Semi-Lagrangian Methods Based on Cartesian Mesh for Plasma Turbulence

Francis FILBET, Chang YANG

University of Lyon

Munich 2013
Outline

I. Semi-Lagrangian Method for Vlasov Equation

II. Diocotron Instability Simulation

III. Ion Turbulence Simulation

IV. Conclusion and Perspectives
The evolution of the density of particles $f(t, x, v)$ in the phase space $(x, v) \in \mathbb{R}^3 \times \mathbb{R}^3$, can be described by the Vlasov equation,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F(t, x, v) \cdot \nabla_v f = 0,$$

where the force field $F(t, x, v)$ is coupled with the distribution function $f$ giving a non linear system. Vlasov Equation (1) has form

$$\frac{\partial f}{\partial t} + A \cdot \nabla f = 0,$$

where $f : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}$ and $A : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}^d$. For given $s \in \mathbb{R}^+$, the differential system

$$\begin{cases}
\frac{dx}{dt} = A(t, X), \\
X(s) = x,
\end{cases}$$

is associated with the transport equation (2). We denote its solution by $X(t; s, x)$. 

Vlasov Equation
The classical semi-Lagrangian method is decomposed into two steps for computing $f^{n+1}$ from $f^n$: (cf. : Sonnendrücker et al., Filbet et al.)

1. For each mesh point $x_i$ of phase space, compute $X(t_n; t_{n+1}, x_i)$.
2. We obtain the value of $f^{n+1}(x_i)$ by computing $f^n(X(t_n; t_{n+1}, x_i))$ by interpolation.

Interpolation methods:
- Cubic spline interpolation: the most used, but has spurious oscillations, high communication
- Lagrange interpolation $P_3$
- Hermite interpolation $H_3$
Lagrange WENO Interpolation $\tilde{P}_3$

To construct $f^n(x)$ in $[x_i, x_{i+1}]$, we define firstly Lagrange polynomials $p_l, p_r, p_3$, which verify:

\[
\begin{align*}
p_r(x_i) &= f_i, \quad p_r(x_{i+1}) = f_{i+1}, \quad p_r(x_{i+2}) = f_{i+2}, \\
p_l(x_{i-1}) &= f_{i-1}, \quad p_l(x_i) = f_i, \quad p_l(x_{i+1}) = f_{i+1}, \\
P_3(x_{i-1}) &= f_{i-1}, \quad P_3(x_i) = f_i, \quad P_3(x_{i+1}) = f_{i+1}, \quad P_3(x_{i+2}) = f_{i+2}.
\end{align*}
\]

Then Lagrange WENO (LWENO) interpolation is written as

\[
\tilde{P}_3(x) = w_l(x) p_l(x) + w_r(x) p_r(x).
\]

where $w_l$ and $w_r$ are WENO weights such that:

- In the case “$f$ is smooth” in $S_3$,
  \[
  w_l(x) \approx c_l(x) = \frac{x_{i+2} - x}{3\Delta x}, \quad w_r(x) \approx c_r(x) = \frac{x - x_{i-1}}{3\Delta x}.
  \]

- In the case “$f$ is smooth” in $S_2^l$ or $S_2^r$,
  \[
  w_l(x) \approx 1, \quad w_r(x) \approx 0 \quad \text{or} \quad w_l(x) \approx 0, \quad w_r(x) \approx 1.
  \]
Computation of Weights and Smoothness Indicators

To measure the smoothness, we introduce smoothness indicators

\[
\beta_l = \int_{x_i}^{x_{i+1}} \Delta x (p'_l)^2 + \Delta x^3 (p''_l)^2 \, dx = \frac{13}{12} (f_{i-1} - 2f_i + f_{i+1})^2 + (f_{i+1} - f_i)^2,
\]

\[
\beta_r = \int_{x_i}^{x_{i+1}} \Delta x (p'_r)^2 + \Delta x^3 (p''_r)^2 \, dx = \frac{13}{12} (f_i - 2f_{i+1} + f_{i+2})^2 + (f_{i+1} - f_i)^2.
\]

WENO weights are given by :

\[
w_l = \frac{\alpha_l}{\alpha_l + \alpha_r}, \quad w_r = 1 - w_l,
\]

where

\[
\alpha_l = \frac{c_l}{(\varepsilon + \beta_l)^2}, \quad \alpha_r = \frac{c_r}{(\varepsilon + \beta_r)^2}.
\]

