Numerical methods for reduced Vlasov equation

P. Helluy, L. Navoret, N. Pham

Inria Tonus & IRMA University of Strasbourg

NUMKIN 2016, IRMA, Octobre 20th 2016
Plan

1. Model reduction
2. Numerical methods and objectives
3. Numerical results
4. Perspectives
Vlasov equation:

\[
\frac{\partial f}{\partial t}(x, v, t) + v \cdot \nabla_x f(x, v, t) + E \cdot \nabla_v f(x, v, t) = 0. \tag{1}
\]

- \( f(x, v, t) \): distribution function at time \( t \geq 0 \), with position \( x = (x_1, x_2, x_3) \in \Omega_x \subset \mathbb{R}^3 \) and velocity \( v \in \Omega_v \subset \mathbb{R}^3 \)
- \( E(x, t) \): electric field which satisfies the Poisson equation

\[ E = -\nabla_x \Phi, \quad \text{with} \quad -\Delta_x \Phi = \rho - 1. \tag{2} \]

- the charge density

\[ \rho(x, t) = \int_{\Omega_v} f(x, v, t) \, dv, \tag{3} \]
In practice $\Omega_x = [0, L]^3$, with $L > 0$ and $\Omega_v = [-V_{\text{max}}, V_{\text{max}}]^3$, where $V_{\text{max}} > 0$ is the maximal velocity.

Boundary condition

- Periodical boundary condition in space:

- Outflow boundary condition in velocity:

$$\left( \mathbf{E}(\mathbf{x}, t) \cdot n_v \right)^- f(\mathbf{x}, v, t) = 0$$

where $n_v$ be the outward normal unit vector on $\partial \Omega_v$. 
Reduced Vlasov equation

Weak formulation: Find \( f \in V = \{ f \in L^2(\Omega_v), E \cdot \nabla_v f \in L^2(\Omega_v) \} \)

\[
\partial_t \int_{\Omega_v} f \varphi + \nabla_x \cdot \int_{\Omega_v} v f \varphi + E \cdot \int_{\Omega_v} \nabla_v f \varphi - \beta \int_{\partial \Omega_v} (E \cdot n_v)^- f \varphi = 0, \quad (4)
\]

with any test function \( \varphi \in V \) and where \( \beta \) is a positive constant.

The distribution function is approximated by

\[
f(x, v, t) \approx \sum_{j=1}^{N_v} w_j(x, t) \varphi_j(v). \quad (5)
\]

We obtain the reduced Vlasov equation

\[
M \partial_t w + A^k \partial_{x_k} w + B(E)w = 0, \quad k = 1 \ldots 3. \quad (6)
\]
Reduced Vlasov equation

\[ \mathcal{M}\partial_t w + A^k \partial_{x_k} w + B(E)w = 0, \quad k = 1 \ldots 3. \]

In which \( w = (w_1, w_2, ..., w_{N_v})^T \). The mass matrix \( \mathcal{M} \), \( A^k, k = 1 \ldots 3 \) and \( B(E) \) are matrices of dimension \( N_v \times N_v \), whose elements are given by

\[
\mathcal{M}_{ij} = \int_{\Omega_v} \varphi_i \varphi_j, \quad A^k_{ij} = \int_{\Omega_v} v_k \varphi_i \varphi_j, \quad k = 1 \ldots 3, \quad (7)
\]

\[
B(E)_{ij} = \int_{\Omega_v} \varphi_i (E \cdot \nabla_v) \varphi_j - \beta \int_{\partial \Omega_v} (E \cdot n_v)^- \varphi_j \varphi_i. \quad (8)
\]
Reduced Vlasov equation

Properties

- The system depends only on $x$ variable and not on the $(x, v)$ variables
- $\mathcal{M}$ is symmetric positive-definite, $A^k$ is symmetric, $k = 1, 2, 3$, $B(E)$ is “almost” skew-symmetric (except on boundaries)
- Hyperbolic system
- For $1/2 \leq \beta \leq 1$, the reduced Vlasov equation is $L^2$ stable

$$
\frac{1}{2} \frac{d}{dt} \left( \int_{\Omega_x \times \Omega_v} f^2 \right) \leq -\frac{1}{2} \int_{\Omega_x \times \partial \Omega_v} (E \cdot n_v)^+ f^2, \quad (9)
$$

- Loss of total charge

$$
\frac{d}{dt} \rho_{tot} = - \int_{\Omega_x \times \partial \Omega_v} (E \cdot n_v)^+ f, \quad (10)
$$
Numerical methods and objectives

Resolution of the hyperbolic system:

- Finite volume (SeLaLib)
- Semi-Lagrangian (SeLaLib)
- Discontinuous Galerkin (CLAC, SCHNAPS)

Objectives:

- Implementation of the three methods
- Comparison between the three methods
Finite volume scheme (2D)

Reduced Vlasov equation in 2D

\[ \partial_t w + \mathcal{M}^{-1} A^1 \partial_{x_1} w + \mathcal{M}^{-1} A^2 \partial_{x_2} w + \mathcal{M}^{-1} B(E) w = 0 \]

