

Vlasov simulations on an adaptive phase-space grid

Eric Sonnendrücker
IRMA, Université Louis Pasteur
Strasbourg, France

Berkeley, September 15, 2003

Co-workers:

Michael Gutnic, Ioana Paun (Strasbourg)

Nicolas Besse, Jean-Louis Lemaire (CEA, Bruyères-Le-Châtel)

Edouard Oudet (Strasbourg)

Francis Filbet (Orléans)

Alex Friedman (LLNL)

Jean-Luc Vay (LBNL)

Outline

- Grid based methods for the Vlasov equation.
- Motivations for adaptiveness
- An adaptive method based on a wavelet decomposition.
- A moving phase space grid method.

The Vlasov equation

- Simulation of plasmas and high intensity beams often require to numerically the Vlasov equation coupled with a self-field solver (Poisson, Maxwell...)
- Distribution function $f(x, v, t)$ is solution of the Vlasov equation

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{q}{m} (E + v \times B) \cdot \nabla_v f = 0,$$

generally coupled with the Poisson or Maxwell equations.

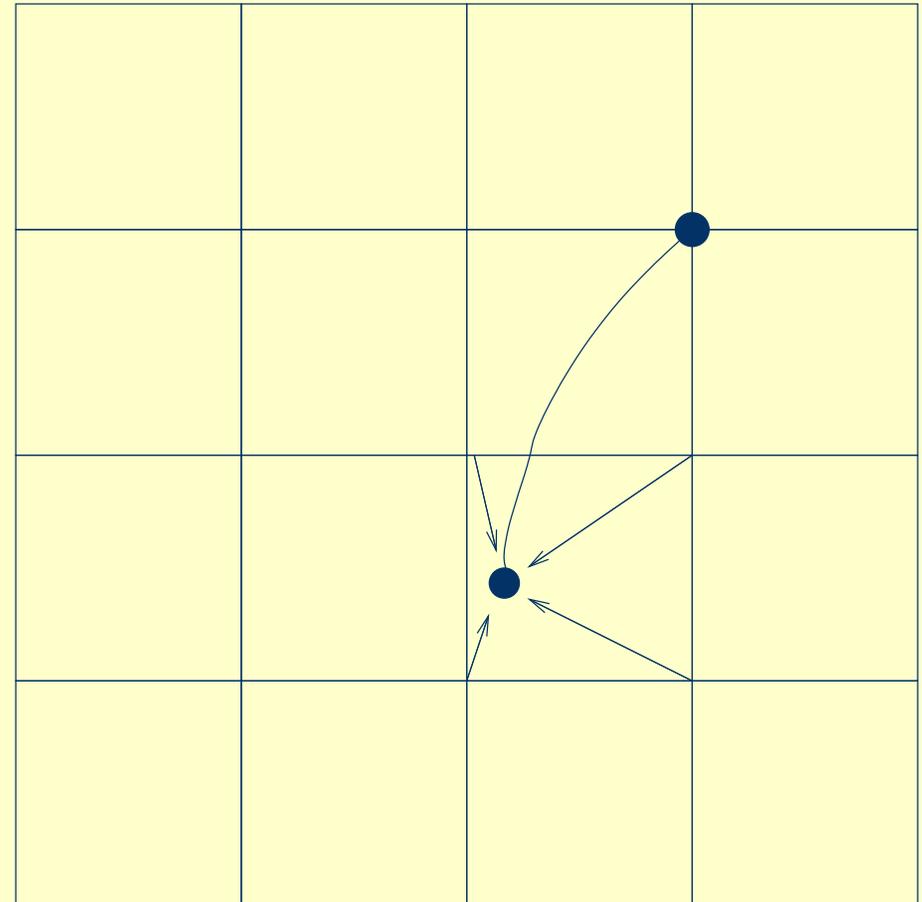
- Numerical simulations are mostly performed using PIC method.

Why use direct Vlasov methods ?

- PIC remains the essential tool for beam and plasma simulations.
- Important noise in PIC methods especially in poorly populated regions of phase space makes it hard to see phenomena like e.g.
 - particle trapping (strong Landau damping) in plasmas
 - halo formation in beams
- Computers now powerful enough to do realistic physics using a grid in phase space.
- Provides alternative to PIC for code benchmarking.

