time discretization

Theorem (Convergence of the Weighted Particle method)

Let $j \in \mathbb{N}$ such that $j \geq 1 + \max_i q_i$, and $\nu, r, \alpha \in \mathbb{N}$ such that $\nu + j > d/2$, $r \geq \max(3(\nu+j), (j-1)(d+1))$, $\alpha \geq 2(r+d)$. Let $K \in \mathbb{N}$, and assume $f_0 \in \mathcal{H}^{r+\alpha}_{\nu+j}$.
Then there exists a constant $C > 0$ such that the following holds: for $\delta \geq 0$, define finite intervals $I_{d+1} := [a_1, b_1], \ldots, I_{2d} = [a_d, b_d]$ and $I_v := I_{d+1} \times \cdots \times I_{2d}$ such that

$$||f_0||_{\mathcal{H}^0_\nu(\mathbb{T}^d_L \times (\mathbb{R}^d \setminus I_v))} \leq \delta.$$

Then for all $K \in \mathbb{N}^*$, and $n = 1, \ldots, N_t$

$$\max_{p=1,\ldots,P} \left(\left| X_p^{K,n} - X(t^n; 0, x_p, v_p) \right| + \left| V_p^{K,n} - V(t^n; 0, x_p, v_p) \right| \right)$$
$$\leq C \left(K^d \left[\delta + K^{\gamma+1} \Delta t^\gamma + \sum_{i=1}^{2d} K^{q_i} \Delta z_i^{q_i} \right] + \frac{1}{(1+K)^{\frac{\alpha+1}{2} - d}} \right)$$

where C is independent of $n, \Delta t, \Delta z_i, K$.

A few remarks

- Need characteristics forward in time (easy).
- Choice of quadratures & time-integration (high-order, can even be symplectic thanks to Hamiltonian structure!).
- Non-Uniform Fast Fourier Transform (NUFFT) can be used to accelerate computation of C^K, S^K.
- Particle method with no deposition step, so provable rates of convergence.

Content

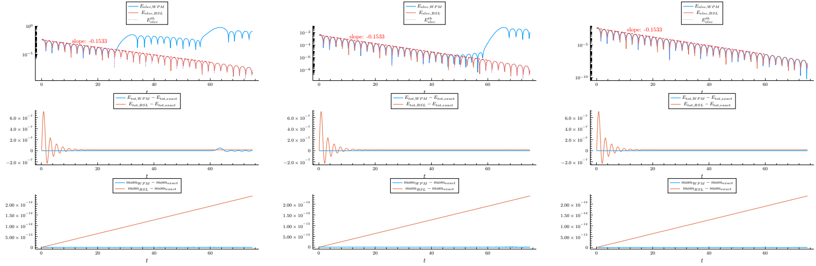
Weak Landau damping

$$f_0(x,v) = (1+\alpha\cos(k_x x))\exp(-v^2/2)\frac{1}{\sqrt{2\pi}}, \quad x \in [0,L], v \in [-v_{\max}, v_{\max}].$$

The numerical parameters are

$$v_{\max} = 12, k_x = 0.5, \alpha = 0.001, \Delta t = 0.1.$$

Weak Landau damping ($K = 1$)

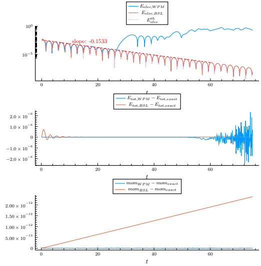


(a) $K = 1$, $N_1 = 64$, $N_2 = 64$

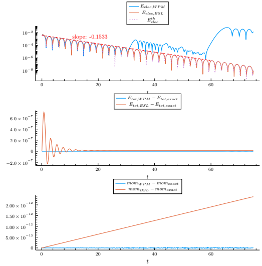
(b) $K = 1$ $N_1 = 128$, $N_2 = 128$

(c) $K = 1$ $N_1 = 256$, $N_2 = 256$

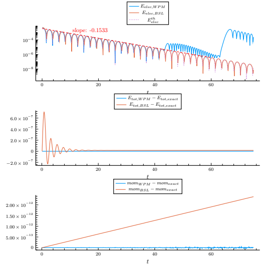
Weak Landau damping ($K = 15$)



