New formulations of the semi-Lagrangian method for Vlasov-type equations

Eric Sonnendrücker

IRMA
Université Louis Pasteur, Strasbourg

projet CALVI
INRIA Nancy Grand Est

17 September 2008

In collaboration with
N. Crouseilles, G. Latu, M. Mehrenberger, T. Respaud

Numerical methods for gyrokinetic model

- Most codes still based on Particle-In-Cell method and its variants.
- More codes now use phase-space grid. Such codes exist in Japan, USA and Europe.
- Most use eulerian methods from fluid dynamics.
- We have contributed to GYSELA code developed by V. Grandgirard which is based on semi-Lagrangian method.

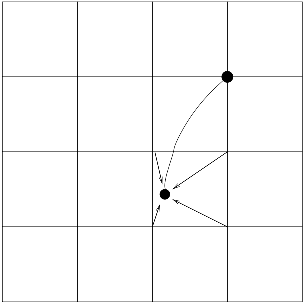
Outline

1. Classical semi-Lagrangian method

2. Conservative semi-Lagrangian method

3. Forward semi-Lagrangian method

The backward semi-Lagrangian Method



- f conserved along characteristics
- Find the origin of the characteristics ending at the grid points
- Interpolate old value at origin of characteristics from known grid values → High order interpolation needed

- Typical interpolation schemes.
  - Cubic spline (Cheng-Knorr)
  - Cubic Hermite with derivative transport (Nakamura-Yabe)

History of semi-Lagrangian schemes for Vlasov

- Cheng-Knorr (JCP 1976): splitting method for 1D Vlasov-Poisson. Cubic spline interpolation.
- ES-Roche-Bertrand-Ghizzo (JCP 1998): general semi-Lagrangian framework for Vlasov-type equations.
- Nakamura-Yabe (CPC 1999): semi-Lagrangian CIP method with Hermite interpolation
- Filbet-ES-Bertrand (JCP 2001) : semi-Lagrangian PFC method: positive and conservative
- N. Besse - ES (JCP 2003) : semi-Lagrangian solver on unstructured grids.
- Crouseilles-Mehrenberger-ES (2008): Equivalence of point based and conservative methods for Vlasov-Poisson + new class of positive filters.
- Crouseilles-Respaud-ES (2008): Forward semi-Lagrangian method.

Convergence of semi-Lagrangian schemes

- Filbet (SINUM 2001): PFC for Vlasov-Poisson
- N. Besse (SINUM 2003): semi-Lagrangian method with linear for Vlasov-Poisson.
- Campos Pinto - Mehrenberger (Numer. Math. 2008): adaptive SL method for Vlasov-Poisson
- N. Besse (SINUM 2008): Convergence of semi-Lagrangian method with cubic Hermite interpolation.
- N. Besse - Mehrenberger (Math of Comp 2008): SL method for different classes of high-order interpolators.

Computation of the origin of the characteristics

- Transport equation

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

- Characteristics

$$\frac{dX}{dt} = \mathbf{a}$$

- Computation of the origin of the characteristics :
  - Explicit solution if $\mathbf{a}$ does not depend on x
  - Else, numerical algorithm needed.

Splitting for exact computation of characteristics

- In many cases splitting can enable to solve a constant coefficient advection at each split step.
- E.g. separable Hamiltonian $H(\mathbf{q},\mathbf{p}) = U(\mathbf{q}) + V(\mathbf{p})$.
  - Vlasov equation in canonical coordinates reads

$$\frac{\partial f}{\partial t} + \nabla_p H \cdot \nabla_q f - \nabla_q H \cdot \nabla_p f = 0.$$

  - Split equations then become

$$\frac{\partial f}{\partial t} + \nabla_p V \cdot \nabla_q f = 0, \quad \frac{\partial f}{\partial t} - \nabla_q U \cdot \nabla_p f = 0,$$

    where U does not depend on $\mathbf{p}$ and V does not depend on $\mathbf{q}$ $\rightarrow$ characteristics can be solved explicitly.
  - Vlasov-Poisson falls into this category with $\mathbf{q} = \mathbf{x}$, $\mathbf{p} = \mathbf{v}$, $H(\mathbf{x},\mathbf{v}) = \frac{1}{2}m\mathbf{v}^2 + q\phi(\mathbf{x},t)$.