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

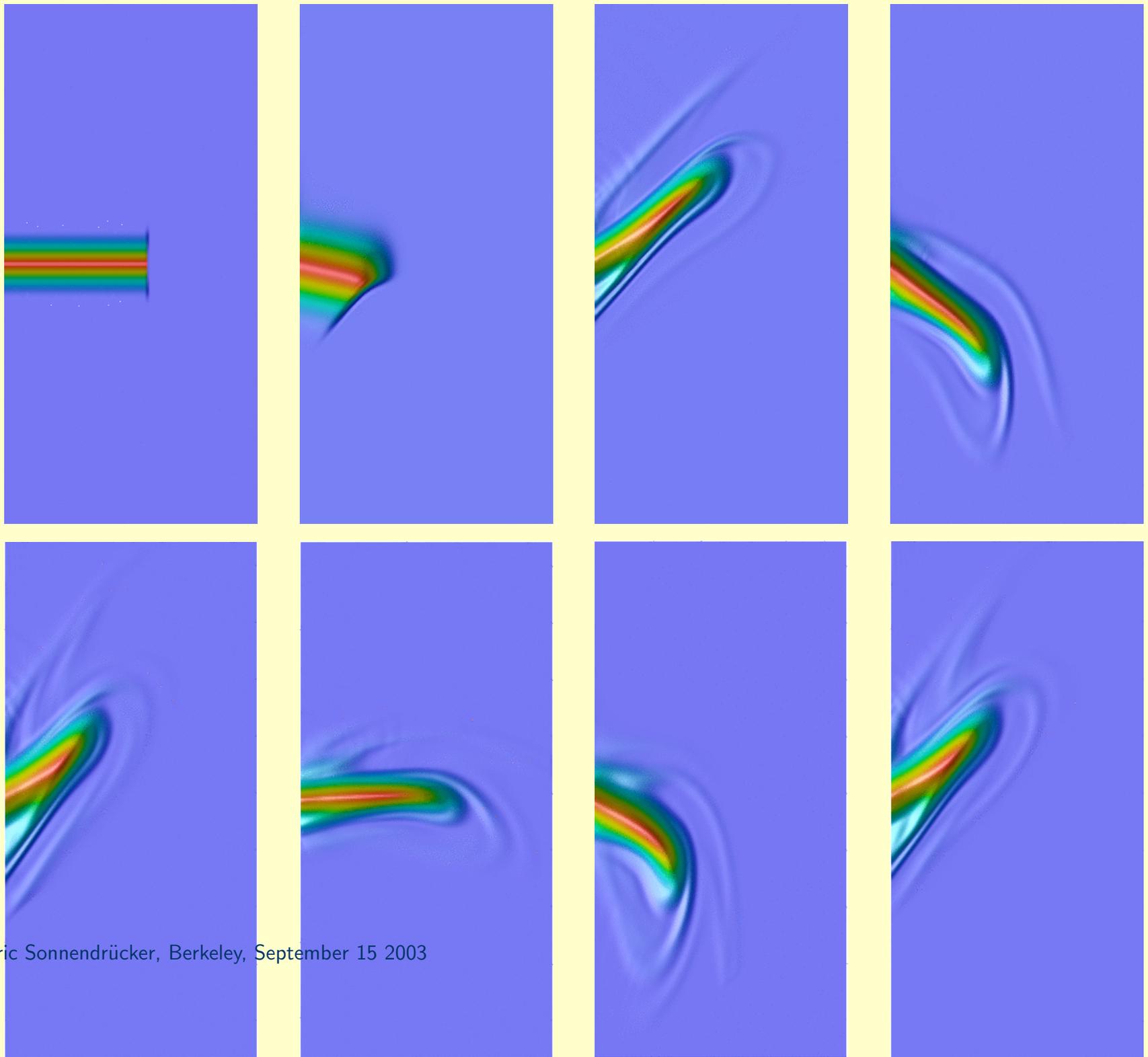


Example of simulation on uniform mesh

- Applied field $\vec{B} = (-\frac{1}{2}B'(z)x, -\frac{1}{2}B'(z)y, B(z))$, with $B(z) = \frac{B_0}{2}(1 + \cos(\frac{2\pi z}{s}))$, with $B_0 = 2T$ and $S = 1m$.
- Semi-Gaussian beam of emittance $\epsilon = 10^{-3}$,

$$f_0(r, v_r, P_\theta) = \frac{n_0}{\pi a^2} \exp\left(-\frac{v_r^2 + (P_\theta/(mr))^2}{2v_{th}^2}\right),$$

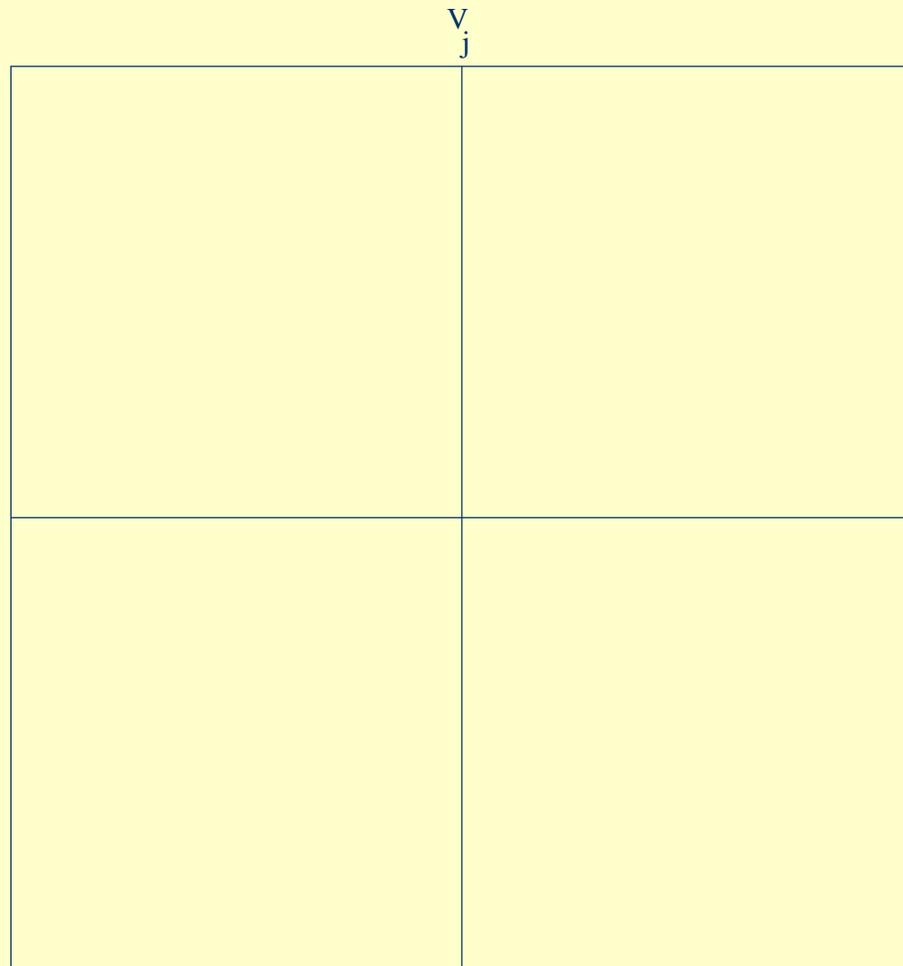
where $P_\theta = mrv_\theta + mB(z)\frac{r^2}{2}$, $n_0 = \frac{I}{qv_z}$, $I = 0.05A$ and $E = 80MeV$ so that $v_z = 626084ms^{-1}$.



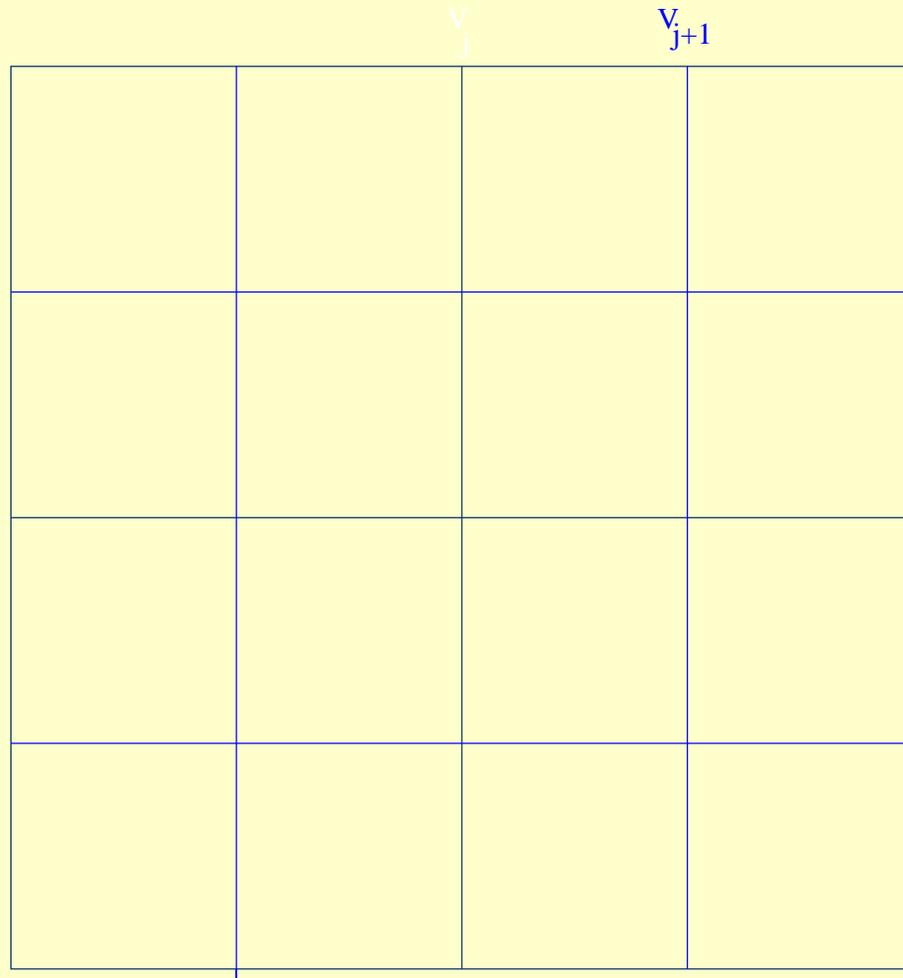
Eric Sonnendrücker, Berkeley, September 15 2003