(a) $K = 15$, $N_1 = 64$, $N_2 = 64$



(b) $K = 15$ $N_1 = 128$, $N_2 = 128$



(c) $K = 15$ $N_1 = 256$, $N_2 = 256$

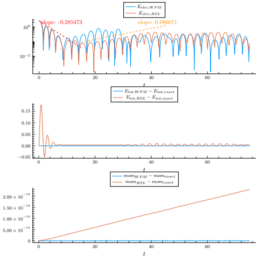
Strong Landau damping

$$f_0(x,v) = (1+\alpha\cos(k_x x))\exp(-v^2/2)\frac{1}{\sqrt{2\pi}}, \quad x \in [0,L], v \in [-v_{\max}, v_{\max}].$$

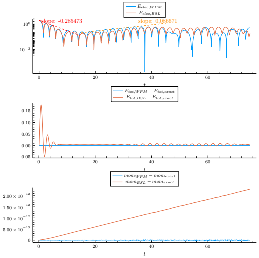
The numerical parameters are

$$v_{\max} = 12, k_x = 0.5, \alpha = 0.5, \Delta t = 0.1.$$

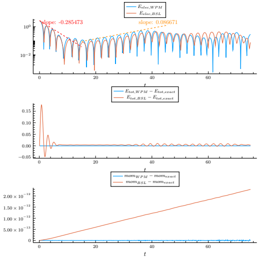
Strong Landau damping ($K = 1$)



(a) $K = 1,\ N_1 = 64,\ N_2 = 64$

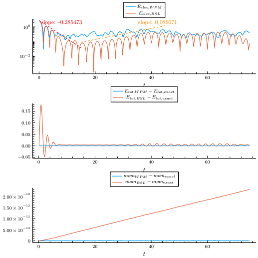


(b) $K = 1,\ N_1 = 128,\ N_2 = 128$

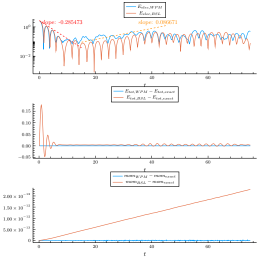


(c) $K = 1,\ N_1 = 256,\ N_2 = 256$

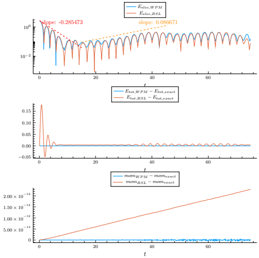
Strong Landau damping ($K = 15$)



(a) $K = 15$, $N_1 = 64$, $N_2 = 64$



(b) $K = 15$, $N_1 = 128$, $N_2 = 128$



(c) $K = 15$, $N_1 = 256$, $N_2 = 256$

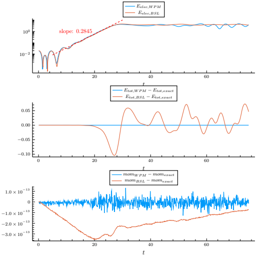
Two-stream Instability

$$f_0(x,v) = (1+\alpha\cos(k_x x))\frac{1}{2\sqrt{2\pi}}\left(\exp\left(-\frac{(v-v_0)^2}{2}\right)+\exp\left(-\frac{(v+v_0)^2}{2}\right)\right),$$

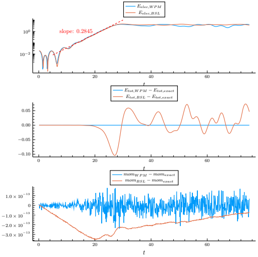
for $x \in [0, 2\pi/k_x], v \in [-v_{\max}, v_{\max}]$. The numerical parameters are

$$\alpha = 0.001, v_{\max} = 12, k_x = 0.2, v_0 = 3, \Delta t = 0.1.$$

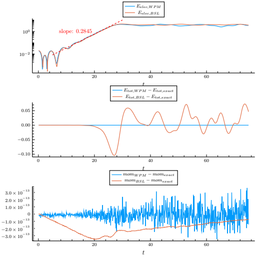
Two-stream Instability ($K = 1$)