First order computation of characteristics

- Consider general case. Characteristics defined by

$$\frac{d\mathbf{X}}{dt} = \mathbf{a}(\mathbf{X}, t).$$

- Backward solution: $\mathbf{X}^n$ is known and $\mathbf{a}^n$ known on the grid.
- Standard procedure to derive first order numerical method for EDO. Integrate on one time step and use quadrature formula for integral (left or right rectangle).

$$\mathbf{X}^{n+1} - \mathbf{X}^n = \Delta t\, \mathbf{a}^n(\mathbf{X}^n) \quad \text{or} \quad \mathbf{X}^{n+1} - \mathbf{X}^n = \Delta t\, \mathbf{a}^{n+1}(\mathbf{X}^{n+1}).$$

- No explicit solution:
  - Fixed point procedure needed in first case (e.g. Newton).
  - Predictor-corrector method on $\mathbf{a}$ needed in second case.

Second order method for Vlasov-Poisson (1/2)

Technique can be used for solving separable Hamiltonian Vlasov equations without splitting. We shall consider Vlasov-Poisson to illustrate this case.

- At time t_n: f^n and E^n are known at grid points. f^{n+1} and E^{n+1} need to be computed.
- Need to solve characteristics backward in time from t_{n+1} to t_n

$$\frac{dV}{dt} = E(X(t), t), \quad \frac{dX}{dt} = V.$$

- Main problem E^{n+1} needed and not known.

Second order method for Vlasov-Poisson (2/2)

A second order in time algorithm predictor-corrector algorithm is defined as follows:

1. Predict $\bar{E}^{n+1}$ using continuity equation.
2. For all grid points $x_i = X^{n+1}, v_j = V^{n+1}$ compute
   - $V^{n+1/2} = V^{n+1} - \frac{\Delta t}{2}\bar{E}^{n+1}(X^{n+1})$,
   - $X^n = X^{n+1} - \Delta t V^{n+1/2}$,
   - $V^n = V^{n+1/2} - \frac{\Delta t}{2}E^n(X^n)$.
   - Interpolate f^n at point (X^n, V^n).
3. Yields first approximation of $f^{n+1}(x_i, v_j) = f^n(X^n, V^n)$ that can be used to correct $\bar{E}^{n+1}$.

Iterate until $\bar{E}^{n+1}$ does not vary anymore.

At most a couple iterations required.

A two step second order method

- Solve characteristics defined by $\frac{d\mathbf{X}}{dt} = \mathbf{a}(\mathbf{X}, t)$.
- Centered quadrature on two time steps:

$$\mathbf{X}^{n+1} - \mathbf{X}^{n-1} = 2\Delta t\, \mathbf{a}^n(\mathbf{X}^n), \quad \mathbf{X}^{n+1} + \mathbf{X}^{n-1} = 2\mathbf{X}^n + O(\Delta t^2).$$

- Use fixed point procedure to compute $\mathbf{X}^{n-1}$ such that

$$\mathbf{X}^{n+1} - \mathbf{X}^{n-1} = \Delta t\, \mathbf{a}^n(\frac{\mathbf{X}^{n+1} + \mathbf{X}^{n-1}}{2}).$$

- Problem: compute f^{n+1} from f^{n-1}. Even and odd order time approximations become decoupled after some time. Artificial coupling needs to be introduced.

A one step predictor-corrector second order method

- Solve characteristics defined by $\frac{d\mathbf{X}}{dt} = \mathbf{a}(\mathbf{X}, t)$.
- Centered quadrature on one time step:

$$\mathbf{X}^{n+1} - \mathbf{X}^{n} = \Delta t\, \mathbf{a}^{n+\frac{1}{2}}(\mathbf{X}^{n+\frac{1}{2}}), \quad \mathbf{X}^{n+1} + \mathbf{X}^{n} = 2\mathbf{X}^{n+\frac{1}{2}} + O(\Delta t^2).$$

- Now $\mathbf{a}^{n+\frac{1}{2}}$ is unknown. Predictor-corrector procedure needed.
- Use fixed point procedure to compute $\mathbf{X}^{n}$ such that

$$\mathbf{X}^{n+1} - \mathbf{X}^{n} = \Delta t\, \bar{\mathbf{a}}^{n+\frac{1}{2}}\left(\frac{\mathbf{X}^{n+1} + \mathbf{X}^{n}}{2}\right).$$

- Both predictor-corrector and fixed point iterations needed.