Adaptive semi-Lagrangian method

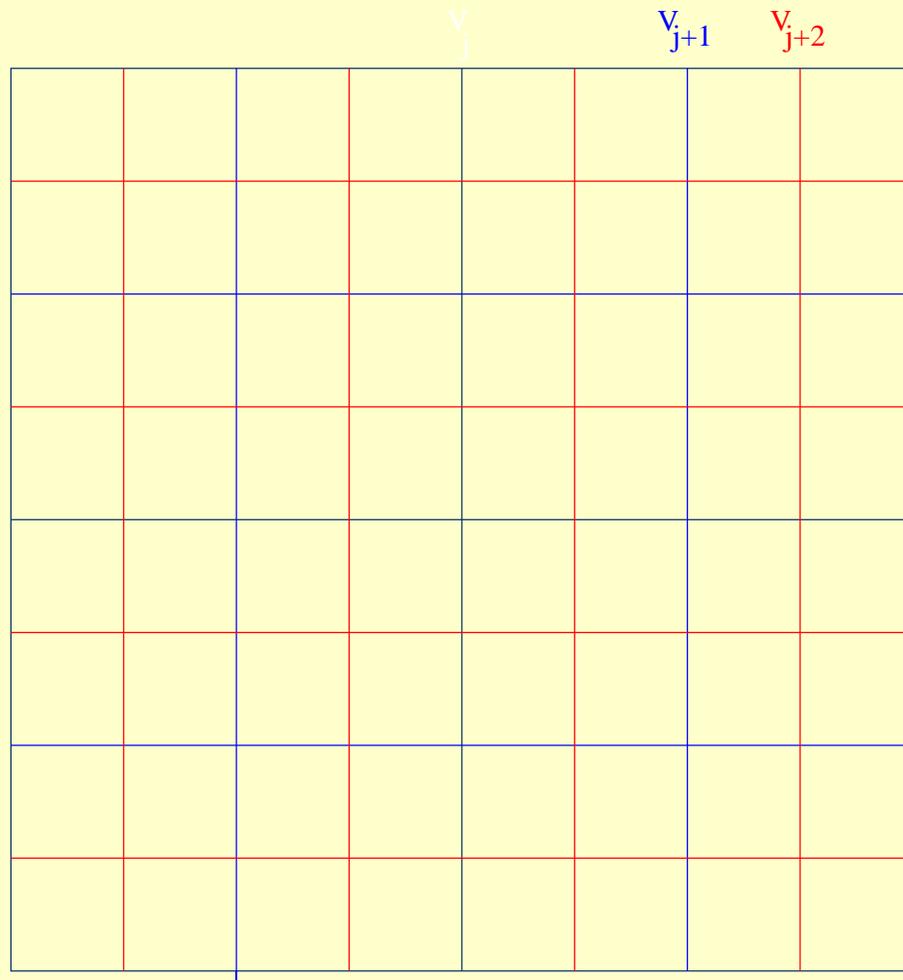
- We want to optimize the number of grid points for a given numerical error.
- Multi-resolution techniques using interpolating wavelets are well suited to determine where refinement is needed.
- Principle of the method
 - Use different levels of meshes
 - At one given level, decompose gridfunction into gridfunction at coarser level + details.



Grid G_j , grid points $x_k^j = k 2^j$, level j



Grid G_{j+1} , grid points $x_k^{j+1} = k 2^{j+1}$, level $j + 1$



Grid G_{j+2} , grid points $x_k^{j+2} = k 2^{j+2}$, level $j + 2$

The wavelet decomposition

- **Idea:** Decompose more precise sample, i.e. values of f at grid points of G_{j+1} (denoted by c_{i+1}) into smaller sample i.e. values of f at grid points of G_j (denoted by c_i) + details (denoted by d_i).
- Details contain difference between exact value and value predicted using interpolation operator.

$$c_{2k}^{j+1} = c_k^j \quad \text{same value at coarse mesh points}$$

$$d_k^j = c_{2k+1}^{j+1} - P_{2N+1}(x_{2k+1}^{j+1}).$$

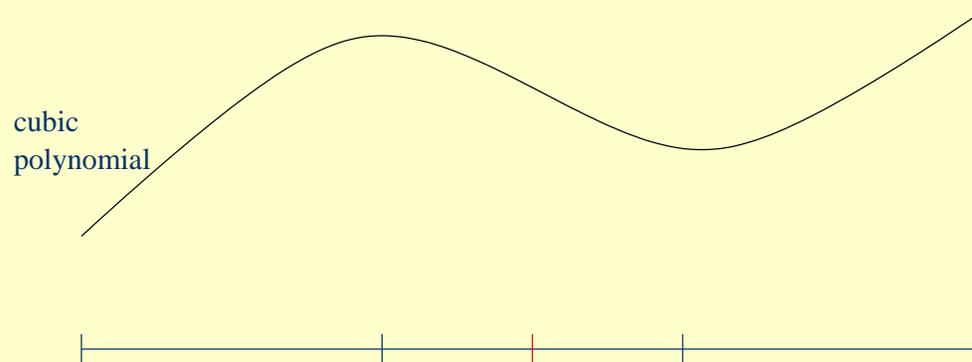
Prediction operator

Predict values at unknown positions of finer level using Lagrange interpolating polynomial on coarser level.



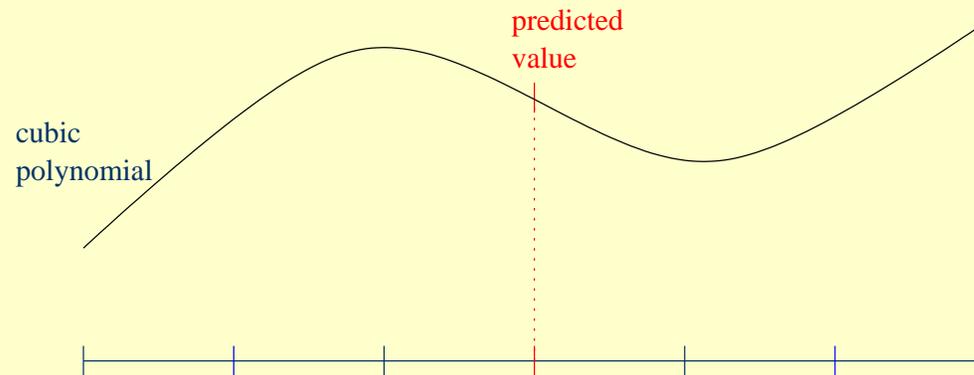
Prediction operator

Predict values at unknown positions of finer level using Lagrange interpolating polynomial on coarser level.