(a) $K = 1,\ N_1 = 64,$ $N_2 = 64$

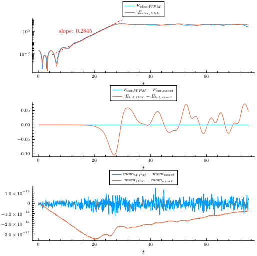


(b) $K = 1,\ N_1 = 128,$ $N_2 = 128$

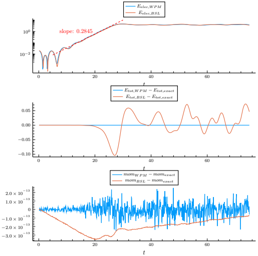


(c) $K = 1,\ N_1 = 256,$ $N_2 = 256$

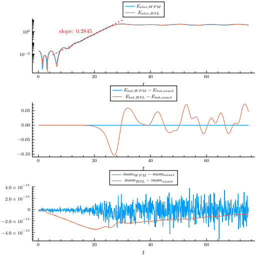
Two-stream instability ($K = 15$)



(a) $K = 15$, $N_1 = 64$, $N_2 = 64$



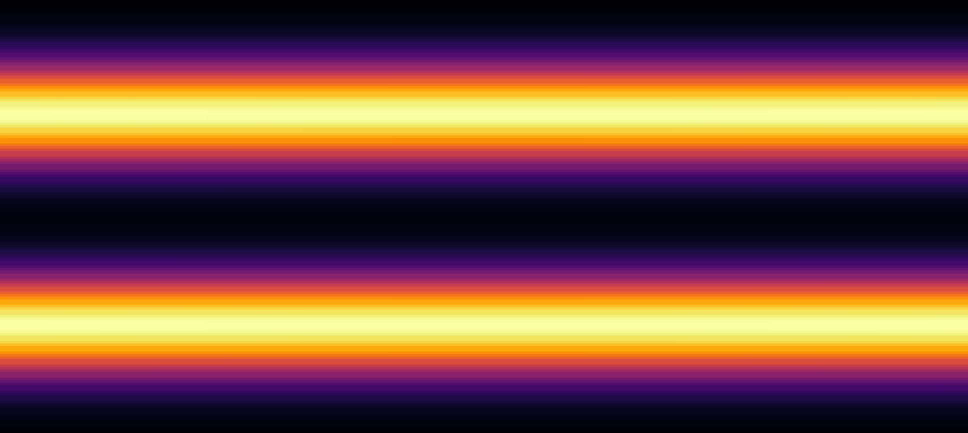
(b) $K = 15$, $N_1 = 128$, $N_2 = 128$



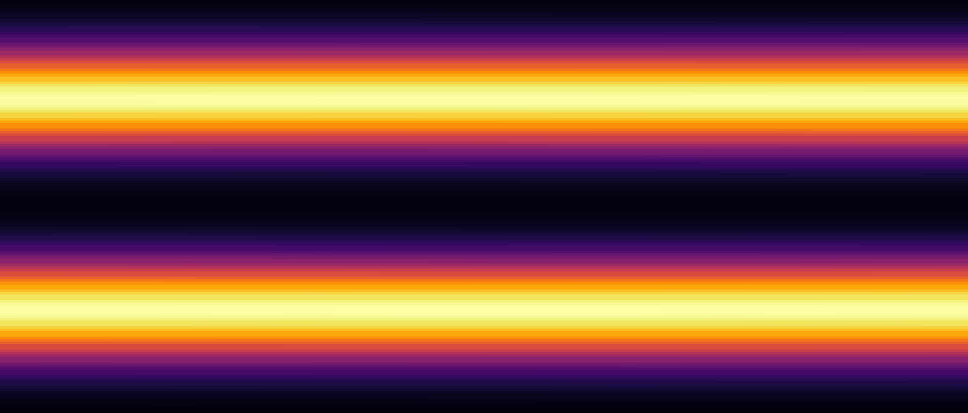