- Finite-volume approximation

\[ \partial_t w_{kl} = - \frac{\mathcal{F}(w_{k,l}, w_{k+1,l}, \nu^1) - \mathcal{F}(w_{k-1,l}, w_{k,l}, \nu^1)}{h_1} \]
\[ - \frac{\mathcal{F}(w_{k,l}, w_{k,l+1}, \nu^2) - \mathcal{F}(w_{k,l-1}, w_{k,l}, \nu^2)}{h_2} + S(w_{kl}) \]

\[ \nu^1 = (1, 0) \text{ and } \nu^2 = (0, 1) \]

- Viscous flux

\[ \mathcal{F}(w_L, w_R, \nu) = \mathcal{M}^{-1} A_i \nu_i \frac{w_L + w_R}{2} - \kappa \frac{(w_R - w_L)}{2}, \]

with \( \kappa \geq 0 \).

(11)

where \( \kappa \) : numerical diffusion.
Finite volume scheme (2D) - Stability

Time discretization : Runge-Kutta (RK) of order 1, 2, 3 ou 4.

1. For the centered flux
   - RK1 and RK2 schemes are unstable
   - RK3 and RK4 schemes : there exists a constant $\varphi \in (0, 1)$ such that the schemes are stable under the CFL condition $\Delta t = \varphi \frac{\Delta x}{V_{\text{max}}}$. 

2. For the viscous flux : The Runge-Kutta schemes of order 1, 2, 3, 4 with the viscous flux are all stable when taking $\Delta t$ sufficiently small (CFL condition).
   - RK1 and RK2 : CFL parameter depends on $\kappa$
   - RK3 and RK4 schemes : CFL parameter does not depend on $\kappa$
Finite volume scheme (2D) - Implementation

Implementation
- SeLaLib library
- MPI parallelization

Drawbacks
- Matrix inversion (skyline storage of band matrices)
- Stability under CFL condition (in space and velocity)
- Low spatial accuracy

Advantages
- Locally conservative (mass, momentum) and $L^2$ stable
- Finite-element refinement in the velocity direction
- Numerical dissipation only on the spatial direction
Semi-Lagrangian scheme (2D)

Lagrange polynomials and Gauss-Lobatto points $\Rightarrow$ diagonal matrices

$$M^{-1} A^i = \begin{pmatrix} v_1^i \\ \cdot \cdot \cdot \\ v_{N_v}^i \end{pmatrix} \Rightarrow \text{Semi-Lagrangian approximation}$$

Splitting method

- Advection in the $x_1$ direction (1D Semi-Lagrangian)
- Advection in the $x_2$ direction (1D Semi-Lagrangian)
- Addition of the source term

Advantages:

- No CFL condition (in space)
- High order method

Drawbacks:

- Mass conservative on uniform 1D mesh
- Difficult to extend to non-cartesian mesh

Implementation:

- Selalib library
- MPI parallelization
In a cylinder consider a strong homogeneous magnetic field $B_{\text{app}} = b e_3$, the Vlasov equation

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + (E + v \times B_{\text{app}}) \cdot \nabla_v f = 0.$$ 

becomes the drift-kinetic equation

$$\partial_t f + E_\perp \cdot \nabla_{x\perp} f + v_\parallel \partial_{x\parallel} f + E_\parallel \partial_{v\parallel} f = 0. \quad (12)$$

Weak formulation (in velocity) of drift-kinetic equation

$$\mathcal{M} \partial_t w + E_\perp \cdot \mathcal{M} \nabla_{\perp} w + A \partial_{x\parallel} w + B(E_{x\parallel}) w = 0. \quad (13)$$
Weak upwind DG formulation

\[
\int_L \partial_t \mathbf{w}^L \cdot \psi_L - \int_L E_\perp \mathbf{w}^L \cdot \nabla_\perp \psi_L + \int_{\partial L \cap \Omega_x} E_\perp (n^+ \mathbf{w}^L + n^- \mathbf{w}_R) \cdot \psi_L + \\
\int_L \mathbf{w}^L \cdot (\mathcal{M}^{-1} A) \partial_3 \psi_L + \int_{\partial L \cap \Omega_x} ((\mathcal{M}^{-1} A)n_3^+ \mathbf{w}^L + (\mathcal{M}^{-1} A)n_3^- \mathbf{w}_R) \cdot \psi_L + \\
\int_L (\mathcal{M}^{-1} B(E_{X3})) \mathbf{w}^L \cdot \psi_L = 0
\]

(14)

where \( n \) the normal vector on \( \partial L \) oriented from the cell \( L \) to the neighboring cells \( R \), \( \psi_{L,k} \) : test function.
Discontinuous Galerkin scheme

Implementation:
- SCHNAPS code
- multipatch, curved hexahedron mesh (gmsh)
- GPU parallelization (OpenCL)
- Starpu (task distribution)

Drawbacks:
- CFL condition (in velocity and space)

Advantages:
- high-order accuracy
- handle complex geometry
- parallelization
| 1 | Model reduction |
| 2 | Numerical methods and objectives |
| 3 | Numerical results |
| 4 | Perspectives |
2D Landau damping

Initial distribution function: \( f_0(x, v) = \frac{1}{2\pi} (1 + \varepsilon \cos(k_1x_1) \cos(k_2x_2)) e^{-\frac{(v_1^2 + v_2^2)}{2}} \).