Prediction operator

Predict values at unknown positions of finer level using Lagrange interpolating polynomial on coarser level.



$$d_k^j = c_{2k+1}^{j+1} - P_{2N+1}(x_{2k+1}^{j+1}) \text{ and } c_{2k}^{j+1} = c_k^j$$

Adaptivity and semi-Lagrangian method

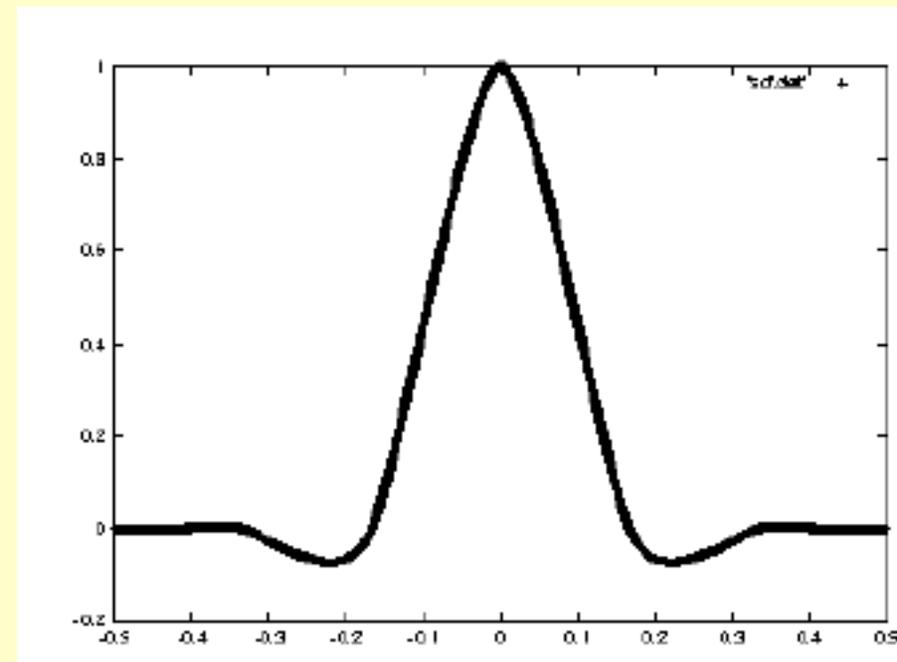
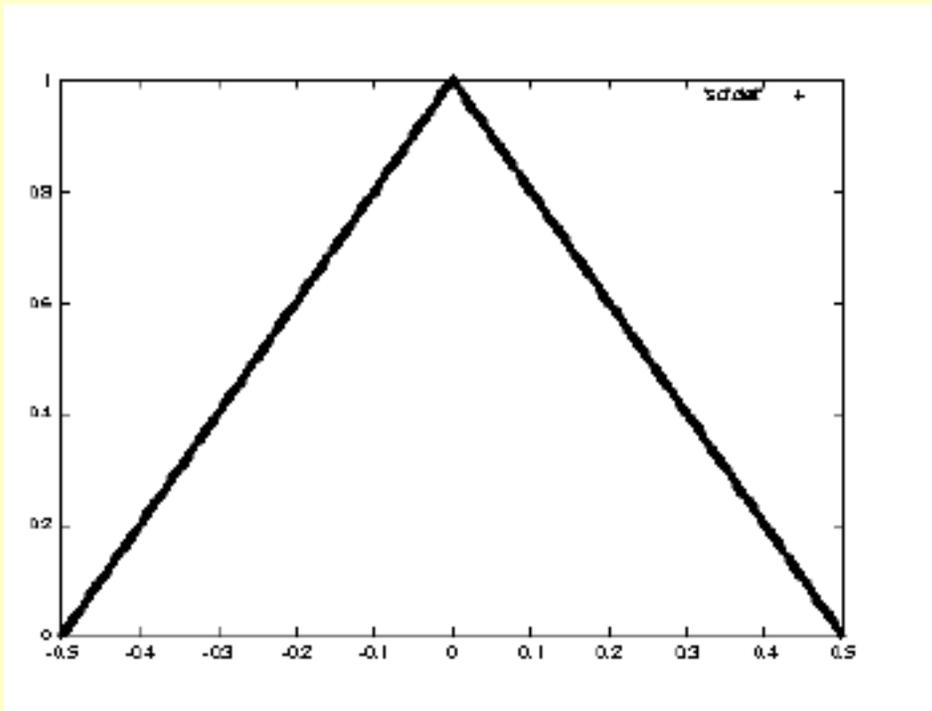
- Semi-Lagrangian method based on polynomial interpolation.
- Main idea for adaptivity: details are small where interpolation does a good job.
- In adaptive method, use wavelet decomposition to eliminate grid points corresponding to small details.
- No loss of information due to wavelet decomposition.
- Grid points where $|d_k^j| < \epsilon$ are removed
→ controlled loss of information.

Hierarchical expression of the information

- Express data given by values on a given set of points as data given on a coarser sample + missing piece.
- Define prediction and restriction operators based on simple operations.
- In terms of function spaces

$$V_{j+1} = V_j \oplus W_j.$$

Scaling function



Wavelet interpolation

Interpolation formula

$$f^*(x, v) = \sum_{k_1, k_2} \left(c_{k_1, k_2}^{j_0} \varphi_{k_1}^{j_0}(x) \varphi_{k_2}^{j_0}(v) \right. \\ \left. + \sum_{j_0}^{j_1-1} \left(d_{k_1, k_2}^{row, j} \psi_{k_1}^{j+1}(x) \varphi_{k_2}^j(v) \right. \right. \\ \left. \left. + d_{k_1, k_2}^{col, j} \varphi_{k_1}^j(x) \psi_{k_2}^j(v) \right. \right. \\ \left. \left. + d_{k_1, k_2}^{mid, j} \psi_{k_1}^{j+1}(x) \psi_{k_2}^{j+1}(v) \right) \right) \quad (1)$$

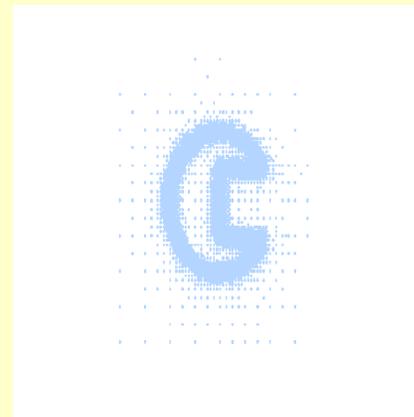
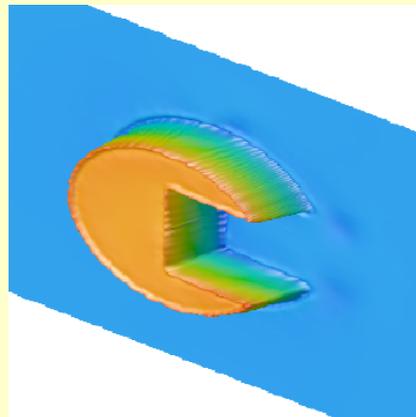
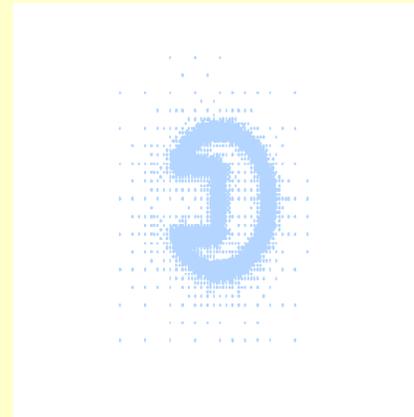
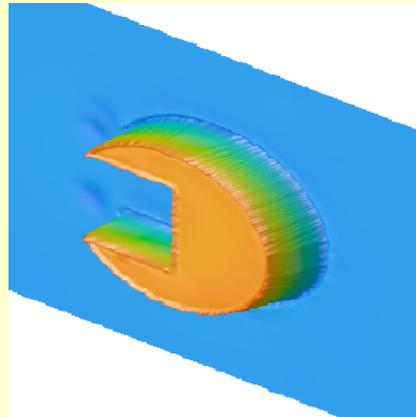
The Algorithm for the Vlasov Problem...