(c) $K = 15$, $N_1 = 256$, $N_2 = 256$

Perspectives

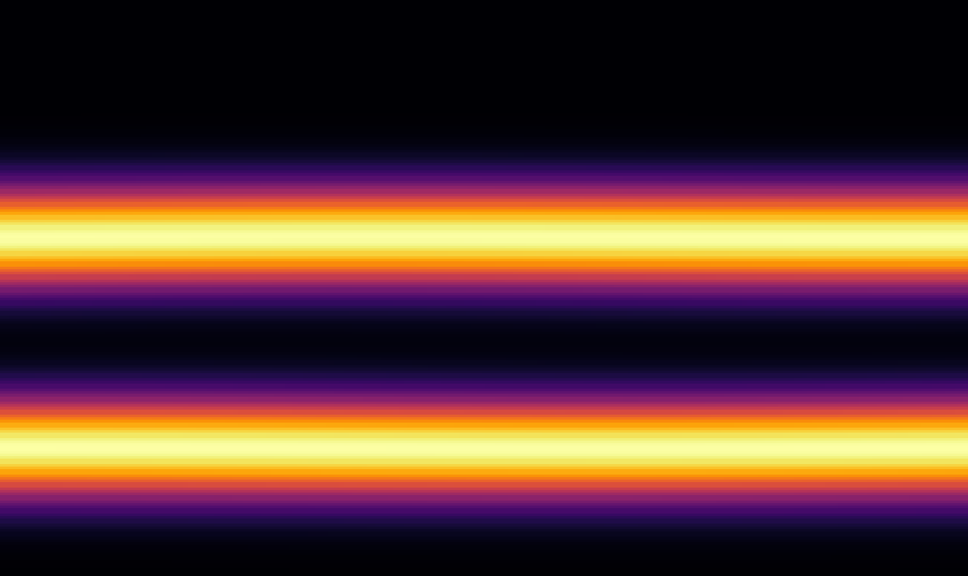
- Use Monte-Carlo integration
- Consider collisions in (1.1a)
- Compare with PIC and BSL in high-dimension



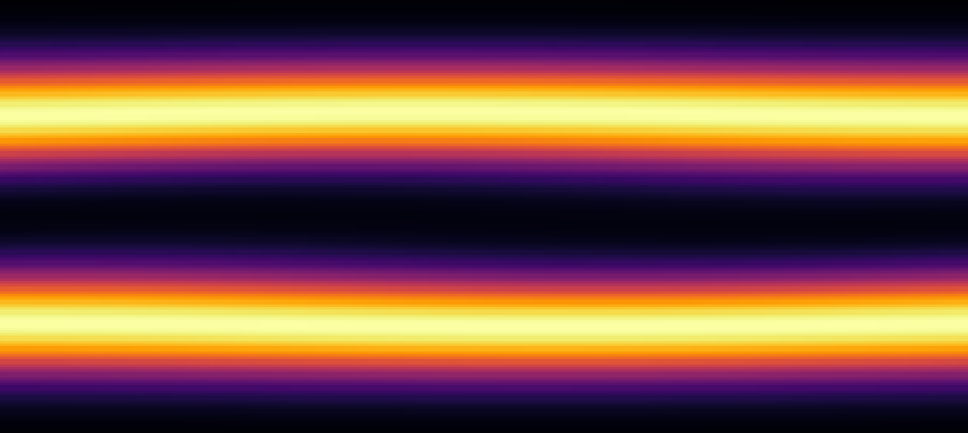
Two-Stream instability, triangulated WPM, $T = 0$, $K = 15$, 256×256 pts



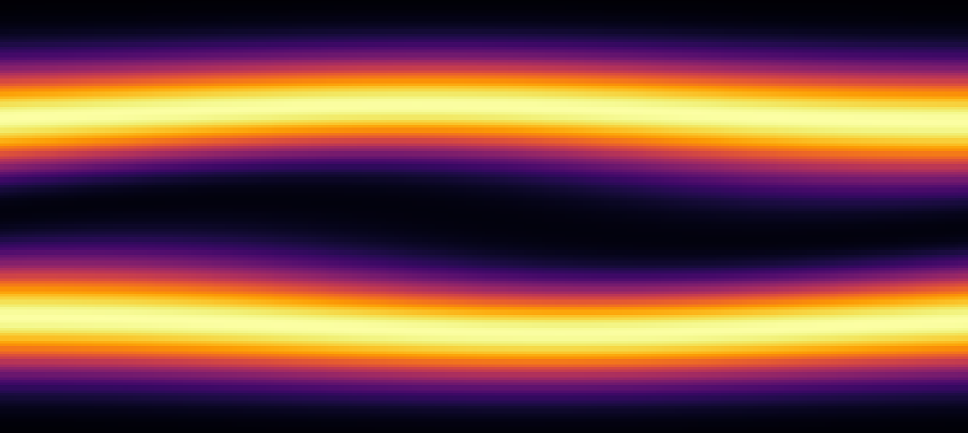
Two-Stream instability, triangulated WPM, $T = 5$, $K = 15$, 256×256 pts