Outline

1 Classical semi-Lagrangian method

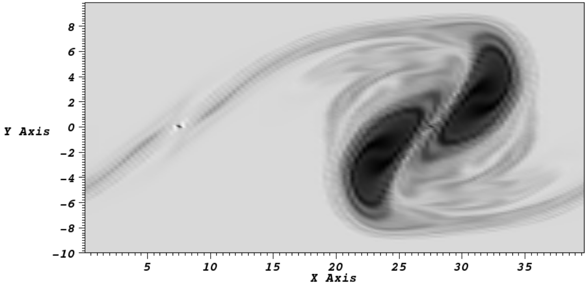
2 Conservative semi-Lagrangian method

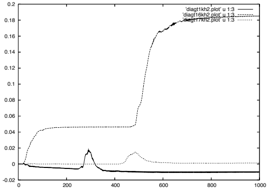
3 Forward semi-Lagrangian method

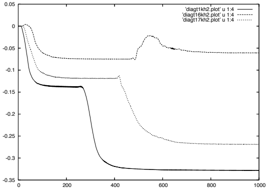
Problem with non conservative Vlasov solver

- When non conservative splitting is used for the numerical solver, the solver is not exactly conservative.
- Does generally not matter when solution is smooth and well resolved by the grid. The solver is still second order and yields good results.
- However: Fine structures develop in non linear simulations and are at some point locally not well resolved by the phase space grid.
- In this case a non conservative solvers can exhibit a large numerical gain or loss of particles which is totally unphysical.
- Lack of robustness.

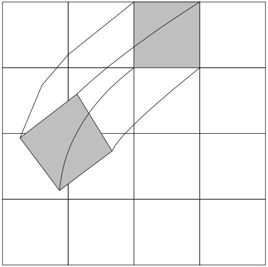
Vortex in Kelvin-Helmholtz instability







Conservative semi-Lagrangian method



- Start from conservative form of Vlasov equation

$$\frac{\partial f}{\partial t} + \nabla \cdot (f\mathbf{a}) = 0.$$

- $\int_V f\, dx\, dv$ conserved along characteristics
- Three steps:
  - High order polynomial reconstruction.
  - Compute origin of cells
  - Project (integrate) on transported cell.
- Efficient with splitting in 1D conservative equations as cells are then defined by their 2 endpoints. A lot more complex for 2D (or more) transport.
- Splitting on conservative form: always conservative.

High order polynomial reconstruction

- We only use the method with 1D splitting with equations in conservative form.
- Unknowns are cell averages: $f_j = \frac{1}{\Delta x} f(x)\,dx$.
- At time step t_n let f_j^n known average value of f^n on cell $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ of length $h_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$.
- Construct polynomial $p_m(x)$ of degree m such that

$$\frac{1}{h_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_m(x)\,dx = f_j^n.$$

- Reconstruction by primitive.

Choice of interpolation

- What interpolation should be chosen for primitive?
- Lagrange interpolation with centered stencil (used in PFC Filbet, ES, Bertrand JCP 2001).
- ENO type interpolation. Lagrange with varying stencil. Not efficient for Vlasov.
- Cubic spline interpolation: cubic polynomial on each cell, globally C^2 $\rightarrow$ reconstructed function is then locally a quadratic polynomial and globally C^1. Linked to cubic spline interpolation for classical semi-Lagrangian method.

Origin of cells and projection

- Compute cell origins:
    - In 1D cell and its origin determined by end points. Compute origin of end points like in classical semi-Lagrangian method.
    - Need to make sure end points do not cross → restriction on time step.
- Compute average value of f^{n+1} on cells using

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f^{n+1}(x)\,dx = \int_{X(t_n;x_{i-\frac{1}{2}},t_{n+1})}^{X(t_n;x_{i+\frac{1}{2}},t_{n+1})} f^n(x)\,dx,$$

where $f^n(x)$ is the high order reconstruction.