- **Initialisation:** decomposition and **compression** of f_0 .
- **Prediction in x** of the grid \tilde{G} (for important details) at the next split time step following the characteristics forward. Retain points at level just finer.
- **Construction of \hat{G} :** grid where we have to compute values of f^* in order to compute its wavelet transform.

...The Algorithm for the Vlasov Problem...

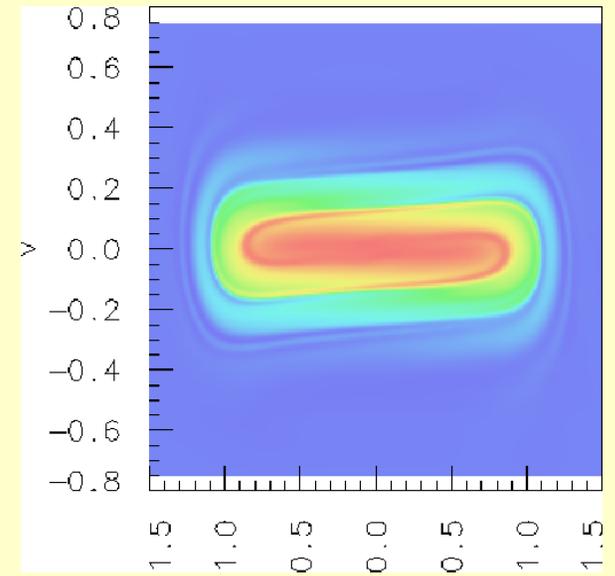
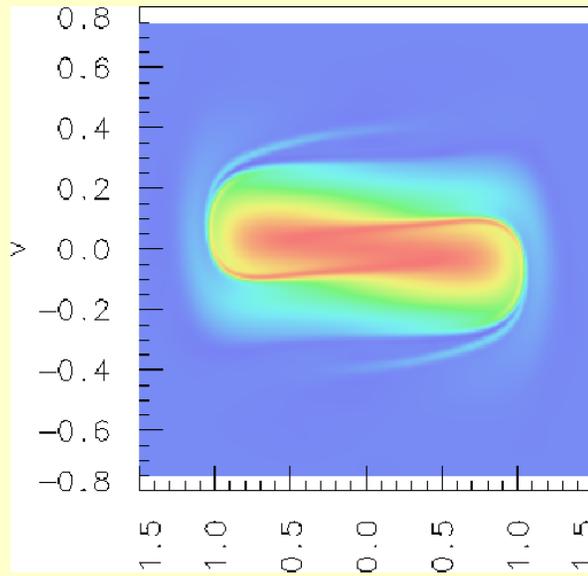
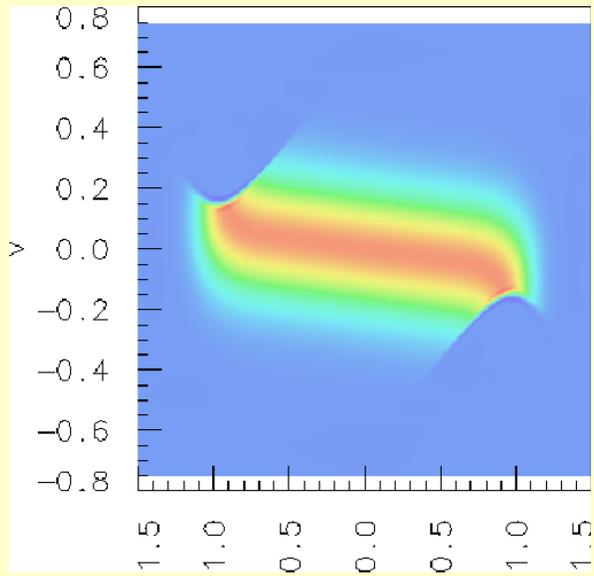
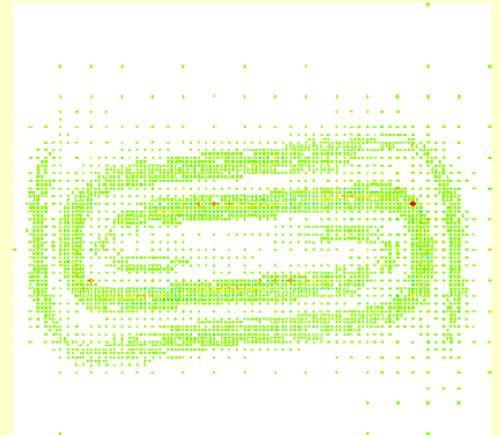
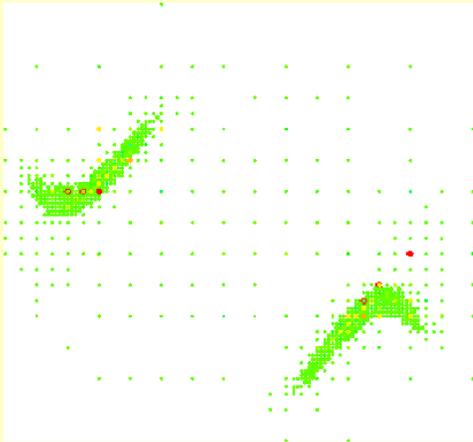
- **Advection-interpolation in x :** follow the characteristics backwards in x and interpolate using wavelet decomposition (1):
$$f^*(x, v) = f^n(x - v \Delta t, v)$$
- **Wavelet transform of f^* :** compute the c_k and d_k coefficients at the points of \tilde{G} .
- **Computation** of electric field from Poisson.
- Same procedure for the velocity advance.