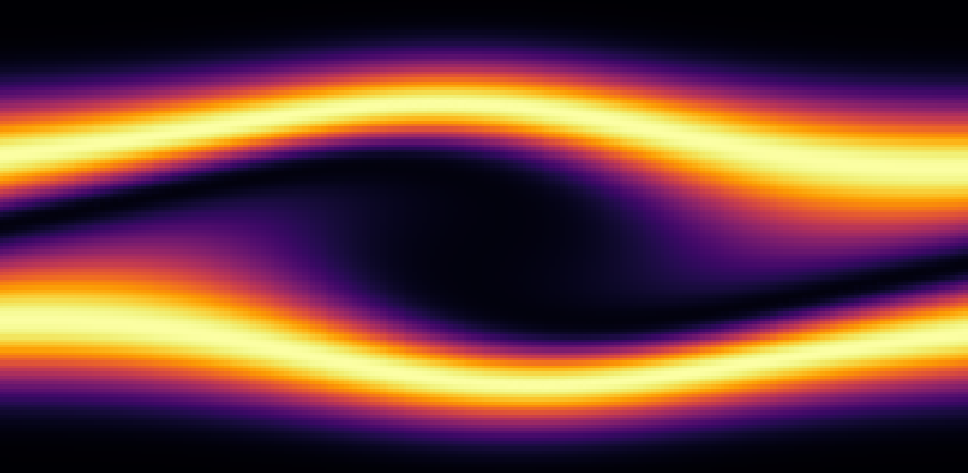
Two-Stream instability, triangulated WPM, $T = 10$, $K = 15$, 256×256 pts



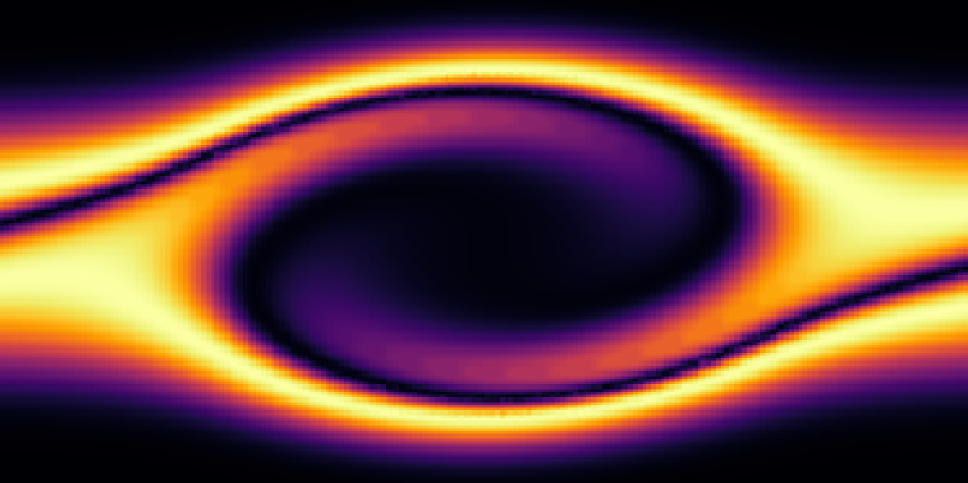
Two-Stream instability, triangulated WPM, $T = 15$, $K = 15$, 256×256 pts