For a fixed \(x_p \in [x_i, x_{i+1}]\),

1. if “f is smooth” in the stencil \(S_3\), then 
   \[f(x_p) - \tilde{P}_3(x_p) = \mathcal{O}(\Delta x^4);\]
2. if “f is at least smooth“ in one of stencils \(S_2^l\) or \(S_2^r\), then
   \[f(x_p) - \tilde{P}_3(x_p) = \mathcal{O}(\Delta x^3).\]
Hermite WENO Interpolation $\tilde{H}_3$

To construct $\tilde{H}_3$ in $[x_i, x_{i+1}]$, we define

\[
H_3(x_i) = f_i, \quad H'_3(x_i) = f'_i, \quad H_3(x_{i+1}) = f_{i+1}, \quad H'_3(x_{i+1}) = f'_{i+1},
\]
\[
h_l(x_i) = f_i, \quad h_l(x_{i+1}) = f_{i+1}, \quad h'_l(x_i) = f'_i,
\]
\[
h_r(x_i) = f_i, \quad h_r(x_{i+1}) = f_{i+1}, \quad h'_r(x_{i+1}) = f'_{i+1}.
\]

Assuming $f'_i$ are known, we introduce Hermite WENO (HWENO1) interpolation

\[
\tilde{H}_3(x) = w_l(x) h_l(x) + w_r(x) h_r(x),
\]

where $w_l$ and $w_r$ are computed as previously, but with a new smoothness indicators $\beta_l, \beta_r$ and convex coefficients $c_l, c_r$

\[
\beta_l = (f_i - f_{i+1})^2 + \frac{13}{3}((f_{i+1} - f_i) - \Delta x f'_i)^2, \quad c_l = \frac{x_{i+1} - x}{\Delta x},
\]
\[
\beta_r = (f_i - f_{i+1})^2 + \frac{13}{3}((f_{i+1} - f_i) - \Delta x f'_{i+1})^2, \quad c_r = \frac{x - x_i}{\Delta x}.
\]
Computation of First Derivative

We use LWENO interpolation to compute first derivatives $\tilde{f}'_i$.

$$f'_l(x_i) = \frac{1}{6\Delta x} (-2f_{i-1} - 3f_i + 6f_{i+1} - f_{i+2}),$$

$$f'_r(x_i) = \frac{1}{6\Delta x} (f_{i-2} - 6f_{i-1} + 3f_i + 2f_{i+1}),$$

$$f'_i = \frac{1}{12\Delta x} (8(f_{i+1} - f_{i-1}) - (f_{i+2} - f_{i-2})).$$

We reduce that

$$f'_i = \frac{1}{2} f'_l(x_i) + \frac{1}{2} f'_r(x_i).$$

Thus optimized first derivative is

$$\tilde{f}'_i = w_l(x_i) f'_l(x_i) + w_r(x_i) f'_r(x_i),$$

where $w_l(x_i), w_r(x_i)$ are LWENO type weights.
Modified Hermite WENO Interpolation

We propose a slightly modified Hermite WENO (HWENO2) interpolation. In the interval \([x_i, x_{i+1}]\), we modify \(\tilde{f}'_i\) and \(\tilde{f}'_{i+1}\) as follows:

- If \(f'_l(x_i) \cdot f'_r(x_i) \leq 0\) or \(f'_l(x_{i+1}) \cdot f'_r(x_{i+1}) \leq 0\)
  \[\tilde{f}'_i = \tilde{f}'_{i+1} = \frac{f_{i+1} - f_i}{\Delta x},\]

- Otherwise
  \[
  \begin{align*}
  \tilde{f}'_i &= w_l(x_i) f'_l(x_i) + w_r(x_i) f'_r(x_i), \\
  \tilde{f}'_{i+1} &= w_l(x_{i+1}) f'_l(x_{i+1}) + w_r(x_{i+1}) f'_r(x_{i+1}).
  \end{align*}
  \]

**Properties of HWENO2 interpolation:**

1. For \(\Delta x\) sufficiently small, HWENO2 interpolation has the same precision as HWENO1 interpolation if “\(f\) is smooth”;
2. HWENO2 interpolation is less oscillating than HWENO1 interpolation.
1D Test

We consider 1D transport equation

$$\partial_t f + v \partial_x f = 0, \quad x \in [0, 1], \quad t \geq 0.$$  

The periodic boundary condition is used.

