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

Model reduction of the Vlasov-Poisson equation

Vlasov equation :

$$\frac{\partial f}{\partial t}(\mathbf{x},\mathbf{v},t) + \mathbf{v}\cdot\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{v},t) + \mathbf{E}\cdot\nabla_{\mathbf{v}} f(\mathbf{x},\mathbf{v},t) = 0. \tag{1}$$

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

$$\mathbf{E} = -\nabla_{\mathbf{x}}\Phi, \quad \text{with} \quad -\triangle_{\mathbf{x}}\Phi = \rho - 1. \tag{2}$$

- the charge density

$$\rho(\mathbf{x},t) = \int_{\Omega_{\mathbf{v}}} f(\mathbf{x},\mathbf{v},t)\, d\mathbf{v}, \tag{3}$$

Boundary condition

In practice $\Omega_{\mathbf{x}} =]0, L[^3$, with $L > 0$ and $\Omega_{\mathbf{v}} =]-V_{\max}, V_{\max}[^3$, where $V_{\max} > 0$ is the maximal velocity

Boundary condition

- Periodical boundary condition in space :
- Outflow boundary condition in velocity :

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

where $n_{\mathbf{v}}$ be the outward normal unit vector on $\partial\Omega_{\mathbf{v}}$

Reduced Vlasov equation

Weak formulation : Find $f \in V = \left\{ f \in L^2(\Omega_{\mathbf{v}}), \mathbf{E} \cdot \nabla_{\mathbf{v}} f \in L^2(\Omega_{\mathbf{v}}) \right\}$

$$\partial_t \int_{\Omega_{\mathbf{v}}} f\varphi + \nabla_{\mathbf{x}} \cdot \int_{\Omega_{\mathbf{v}}} \mathbf{v} f\varphi + \mathbf{E} \cdot \int_{\Omega_{\mathbf{v}}} \nabla_{\mathbf{v}} f\varphi - \beta \int_{\partial\Omega_{\mathbf{v}}} (\mathbf{E} \cdot n_{\mathbf{v}})^- f\varphi = 0, \quad (4)$$

with any test function $\varphi \in V$ and where β is a positive constant.
The distribution function is approximated by

$$f(\mathbf{x}, \mathbf{v}, t) \approx \sum_{j=1}^{N_{\mathbf{v}}} w_j(\mathbf{x}, t)\, \varphi_j(\mathbf{v}). \quad (5)$$

We obtain the reduced Vlasov equation

$$\mathcal{M}\partial_t \mathbf{w} + A^k \partial_{x_k} \mathbf{w} + B(\mathbf{E})\mathbf{w} = 0, \quad k = 1 \dots 3. \quad (6)$$

Reduced Vlasov equation

Reduced Vlasov equation

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

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

$$\mathcal{M}_{ij} = \int_{\Omega_\mathbf{v}} \varphi_i \varphi_j, \quad A^k_{ij} = \int_{\Omega_\mathbf{v}} v_k \, \varphi_i \varphi_j, \quad k = 1 \dots 3, \tag{7}$$

$$B(\mathbf{E})_{ij} = \int_{\Omega_\mathbf{v}} \varphi_i (\mathbf{E} \cdot \nabla_\mathbf{v}) \varphi_j - \beta \int_{\partial\Omega_\mathbf{v}} (\mathbf{E} \cdot n_\mathbf{v})^- \varphi_j \varphi_i. \tag{8}$$

Reduced Vlasov equation

Properties

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

$$\frac{1}{2}\frac{d}{dt}\left(\int_{\Omega_x \times \Omega_{\mathbf{v}}} f^2\right) \leqslant -\frac{1}{2}\int_{\Omega_x \times \partial\Omega_{\mathbf{v}}} (\mathbf{E} \cdot n_{\mathbf{v}})^+ f^2, \tag{9}$$

- Loss of total charge

$$\frac{d}{dt}\rho_{\mathtt{tot}} = -\int_{\Omega_{\mathbf{x}} \times \partial\Omega_{\mathbf{v}}} (\mathbf{E} \cdot n_{\mathbf{v}})^+ f, \tag{10}$$

Plan

Numerical methods and objectives

Resolution of the hyperbolic system :

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

Objectives :

- Implementation of the three methods
- Comparison between the three methods

Finite volume scheme (2D)