Sharp edge beam evolution in applied field



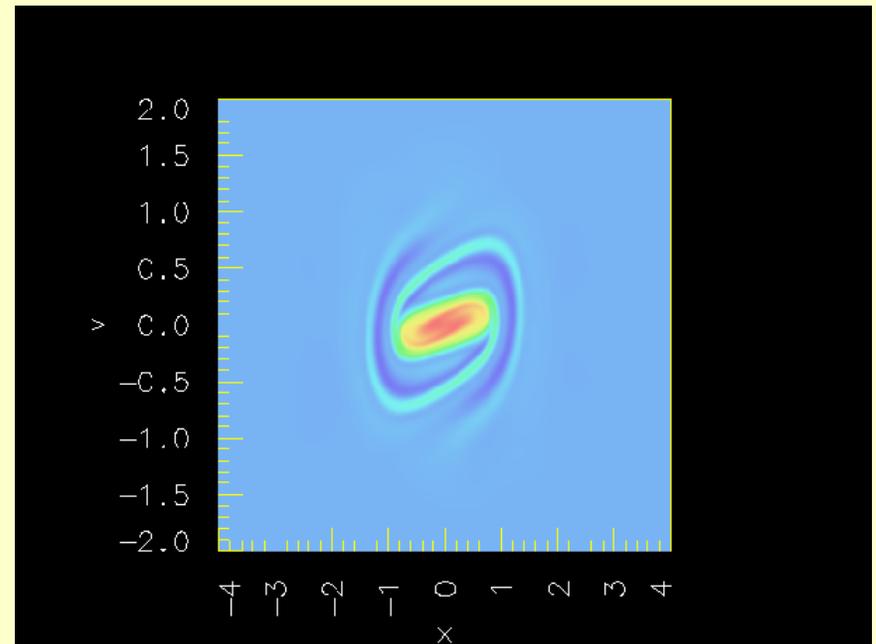
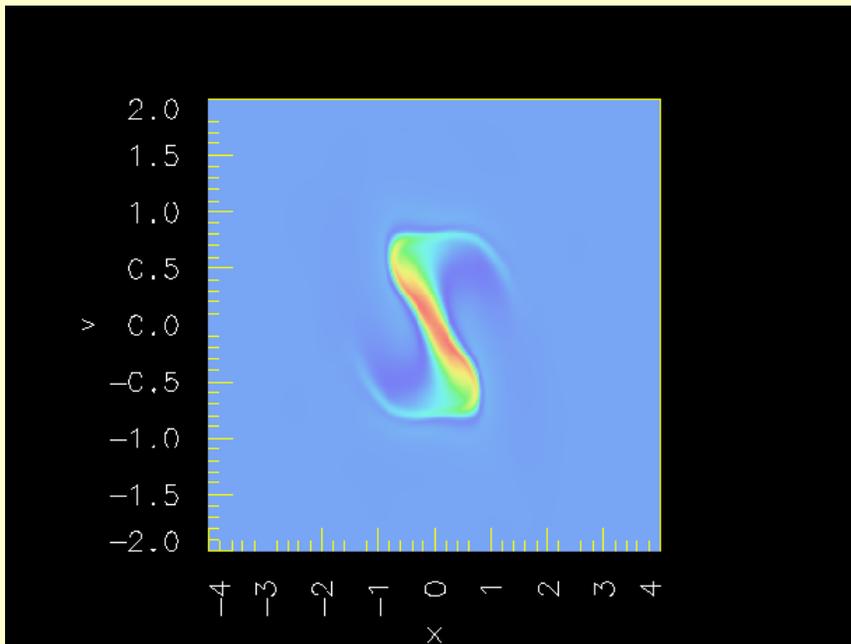
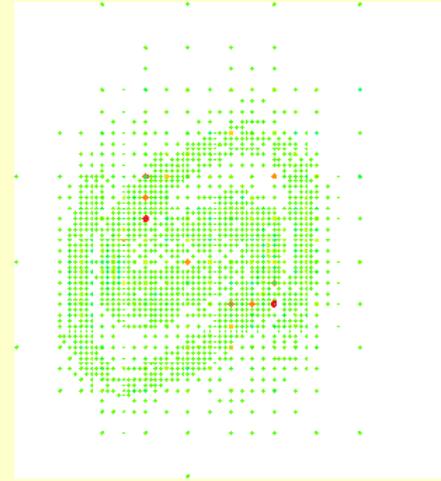
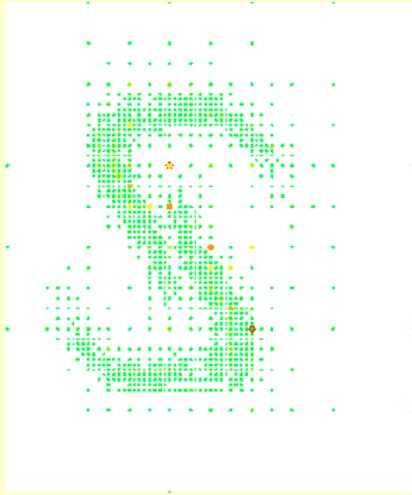
Semi-Gaussian beam evolution in uniform focusing channel

- Potassium ions
- Beam energy 80 keV
- Uniform focusing
- Tune depression 0.25



Semi-Gaussian beam evolution in periodic focusing channel

- Potassium ions
- Beam energy 80 keV
- Periodic focusing field of the form $\alpha(1 + \cos 2\pi z/S)$.
- Tune depression 0.17



What about a moving computational domain

- For beam simulations large gain can already be expected by moving computation box.
- Computation box could be determined from envelope.
- Expect much easier implementation.
- Splitting algorithm would not work anymore.

The splitting algorithm

- Enables to bring back the Vlasov equation to two constant coefficients advection equations on each time step :

- First solve

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = 0$$

- Then solve

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

- High order accuracy can be achieved by alternating the solves in the right way.

Pros and cons of splitting algorithm

- Time splitting enables to bring back a Vlasov solver to a succession of constant coefficient advections → no ODE solver needed for the characteristics.
- **HOWEVER:**
 - Axes are privileged directions in such an algorithm.
 - Does not work anymore if grid directions are not parallel to x and v directions as can happen in transform method.

A non split time stepping algorithm

Knowing the final position (X^{n+1}, V^{n+1}) at time step t_{n+1} , as well as f^n , ρ^{n-1} , \mathbf{E}^n we can compute the initial position (X^n, V^n) using the following algorithm:

1. Predict $\bar{\mathbf{E}}^{n+1}$ using the continuity equation (or directly Ampere's law in 1D)

$$\rho^{n+1} = \rho^{n-1} - 2\Delta t \nabla \cdot \mathbf{J}^n, \quad \mathbf{J}^n = q \int f^n(\mathbf{x}, \mathbf{v}) \mathbf{v} d\mathbf{v},$$

$$-\nabla^2 \phi^{n+1} = \frac{\rho^{n+1}}{\epsilon_0}, \quad \bar{\mathbf{E}}^{n+1} = -\nabla \phi^{n+1}.$$

2. $\mathbf{V}^{n+\frac{1}{2}} = \mathbf{V}^{n+1} - \frac{\Delta t}{2} \bar{\mathbf{E}}^{n+1}(\mathbf{X}^{n+1});$
3. $\mathbf{X}^n = \mathbf{X}^{n+1} - \Delta t \mathbf{V}^{n+\frac{1}{2}};$
4. $\mathbf{V}^n = \mathbf{V}^{n+\frac{1}{2}} - \frac{\Delta t}{2} \mathbf{E}^n(\mathbf{X}^n).$
5. $f^{n+1}(\mathbf{X}^{n+1}, \mathbf{V}^{n+1}) = \textit{interpolation}(f^n)(\mathbf{X}^n, \mathbf{V}^n);$
6. $\rho^{n+1} = \int f^{n+1} dv,$
7. Correct $\bar{\mathbf{E}}^{n+1}$ using $-\nabla^2 \phi^{n+1} = \frac{\rho^{n+1}}{\epsilon_0}, \quad \bar{\mathbf{E}}^{n+1} = -\nabla \phi^{n+1}.$
8. If $\|\bar{\mathbf{E}}^{n+1} - \bar{\mathbf{E}}_{prev}^{n+1}\| > \textit{threshold}$ go back to 2.

Towards the moving grid algorithm

The semi-Lagrangian method consists in two conceptually different steps:

1. **Advection step:** follow particle trajectories. *completely independent of the grid and most naturally performed in the physical space*
2. **Interpolation step:** Interpolation grid needed to reconstruct the distribution function at every point in phase space at one given time step, needs not be the same at two different time steps. Ideal if grid points exactly on particle trajectories.