Link between classical and conservative semi-Lagrangian methods

- For constant coefficient advections it can be shown that

$$\text{C-Lag(2d)} \Longleftrightarrow \text{SL-Lag(2d+1)}$$
$$\text{PSM} \Longleftrightarrow \text{SPL}$$

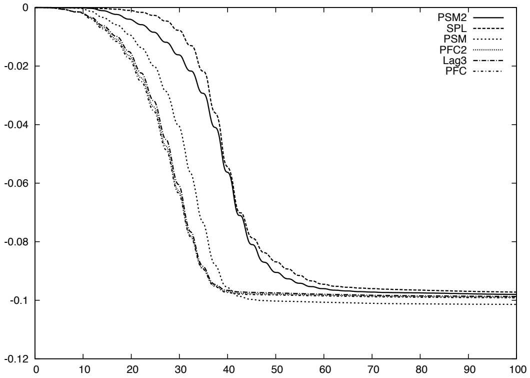
- Consequences :
  1. Classical and conservative semi-Lagrangian methods equivalent for constant coefficients split equations.
  2. The PFC method (Filbet-ES-Bertrand, JCP 2001) corresponds for the Vlasov-Poisson (or Vlasov-Maxwell) systems to a classical semi-Lagrangian method with cubic Lagrange interpolation.

Positive reconstruction

- Physical distribution function is always positive.
- High-order interpolation can lead to negative values in some zones.
- Reconstructed polynomial can be locally modified to remain positive.
- Performed for Lagrange reconstruction in PFC method.
- Introduces a little more dissipativity, but far less as monotonic reconstructions performed in fluid dynamics.

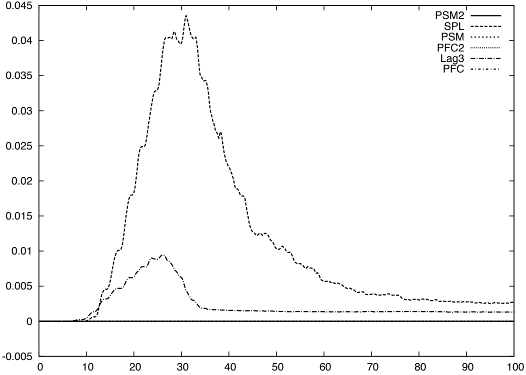
Simulations

Evolution of L^2 norm for $N = 128$ for SLD



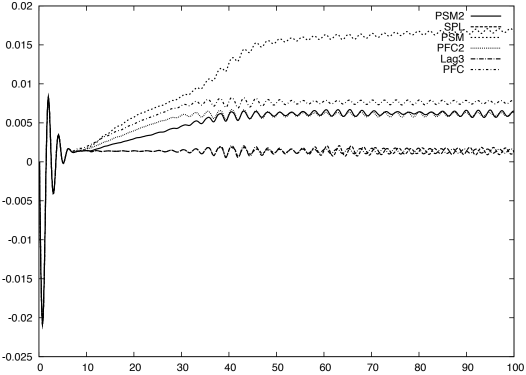
Simulations

Evolution of L^1 norm for $N = 128$ for SLD



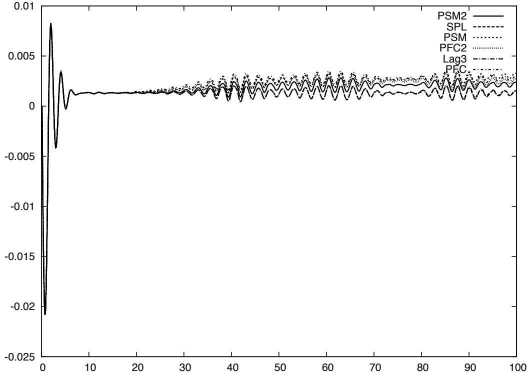
Simulations

Evolution of total energy for $N = 128$ for SLD



Simulations

Evolution of total energy for $N = 512$ for SLD



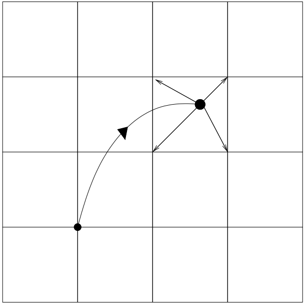
Outline

1. Classical semi-Lagrangian method

2. Conservative semi-Lagrangian method

3. Forward semi-Lagrangian method

The forward semi-Lagrangian method



- f conserved along characteristics
- Characteristics advanced with same time schemes as in PIC method.
- Leap-Frog Vlasov-Poisson
- Runge-Kutta for guiding-center or gyrokinetic

- Values of f deposited on grid of phase space using convolution kernel.
- Identical to PIC deposition scheme but in whole phase space instead of configuration space only.
- Similar to PIC method with reconstruction introduced by Denavit (JCP 1972).