Reduced Vlasov equation in 2D

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

- Finite-volume approximation

$$\partial_t \mathbf{w}_{kl} = -\frac{\mathcal{F}(\mathbf{w}_{k,l}, \mathbf{w}_{k+1,l}, \nu^1) - \mathcal{F}(\mathbf{w}_{k-1,l}, \mathbf{w}_{k,l}, \nu^1)}{h_1} - \frac{\mathcal{F}(\mathbf{w}_{k,l}, \mathbf{w}_{k,l+1}, \nu^2) - \mathcal{F}(\mathbf{w}_{k,l-1}, \mathbf{w}_{k,l}, \nu^2)}{h_2} + S(\mathbf{w}_{kl})$$

  $\nu^1 = (1,0)$ and $\nu^2 = (0,1)$

- Viscous flux

$$\mathcal{F}(\mathbf{w}_L, \mathbf{w}_R, \nu) = \mathcal{M}^{-1} A_i \nu_i \frac{\mathbf{w}_L + \mathbf{w}_R}{2} - \kappa \frac{(\mathbf{w}_R - \mathbf{w}_L)}{2}, \qquad \text{with } \kappa \geq 0. \tag{11}$$

  where κ : 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 $\wp \in (0,1)$ such that the schemes are stable under the CFL condition $\Delta t = \wp \frac{\Delta x}{V_{\text{max}}}$.
2. For the viscous flux : The Runge-Kutta schemes of order $1, 2, 3, 4$ with the **viscous** flux are all stable when taking Δt sufficiently small (CFL condition).
   - RK1 and RK2 : CFL parameter depends on κ
   - RK3 and RK4 schemes : CFL parameter does not depend on κ

Finite volume scheme (2D) - Implementation

Implementation

- SeLaLib libraby
- 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

$$M^{-1}A^i = \begin{pmatrix} v_1^i & & \\ & \ddots & \\ & & v_{N_{\mathbf{v}}}^i \end{pmatrix} \Rightarrow \text{Semi-Lagrangian approximation}$$

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

Implementation :

- Selalib library
- MPI parallelization

Advantages :

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

Drawbacks :

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

Discontinuous Galerkin scheme - Drift-kinetic model

In a cylinder consider a strong homogeneous magnetic field $\mathbf{B}_{\text{app}} = b\,\mathbf{e}_3$, the Vlasov equation

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} f + (\mathbf{E} + \mathbf{v}\times\mathbf{B}_{\text{app}})\cdot\nabla_{\mathbf{v}} f = 0.$$

becomes the drift-kinetic equation

$$\partial_t f + \mathbf{E}_\perp^\perp\cdot\nabla_{\mathbf{x}_\perp} f + v_{||}\partial_{x_{||}} f + E_{||}\partial_{v_{||}} f = 0. \tag{12}$$

Weak formulation (in velocity) of drift-kinetic equation

$$\mathcal{M}\partial_t \mathbf{w} + \mathbf{E}_\perp\cdot\mathcal{M}\nabla_\perp \mathbf{w} + A\,\partial_{x_{||}}\mathbf{w} + B(E_{x_{||}})\,\mathbf{w} = 0. \tag{13}$$

Discontinuous Galerkin scheme - Drift-kinetic model

Weak upwind DG formulation

$$\int_L \partial_t \mathbf{w}^L \cdot \psi_L - \int_L E_\perp \mathbf{w}^L \cdot \nabla_\perp \psi^L + \int_{\partial L \cap \Omega_{\mathbf{x}}} E_\perp \left(n^+ \mathbf{w}_L + n^- \mathbf{w}_R\right) \cdot \psi_L +$$
$$\int_L \mathbf{w}^L \cdot (\mathcal{M}^{-1} A) \partial_3 \psi^L + \int_{\partial L \cap \Omega_{\mathbf{x}}} ((\mathcal{M}^{-1} A) n_3^+ \mathbf{w}_L + (\mathcal{M}^{-1} A) n_3^- \mathbf{w}_R) \cdot \psi_L +$$
$$\int_L (\mathcal{M}^{-1} B(E_{X_3})) \mathbf{w}^L \cdot \psi^L = 0 \tag{14}$$

where n the normal vector on ∂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)