- Computational time for $n_x = 1024$

<table>
<thead>
<tr>
<th></th>
<th>Spline</th>
<th>Lagrange</th>
<th>LWENO</th>
<th>Hermite</th>
<th>HWENO1</th>
<th>HWENO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>1.87</td>
<td>1.65</td>
<td>1.65</td>
<td>1.66</td>
<td>1.68</td>
<td>1.68</td>
</tr>
</tbody>
</table>

- Error between exact solution and approximated solution for smooth solution case

<table>
<thead>
<tr>
<th>$n_x$</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$| \cdot |_1$</td>
<td>$r$</td>
<td>$| \cdot |_1$</td>
<td>$r$</td>
</tr>
<tr>
<td>Spline</td>
<td>1.28e-7</td>
<td>2.94</td>
<td>1.60e-8</td>
<td>3.00</td>
</tr>
<tr>
<td>Lagrange</td>
<td>1.53e-6</td>
<td>2.94</td>
<td>1.92e-7</td>
<td>3.00</td>
</tr>
<tr>
<td>LWENO</td>
<td>1.55e-6</td>
<td>2.99</td>
<td>1.92e-7</td>
<td>3.01</td>
</tr>
<tr>
<td>Hermite</td>
<td>1.30e-7</td>
<td>2.99</td>
<td>1.61e-8</td>
<td>3.01</td>
</tr>
<tr>
<td>HWENO1</td>
<td>1.31e-7</td>
<td>3.04</td>
<td>1.61e-8</td>
<td>3.00</td>
</tr>
<tr>
<td>HWENO2</td>
<td>1.31e-7</td>
<td>3.04</td>
<td>1.61e-8</td>
<td>3.00</td>
</tr>
</tbody>
</table>
Error between exact solution and approximated solution for discontinuous solution case

<table>
<thead>
<tr>
<th>$n_x$</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|\cdot|_1$</td>
<td>$r$</td>
<td>$|\cdot|_1$</td>
<td>$r$</td>
</tr>
<tr>
<td>Spline</td>
<td>1.25e-2</td>
<td>0.89</td>
<td>6.69e-3</td>
<td>0.91</td>
</tr>
<tr>
<td>Lagrange</td>
<td>1.43e-2</td>
<td>0.65</td>
<td>8.78e-3</td>
<td>0.70</td>
</tr>
<tr>
<td>LWENO</td>
<td>1.72e-2</td>
<td>0.67</td>
<td>1.07e-2</td>
<td>0.67</td>
</tr>
<tr>
<td>Hermite</td>
<td>1.37e-2</td>
<td>0.90</td>
<td>7.78e-3</td>
<td>0.82</td>
</tr>
<tr>
<td>HWENO1</td>
<td>1.60e-2</td>
<td>0.74</td>
<td>9.70e-3</td>
<td>0.72</td>
</tr>
<tr>
<td>HWENO2</td>
<td>1.63e-2</td>
<td>0.75</td>
<td>9.61e-3</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Total variation of discontinuous solution wrt exact total variation

<table>
<thead>
<tr>
<th>$n_x$</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline</td>
<td>9.75e-1</td>
<td>8.25e-1</td>
<td>9.08e-1</td>
<td>7.90e-1</td>
</tr>
<tr>
<td>Lagrange</td>
<td>4.90e-1</td>
<td>4.65e-1</td>
<td>4.65e-1</td>
<td>4.83e-1</td>
</tr>
<tr>
<td>LWENO</td>
<td>2.87e-5</td>
<td>6.77e-5</td>
<td>1.22e-4</td>
<td>1.87e-4</td>
</tr>
<tr>
<td>Hermite</td>
<td>8.93e-1</td>
<td>9.87e-1</td>
<td>9.25e-1</td>
<td>9.75e-1</td>
</tr>
<tr>
<td>HWENO1</td>
<td>7.74e-4</td>
<td>1.41e-3</td>
<td>2.05e-3</td>
<td>2.73e-3</td>
</tr>
<tr>
<td>HWENO2</td>
<td>-4.44e-16</td>
<td>8.88e-16</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
1D Test

We take $n_x = 128$, cfl=10.
Intermediate Conclusion

From numerical evidence, we observe:

- 3rd order method in space for smooth solution.
- Control of spurious oscillations, i.e. control of TV.
- Preserve positivity, i.e.
  \[ f_0 \geq 0 \implies f(t) \geq 0. \]
- Control maximum, i.e.
  \[ \|f(t)\|_\infty \leq \|f_0\|_\infty. \]
- Mass conservation is observed for free transport.