Discrete distribution function

- Function projected on partition of unity basis for conservativity.
- Linear B-splines very diffusive. Not useful in practice.
- Good choice is cubic B-splines.
- f is reprojected on mesh at each time step.
- Between t_n and t_{n+1}

  $$f_h(x,v,t) = \sum_{i,j} w_{i,j} B(x - X(t;x_i,v_j,t_n)) B(v - X(t;x_i,v_j,t_n)).$$

  Weight $w_{i,j}$ associated to the particle starting from grid point (x_i, v_j) at t_n is coefficient of spline satisfying interpolation conditions

  $$f_h(x_k,v_l,t_n) = \sum_{i,j} w_{i,j} B(x_k - x_i) B(v_l - v_j).$$

- Projection on phase space mesh is obtained with formula

$$f^{n+1}(x_k,v_l) = \sum_{i,j} w_{i,j} B(x_k - X(t_{n+1},x_i,v_j,t_n)) B(v_l - V(t_{n+1},x_i,v_j,t_n)).$$

Time advance for Vlasov-Poisson

- As opposite to BSL, trajectories are advanced forward in time.
- Advection field is known at initial time. Standard ODE algorithms can be applied.
- For Vlasov-Poisson, we have $z(t^n) = (x^n, v^n)$, and $E(t^n, z^n) = E(t^n, x^n)$. Separable Hamiltonian.
- Natural scheme is Verlet algorithm, which is second order accurate in time

$$\text{Step1}: v^{n+\frac{1}{2}} - v^n = \frac{\Delta t}{2} E(t^n, x^n),$$

$$\text{Step2}: x^{n+1} - x^n = \Delta t \, v^{n+1/2},$$

$$\text{Step3}: v^{n+1} - v^{n+\frac{1}{2}} = \frac{\Delta t}{2} E(t^{n+1}, x^{n+1}).$$

Time advance for guiding-center

- Explicit Euler and Runge Kutta have been implemented.
- Second-order Runge Kutta method

$$\text{Step1}: X^{n+1} - X^n = \Delta t E^{\perp}(t^n, X^n)$$

$$\text{Step2}: \text{Compute } E^{\perp}(t^{n+1}, X^{n+1})$$

$$\text{Step3}: X^{n+1} - X^n = \frac{\Delta t}{2}\left[E^{\perp}(t^n, X^n) + E^{\perp}(t^{n+1}, X^{n+1})\right]$$

- Fourth order Runge-Kutta

$$\text{Step1}: k_1 = E^{\perp}(t^n, X^n)$$

$$\text{Step2}: \text{Compute } k_2 = E^{\perp}(t^{n+\frac{1}{2}}, X^n + \frac{\Delta t}{2} k_1)$$

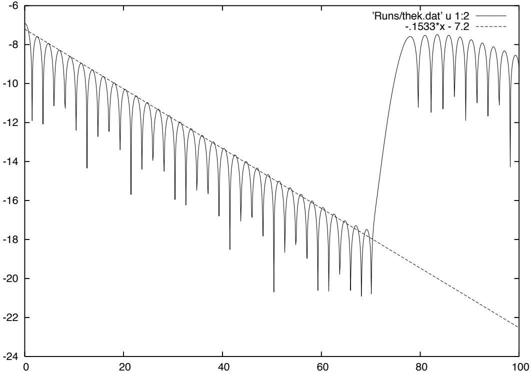
$$\text{Step3}: \text{Compute } k_3 = E^{\perp}(t^{n+\frac{1}{2}}, X^n + \frac{\Delta t}{2} k_2)$$