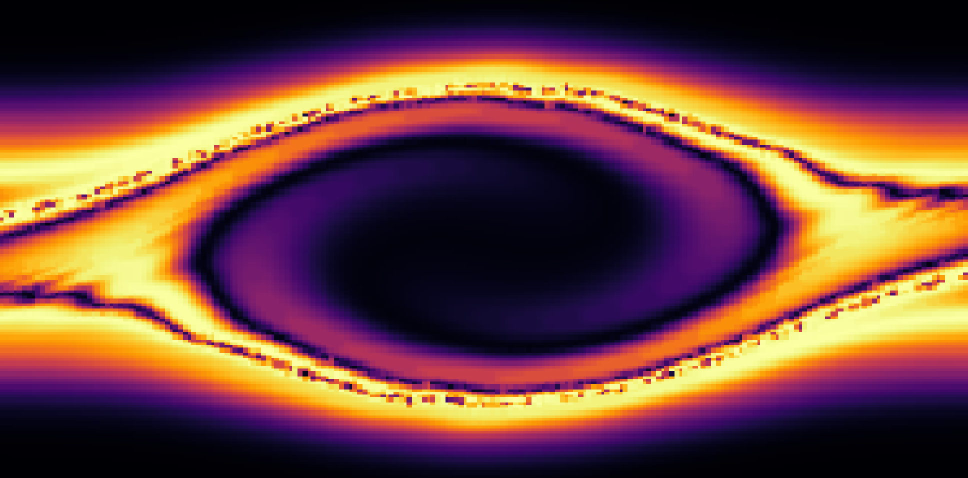
Two-Stream instability, triangulated WPM, $T = 20$, $K = 15$, 256×256 pts



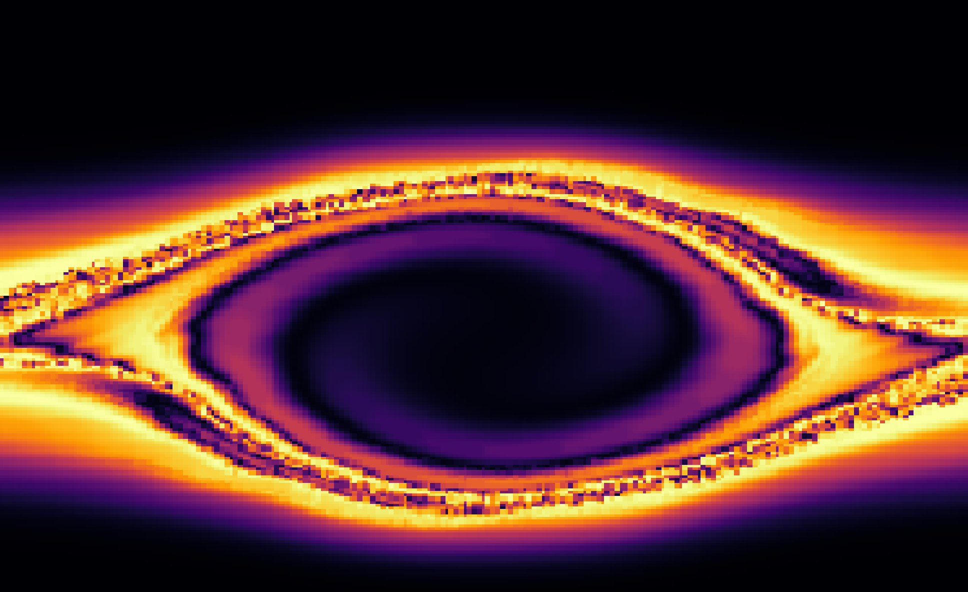
Two-Stream instability, triangulated WPM, $T = 25$, $K = 15$, 256×256 pts



Two-Stream instability, triangulated WPM, $T = 30$, $K = 15$, 256×256 pts



Two-Stream instability, triangulated WPM, $T = 40$, $K = 15$, 256×256 pts



Two-Stream instability, triangulated WPM, $T = 50$, $K = 15$, 256×256 pts

Thank you!

Content

Weighted Sobolev spaces

Weighted Sobolev spaces

$$||f||^2_{\mathcal{H}^r_\nu} = \sum_{\substack{(m,p,q)\in(\mathbb{N}^d)^3 \\ |p|+|q|\leq r \\ |m|\leq \nu}} \int_{\mathbb{T}\times\mathbb{R}} \left|v^m \partial_x^p \partial_v^q f(x,v)\right|^2 dv dx.$$