Prove the above properties of our schemes.
Guiding-Center Model

The Guiding-Center model has been derived to describe highly magnetized plasma in the transverse plane of a Tokamak.

\[
\begin{cases}
\frac{\partial \rho}{\partial t} + U \cdot \nabla_{\mathbf{x}_\perp} \rho = 0, \\
U = E^\perp, \\
-\Delta_{\mathbf{x}_\perp} \phi = \rho.
\end{cases}
\]

Boundary condition :

\[\phi(\mathbf{x}_\perp) = 0, \quad \mathbf{x}_\perp \in \partial D,\]

where \(\partial D\) can be arbitrary boundary.

If \(f\) is smooth, we have

(1) Maximum principle : \(0 \leq \rho(t, \mathbf{x}_\perp) \leq \max_{\mathbf{x}_\perp \in D}(\rho(0, \mathbf{x}_\perp))\).

(2) \(L^p\) norm conservation : \(\frac{d}{dt} \left( \int_D (\rho(t, \mathbf{x}_\perp))^p \, d\mathbf{x}_\perp \right) = 0\).

(3) Energy conservation : \(\frac{d}{dt} \left( \int_D |\nabla \phi|^2 \, d\mathbf{x}_\perp \right) = 0\).
We use the semi-Lagrangian method for solving the transport equation.

1. To find the characteristic foot is equivalent to solve

\[
\begin{align*}
\frac{d}{dt} X &= U(t, X), \\
X(s) &= x_\perp.
\end{align*}
\]

This system can be solved by using the parabolic assumption: A second order scheme reads

\[
\frac{x_\perp - X(t^{n-1}, t^{n+1}, x_\perp)}{2\Delta t} = U(X(t^n, t^{n+1}, x_\perp), t_n).
\]

Assuming that \(U\) is constant between \(t_{n+1}\) and \(t_{n-1}\), we get

\[
d = \Delta t U(x_\perp - d, t_n),
\]

where \(d\) is the shift vector in \(x_\perp\) plane.

This equation can be solved by a Taylor method.

2. HWENO2 interpolation method is used.

2D interpolation can be proceeded dimension by dimension.
Discretization of Poisson Equation

We discretize Poisson equation in an arbitrary domain.

- Classical five points finite difference scheme is used.
- Extrapolation technique for treatment of B.C.

We extrapolate $\phi_{i,j-1}$ on the normal direction $n$

$$\phi_{i,j-1} = \tilde{w}_p \phi(x_p) + \tilde{w}_h \phi(x_h) + \tilde{w}_{2h} \phi(x_{2h}),$$

where

- $\phi(x_p)$ is known by Dirichlet B.C.
- $\phi(x_h), \phi(x_{2h})$ are determined by interpolation.

Therefore, $\phi_{i,j-1}$ is approximated from the interior domain.

- is interior point, ■ is ghost point, □ is the point at the boundary, ○ is the point for interpolation, the dashed line is the boundary.
Convergence of Guiding-Center Model

For a smooth solution, we have
We now consider the diocotron instability for an annular electron layer. The initial data is given by (cf. : Pétri, Mehrenberger)

\[ \rho_0(x) = \begin{cases} 
1 + \varepsilon \cos(\ell \theta), & \text{if } r^- \leq \sqrt{x^2 + y^2} \leq r^+, \\
0, & \text{otherwise}, 
\end{cases} \]

where \( \ell = 7 \) and \( \varepsilon = 0.1 \).

Let us consider a disk domain \( D = \{(x,y) \in \mathbb{R}^2: \sqrt{x^2 + y^2} \leq R\} \). cfl \( \approx 10 \).
Evolution of Diocotron instability simulation
Enforce Mass Conservation

Note that the semi-Lagrangian method does not preserve mass. We thus add a least square method, denoted by LS, to enforce the mass conservation.

Suppose that the density $\rho^{n+1}$ is obtained by the semi-Lagrangian method, then in this particular case the LS procedure reads

$$\text{LS}(\rho) = \frac{M^n}{M^{n+1}} \rho^{n+1},$$

where $M^n = \int_{\Omega_x} \rho^n dx_\perp$.