$$\text{Step4}: \text{Compute } k_4 = E^{\perp}(t^{n+1}, X^n + \Delta t k_3)$$

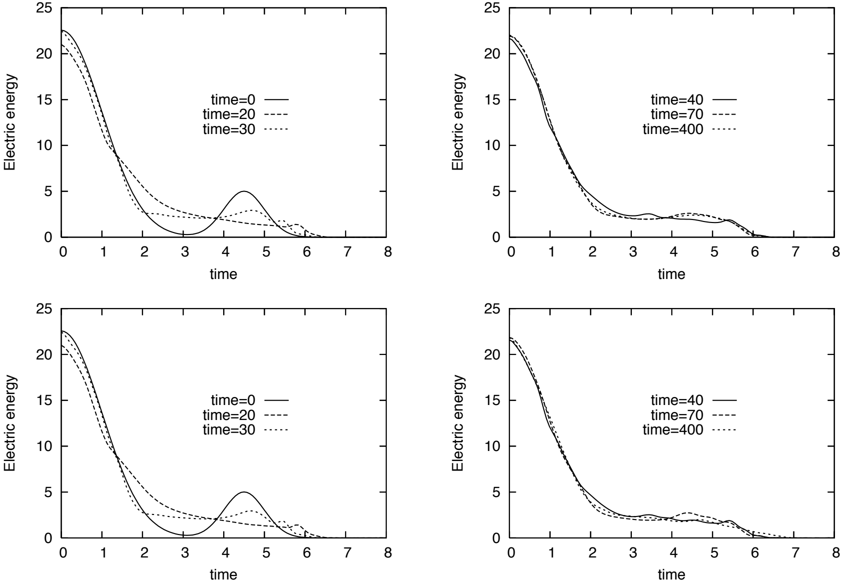
$$\text{Step5}: X^{n+1} - X^n = \frac{\Delta t}{6}[k_1 + 2k_2 + 2k_3 + k_4]$$

Landau damping

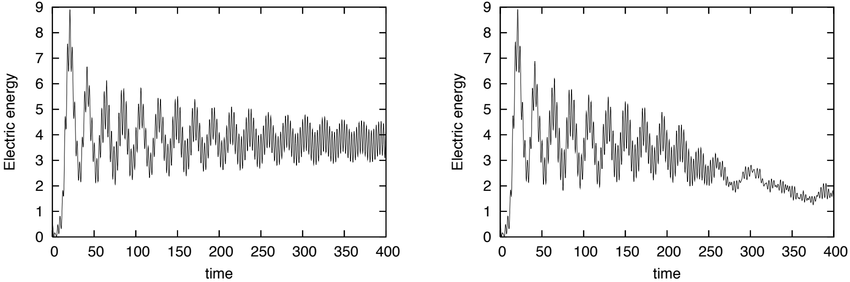
$$f_0(x, v) = (1 + 0.001 \cos(kx)) \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}, \quad L = 4\pi.$$



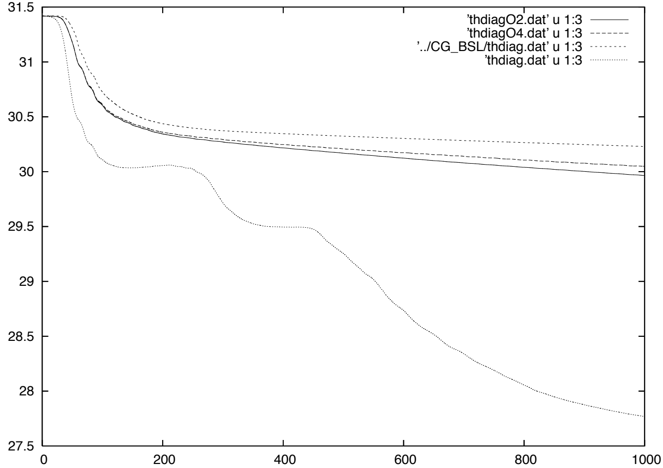
Bump on tail (BSL (top) vs. FSL (bottom))



Bump on tail: potential energy (BSL (left) vs. FSL (right))



Energy conservation for Kelvin-Helmoltz instability



FSL vs. BSL

- Forward semi-Lagrangian method very promising.
- Accurate description of whole phase-space in particular tail of distribution function and small perturbations.
- A little more diffusive than BSL. Better with smaller time steps.
- Some advantages: classical explicit EDO solver can be used, in particular high order if needed. No need for predictor-corrector or fixed point algorithm.
- Can benefit from charge conserving PIC algorithms (Villasenor-Buneman).