Parameters: \( k_1 = k_2 = 0.5, \varepsilon = 5 \times 10^{-3} \); \( x \in [0, 4\pi] \times [0, 4\pi] \); \( v \in [-6, 6] \times [-6, 6] \); \( N_1 = N_2 = 32 \) (space), \( NC_1 = NC_2 = 32 \), degree 2 (velocity); 4 processors.

<table>
<thead>
<tr>
<th></th>
<th>( \Delta t )</th>
<th>cpu time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FV</td>
<td>1.47 \times 10^{-2}</td>
<td>10182</td>
</tr>
<tr>
<td>SL</td>
<td>0.1</td>
<td>748</td>
</tr>
<tr>
<td>SL</td>
<td>0.5</td>
<td>171</td>
</tr>
</tbody>
</table>
Speed-up

2D Landau damping: Computed on the supercomputer Curie (TGCC).
Parameters: 100 iterations, $256^2 \times 65^2 \approx 277.10^6$ unknowns

<table>
<thead>
<tr>
<th>Number of CPU</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>computation time</td>
<td>21280</td>
<td>13992</td>
<td>7237</td>
<td>3685</td>
<td>1869</td>
<td>985</td>
</tr>
<tr>
<td>speed-up (relative)</td>
<td>1.52</td>
<td>1.93</td>
<td>1.96</td>
<td>1.97</td>
<td>1.89</td>
<td></td>
</tr>
<tr>
<td>speed-up</td>
<td>6.08</td>
<td>11.76</td>
<td>23.1</td>
<td>45.53</td>
<td>86.41</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Computational time and speedup for the 2D **finite-volume** code.

<table>
<thead>
<tr>
<th>Number of processor units (CPU)</th>
<th>8</th>
<th>16</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>computation time (second)</td>
<td>7398</td>
<td>3673</td>
<td>943</td>
<td>491</td>
</tr>
<tr>
<td>speed-up (relative)</td>
<td>2.01</td>
<td>3.89</td>
<td>1.92</td>
<td></td>
</tr>
<tr>
<td>speed-up</td>
<td>16.11</td>
<td>62.76</td>
<td>120.54</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Computational time and speedup for the 2D **semi-Lagrangian** code.
The efficiency defined by

\[ \text{eff} := \frac{s \times \text{nb}_{\text{points}} \times \text{nb}_{\text{iterations}}}{T \times 10^6 \times \text{nb}_{\text{processors}}}. \]  

- With the finite volume code, \( \text{eff} = 1.75 \)
- With the semi-Lagrangian code, \( \text{eff} = 3.5 \).

→ The efficiency of semi-Lagrangian method is twice as much than this of the finite volumes method. Reason: semi-Lagrangian is without matrices inversions.
Guiding center model

- Guiding center model in a disc

\[ \partial_t \rho + \mathbf{E}^\perp \cdot \nabla_x \rho = 0 \]
\[ - \Delta_x \Phi = \rho - \bar{\rho}, \quad \mathbf{E} = -\nabla_x \Phi \]

- Diocotron instability test-case [Davidson, 2013]: perturbed annular electron layer
Speed-up

Computational time and speedup for the 2D schnaps code on CPU. Computed on the Irma Atlas cluster. Parameters: 20 iterations, $M_x = 20$, $d_x = 3$, $N_v = 31$, $d_v = 3$, $M_v = 20$.

<table>
<thead>
<tr>
<th>Nb of CPU</th>
<th>comp. time</th>
<th>speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2401.3</td>
<td>1.72</td>
</tr>
<tr>
<td>2</td>
<td>1396.12</td>
<td>2.997</td>
</tr>
<tr>
<td>4</td>
<td>801.39</td>
<td>4.759</td>
</tr>
<tr>
<td>8</td>
<td>504.6</td>
<td>8.67</td>
</tr>
<tr>
<td>16</td>
<td>276.89</td>
<td>12.38</td>
</tr>
<tr>
<td>32</td>
<td>193.9</td>
<td>16.9</td>
</tr>
<tr>
<td>64</td>
<td>142.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sequential code</th>
<th>OpenCL code 1 core</th>
<th>OpenCL code 1GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 iterations</td>
<td>3532</td>
<td>2401.3</td>
</tr>
<tr>
<td>1 iterations</td>
<td>168.19</td>
<td>114.34</td>
</tr>
<tr>
<td>relative speed-up</td>
<td>1.47</td>
<td>63.5</td>
</tr>
</tbody>
</table>
1. Apply the code in schnaps to solve the drift-kinetic model and the gyro-kinetic model.

2. Resolve the drift-kinetic in SELALIB to compare with the one in schnaps.

3. Study the Vlasov model with the collision term.
THANKS FOR YOUR ATTENTION!