Without LS procedure | With LS procedure
Enforce Mass Conservation

Without LS procedure

With LS procedure
Normalized Drift-Kinetic model reads (cf. Grandgirard *et al.*)

\[
\begin{align*}
\frac{\partial f}{\partial t} + \mathbf{U} \cdot \nabla_{\mathbf{x}} f + \mathbf{v} \cdot \nabla_z f + E \cdot \nabla_\mathbf{v} f &= 0, \\
\mathbf{U} &= \frac{\mathbf{E} \times \mathbf{B}}{B^2}, \\
-\nabla_\perp \cdot \left( \frac{\rho_0(\mathbf{x}_\perp)}{B} \nabla_\perp \phi \right) + \frac{\rho_0(\mathbf{x}_\perp)}{T_e(\mathbf{x}_\perp)} (\phi - \bar{\phi}) &= \rho - \rho_0.
\end{align*}
\]

In the following simulation, we consider a cylinder domain

\[
\Omega = \{(x, y, z) \in \mathbb{R}^3 : \sqrt{x^2 + y^2} \leq R, 0 \leq z \leq L_z\}.
\]

Boundary condition:

- \( \phi(\mathbf{x}) = 0 \) on \( \partial D \times [0, L_z] \), where \( \partial D = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = R\} \).
- Periodic boundary condition in \( z \)-direction.
Discretization of Drift-Kinetic Model

- The Drift-Kinetic Vlasov equation can be split into three equations

\[ \begin{align*}
\frac{\partial f}{\partial t} + \mathbf{U} \cdot \nabla_{x_\perp} f &= 0, \\
\frac{\partial f}{\partial t} + \mathbf{v}_\parallel \partial_z f &= 0, \\
\frac{\partial f}{\partial t} + \mathbf{E}_\parallel \partial v_\parallel f &= 0.
\end{align*} \]

The Strang splitting method can be used for time discretization.

- Averaging the the quasi-neutrality equation in $z$-direction, we get a 2D average equation ($\bar{\phi} = \int_0^{L_z} \phi dz$):

\[-\nabla_\perp \cdot \left( \frac{\rho_0(x_\perp)}{B} \nabla_\perp \bar{\phi} \right) = \bar{\rho} - \rho_0.\]

Taking difference between the quasi-neutrality equation and the average equation, it yields a fluctuation equation ($\phi' = \phi - \bar{\phi}$):

\[-\nabla_\perp \cdot \left( \frac{\rho_0(x_\perp)}{B} \nabla_\perp \phi' \right) + \frac{\rho_0(x_\perp)}{T_e(x_\perp)} \phi' = \rho' - \bar{\rho}.\]

The fluctuation equation can be solve slice by slice in $z$-direction.
Ion Turbulence Simulation

The plasma is initialized by exciting a single or random ion temperature gradient (ITG) model \((m, n)\) (where \(m\) is a poloidal mode and \(n\) is a toroidal mode) :

\[
f = f_{eq} + \delta f.
\]

The equilibrium part \(f_{eq}\) is chosen as a local Maxwellian

\[
f_{eq}(r, v_\parallel) = \frac{n_0(r)}{(2\pi T_i(r))^{1/2}} \exp \left( -\frac{v_\parallel^2}{2T_i(r)} \right),
\]

while the perturbation \(\delta f\) is determined as

\[
\delta f = f_{eq}g(r)h(v_\parallel)\delta p(z, \theta),
\]

where

\[
g(r = 0) \sim g(r = r_{max}) \sim 0,
\]

\[
h(v_\parallel = v_\parallel \text{min}) \sim h(v_\parallel = v_\parallel \text{max}) \sim 0,
\]

\[
\delta p(z, \theta) = \varepsilon \cos \left( \frac{2\pi n}{L_z} z + m\theta \right), \quad \delta p(z, \theta) = \sum_{n, m} \varepsilon_{n, m} \cos \left( \frac{2\pi n}{L_z} z + m\theta + \phi_{n, m} \right).
\]
Evolution of ion turbulence simulation (single mode, cfl \approx 5)
Comparison of different interpolation methods (random mode)

Cubic Hermite

HWENO2
Conclusion and Perspectives

Conclusion

- Classical semi-Lagrangian method for Vlasov equation
  - Parabolic assumption method for search characteristic
  - WENO type method for interpolation
  - HWENO2 interpolation method: quasi non-oscillating, high accuracy, low communication

- Plasma turbulence simulation based on Cartesian mesh
  - Extrapolation technique for Dirichlet B.C. of elliptic equation
  - 2D Diocotron instability simulation
  - 4D ion turbulence simulation
  - Advantages:
    - Non singularity as in polar coordinates
    - Easier to adapt arbitrary domain

Perspectives

- Prove the properties of semi-Lagrangian method for Vlasov equation
- Improve conservation laws of semi-Lagrangian method for Vlasov equation