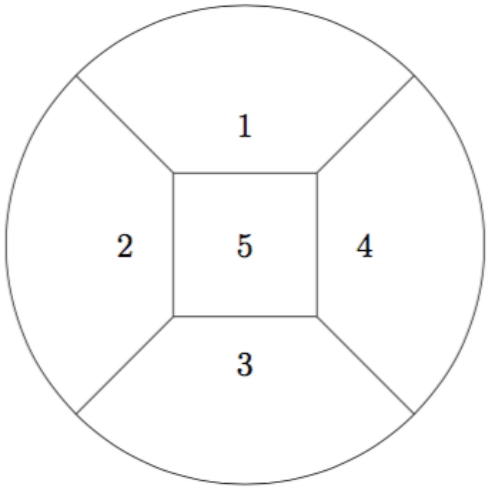


FIGURE: Macro-mesh of a simple disk

Advantages :

- high-order accuracy
- handle complex geometry
- parallelization

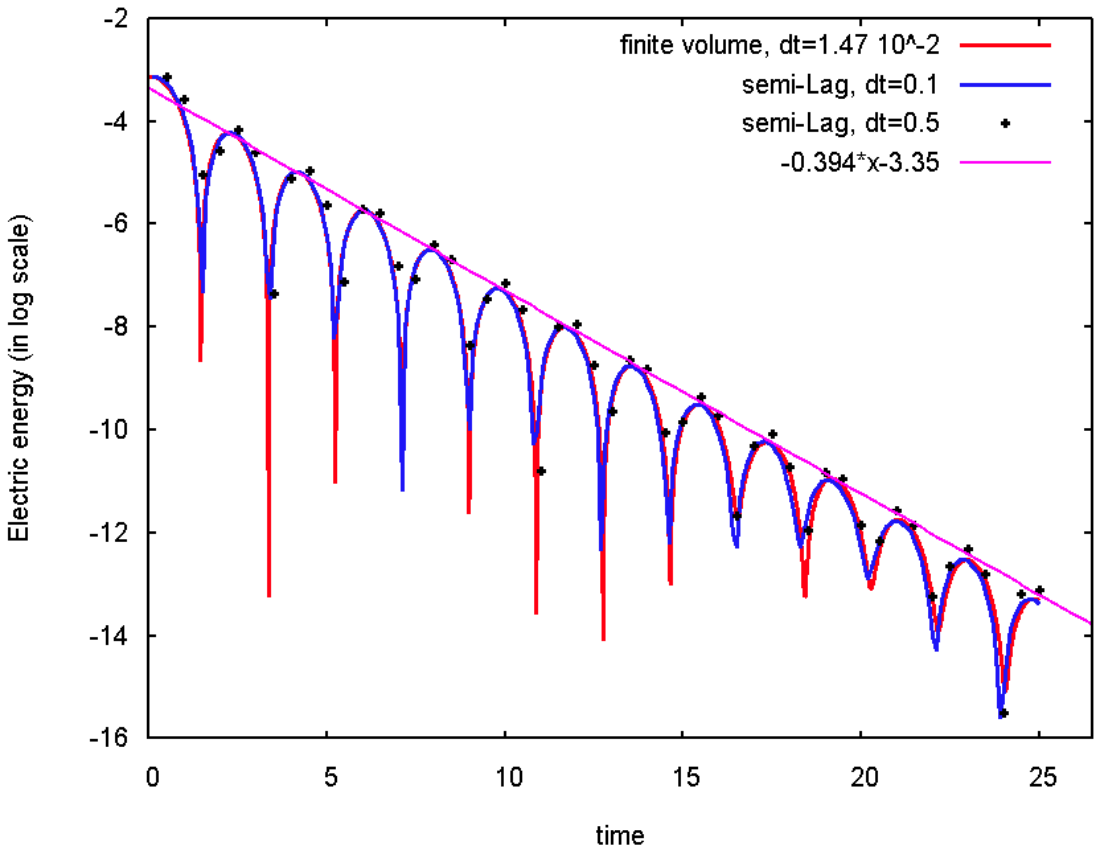
Plan

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 : $k_1 = k_2 = 0.5$, $\varepsilon = 5\times10^{-3}$; $x \in [0,4\pi]\times[0,4\pi]$, $v \in]-6,6[\times]-6,6[$; $N_1 = N_2 = 32$ (space), $NC_1 = NC_2 = 32$, degree 2 (velocity) ; 4 processors.