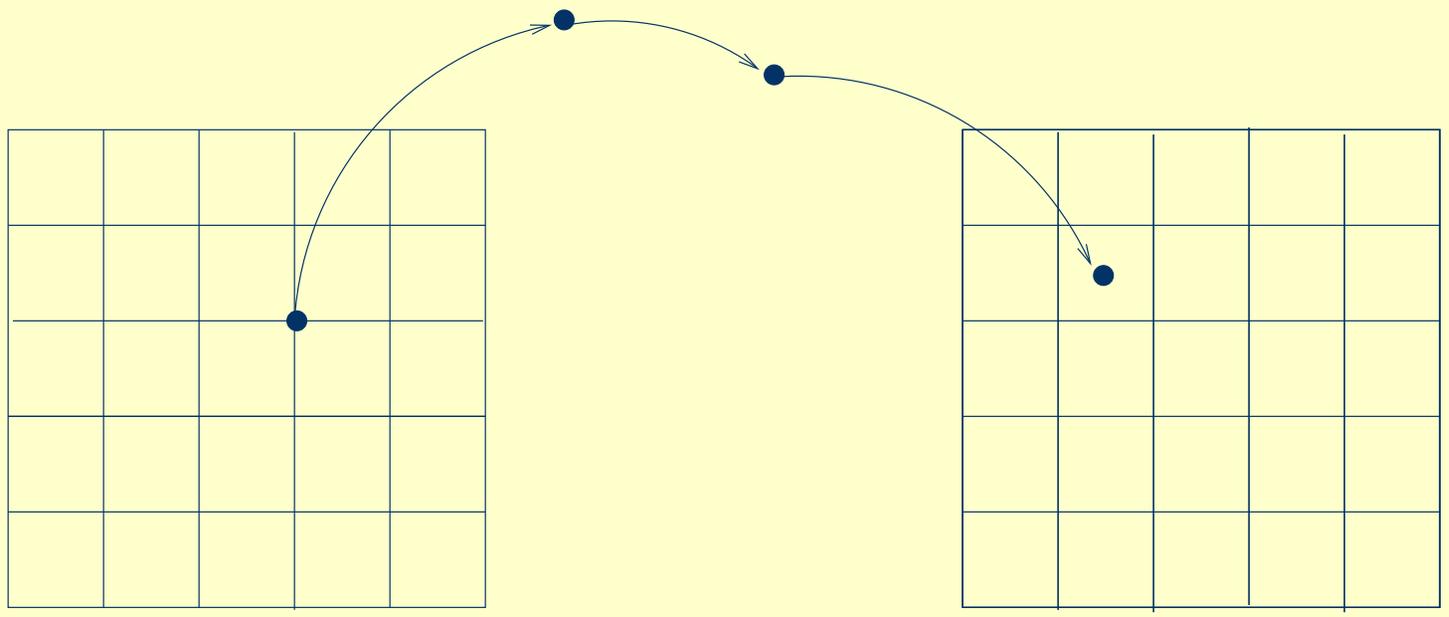
Desired features of efficient solver

- Use **optimal number of grid points** to reconstruct distribution function f at any given time with a given accuracy.
- Minimize number of wasted grid points (computations in zones of vanishing f).
- Have grid points follow particle trajectories → minimize interpolation errors.

The transform method

- Define at each time step an invertible mapping φ_t from a logical grid to the physical grid.
- This mapping needs to be known or constructed automatically.
- Distribution function on logical grid $f^*(x^*, v^*, t) = f(\varphi_t(x^*, v^*), t)$.
* denotes quantities on logical grid.
- f^* satisfies the following conservation property

$$f^*(x^*, v^*, t) = f^*(X^*(s; x, v, t), V^*(s; x, v, t), s)$$



The algorithm

1. Compute positions in physical phase-space of grid points where f^{n+1} is to be computed: $(x_{i,j}^{n+1}, v_{i,j}^{n+1}) = \varphi_{n+1}(x_i^*, v_j^*)$, where (x_i^*, v_j^*) are the nodes of the logical grid.
2. Compute origin of grid points $(x_{i,j}^{n+1}, v_{i,j}^{n+1})$ using algorithm 1 or similar. We denote by $(X_{i,j}^n, V_{i,j}^n)$ these origins.
3. Transform $(X_{i,j}^n, V_{i,j}^n)$ back to the logical grid at time t_n :
 $(X_{i,j}^{*n}, V_{i,j}^{*n}) = \varphi_n^{-1}(X_{i,j}^n, V_{i,j}^n)$.
4. Interpolate f^{*n} at origin of characteristics on logical grid to get f^{n+1} , as $f^{*(n+1)}(x_i^*, v_j^*) = f^{*n}(X_{i,j}^{*n}, V_{i,j}^{*n})$.

Coupling with the Poisson equation

- **Uniform grid:** ρ obtained by summing grid values of f over v grid values for each x .
- **Moving grid:** In general grid lines do not follow x or v . Hence additional interpolation step might be necessary to compute ρ .
- e.g. integrate using Gauss quadrature and interpolate ρ at Gauss points.
- Other option: try and keep transformation which does not change x .

Beam simulation in transverse phase space

Find transform following beam envelope.

e.g. RMS equivalent ellipse satisfies

$$\tan 2\theta = \frac{2\langle xx' \rangle}{\langle x^2 \rangle - \langle x'^2 \rangle},$$

$$a = \sqrt{2(\cos^2 \theta \langle x^2 \rangle + \sin^2 \theta \langle x'^2 \rangle + 2 \sin \theta \cos \theta \langle xx' \rangle)},$$

$$b = \sqrt{2(\sin^2 \theta \langle x^2 \rangle + \cos^2 \theta \langle x'^2 \rangle - 2 \sin \theta \cos \theta \langle xx' \rangle)}.$$

Numerical results

Toy problem: transverse axisymmetric Vlasov-Poisson equation with vanishing canonical angular momentum. This problem reads