	Δt	cpu time (s)
FV	1.47 10^{-2}	10182
SL	0.1	748
SL	0.5	171



Speed-up

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

Number of CPU	4	8	16	32	64	128
computation time	21280	13992	7237	3685	1869	985
speed-up (relative)		1.52	1.93	1.96	1.97	1.89
speed-up		6.08	11.76	23.1	45.53	86.41

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

Number of processor units (CPU)	8	16	64	128
computation time (second)	7398	3673	943	491
speed-up (relative)		2.01	3.89	1.92
speed-up		16.11	62.76	120.54

TABLE: Computational time and speedup for the 2D **semi-Lagrangian** code.

Efficiency of code

The efficiency defined by

$$\text{eff} := \frac{s * nb_{points} * nb_{iterations}}{T * 10^6 * nb_{\text{processors}}}. \tag{15}$$

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

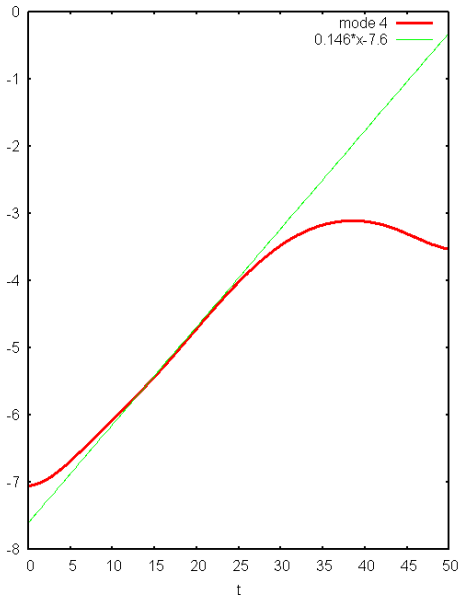
$\rightarrow$ The effeciency of semi-Lagrangian method is twice as much than this of the finite volumes method. Reason : semi-Lagrangian is without matrices inversions.

Guiding center model

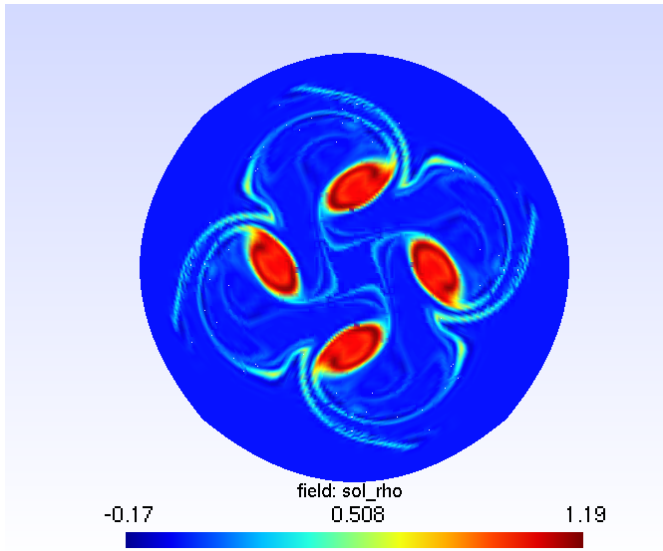
- Guiding center model in a disc

$$\partial_t \rho + \mathbf{E}^\perp \cdot \nabla_{\mathbf{x}} \rho = 0$$
$$-\Delta_{\mathbf{x}} \Phi = \rho - \bar{\rho}, \quad \mathbf{E} = -\nabla_{\mathbf{x}} \Phi$$

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



Growth-rate of the 4th Fourier mode



ρ at time $t = 100$

Speed-up

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

	sequential code 1 core	OpenCL code 1 core	OpenCL code 1GPU
20 iterations	3532	2401.3	39.2
1 iterations	168.19	114.34	1.8
relative speed-up		1.47	63.5

Nb of CPU	1	2	4	8	16	32	64
comp. time	2401.3	1396.12	801.39	504.6	276.89	193.9	142.5
speed-up		1.72	2.997	4.759	8.67	12.38	16.9

Plan

Perspectives

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 !