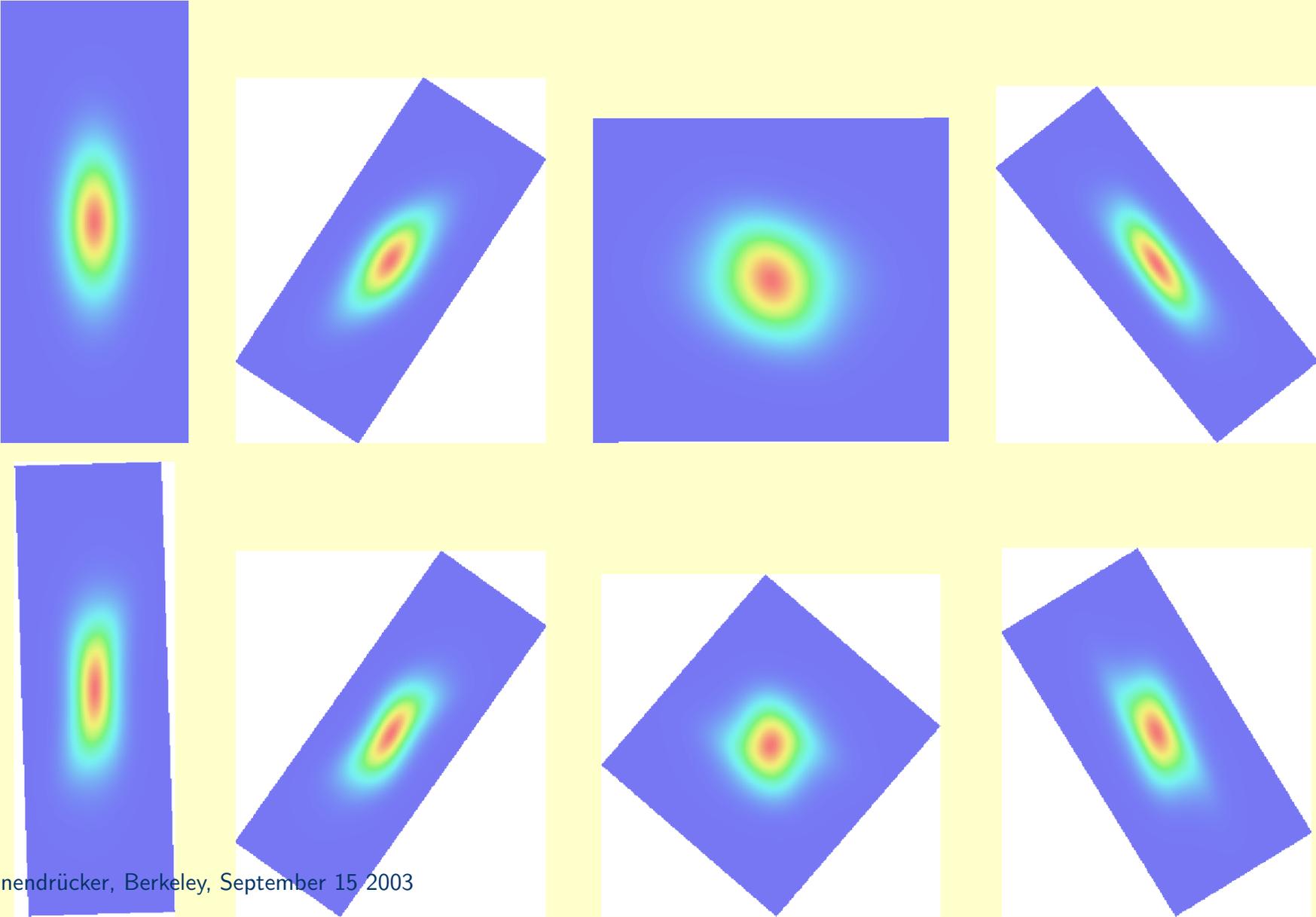
$$\frac{\partial f}{\partial t} + v_r \frac{\partial f}{\partial r} + \left(F_{app} + \frac{q}{m} E_r \right) \frac{\partial f}{\partial v_r} = 0,$$

$$\frac{1}{r} \frac{d}{dr} (r E_r) = \rho = \int f dv_r.$$

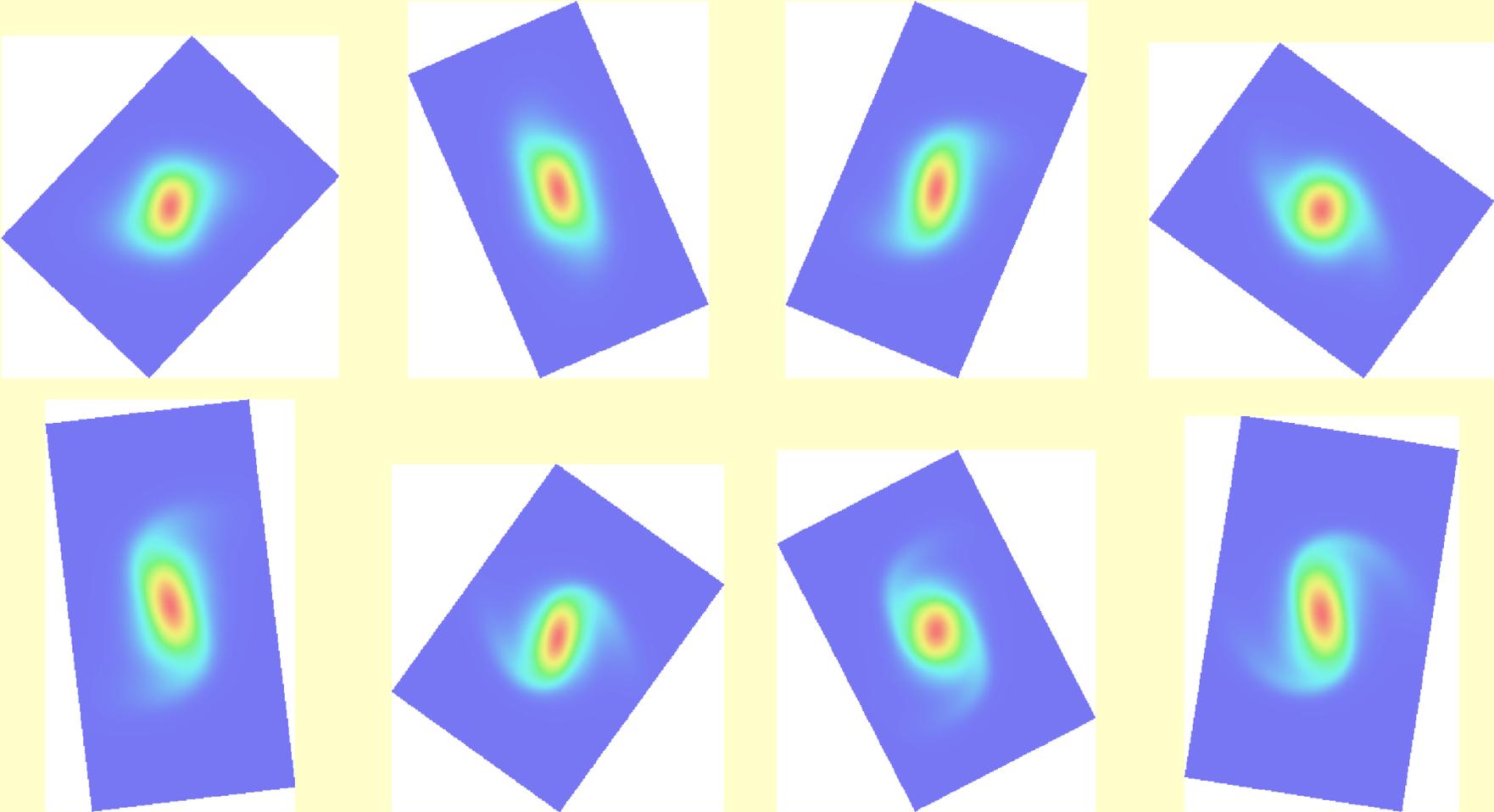
Examples with important envelope motion.

1. Mismatched beam in a uniform focusing channel.
2. Matched beam in a periodic focusing channel.

Evolution of mismatched Gaussian beam in uniform focusing channel



Evolution of matched Gaussian beam in peirodic focusing channel



Conclusions

- Adaptive method looks promising in 2D phase space.
- Wavelet method does a very good job in finding useful grid points.
- **However**
 - Large overhead, due to adaptivity. Method efficient if there are a lot fewer points in adaptive grid.
 - Implementation more complex due to adaptive mesh structure.
- Moving grid semi-Lagrangian method feasible and easy to implement.

- Interesting for beam simulations where computing box can be easily determined from envelope motion.
- Needs to be implemented for realistic 2D simulations.
- Other kinds of transforms enabling to follow more closely the particle trajectories should be tried.