Vlasov simulations in 6D phase space based on compressed data formats

Katharina Kormann & Eric Sonnendrücker

Max-Planck Institute for Plasma Physics

and

TU Munich

IPAM, UCLA, February 2, 2017

Outline

An short overview of magnetic fusion models

Kinetic or gyrokinetic model

The Geometric Particle-In-Cell method (GEMPIC) (M. Kraus, K. Kormann, P. Morrison, ES)

Semi-Lagrangian Vlasov solvers

Implementation on a sparse grid

Implementation with hierarchical tensor decomposition

Outline

An short overview of magnetic fusion models

Kinetic or gyrokinetic model

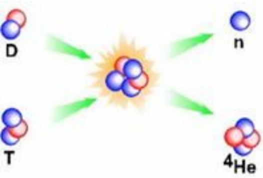
The Geometric Particle-In-Cell method (GEMPIC) (M. Kraus, K. Kormann, P. Morrison, ES)

Semi-Lagrangian Vlasov solvers

Implementation on a sparse grid

Implementation with hierarchical tensor decomposition

Controlled thermonuclear fusion

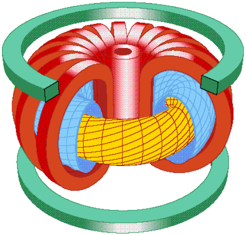


- Fusion conditions:
  $nT\tau_E$ large enough.
- $T \approx 100$ million ^oC
  fully ionized gas=plasma.

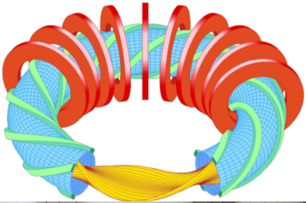
- Magnetic confinement (ITER)
- Inertial confinement
  - by laser (LMJ, NIF)
  - by heavy ions

Two devices for magnetic fusion: tokamaks and stellarators

Tokamak



Stellarator



Modelling of Tokamak plasmas

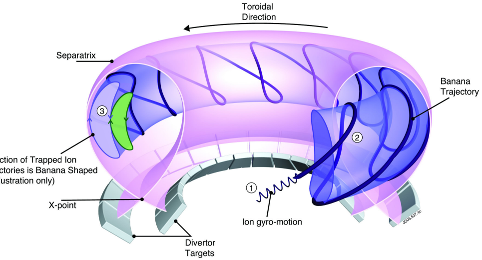
- A plasma is a collection of different species of charged particles.
- Basic model is Newton's law with pairwise interaction between particles which is largely dominated by electromagnetic force. Too many particles $n \approx 10^{19} m^{-3}$, numerically intractable.
- First reduced model: Kinetic Vlasov-Maxwell (+Landau collisions)
- Second reduced model: multi-fluid Euler-Maxwell
- Third reduced model: single fluid MHD
- Other reduced model: Maxwell's equation with dielectric tensor representing plasma

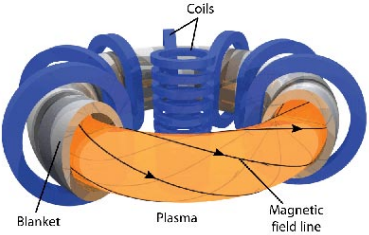
Kinetic models: Turbulent transport

- Plasma not very collisional and far from fluid state
  $\Rightarrow$ Kinetic description necessary for shorter time scales. Fluid and kinetic simulations of turbulent transport yield very different results.
- Vlasov (6D phase space) coupled to 3D Maxwell

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

- Toroidal geometry





Outline

An short overview of magnetic fusion models

Kinetic or gyrokinetic model

The Geometric Particle-In-Cell method (GEMPIC) (M. Kraus, K. Kormann, P. Morrison, ES)

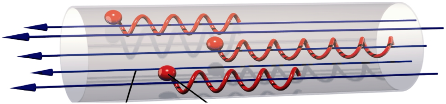
Semi-Lagrangian Vlasov solvers

Implementation on a sparse grid

Implementation with hierarchical tensor decomposition

Numerical issues with 6D Vlasov-Maxwell

- Posed in 6D phase space! Dimension reduction if possible would help.
- Large magnetic field imposes very small time step to resolve the rotation of particles along field lines.

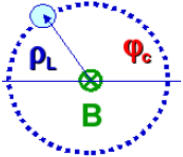


- Physics of interest is low frequency. Remove light waves: Darwin instead of Maxwell.
- Debye length small compared to ion Larmor radius. Quasi-neutrality assumption $n_e = n_i$ needs to be imposed instead of Poisson equation for electric field.

The gyrokinetic model

- Scale separation: fast motion around magnetic field lines can be averaged out.
- Idea: separate motion of the guiding centre from rotation by a change of coordinates.

- For constant magnetic field can be done by change of coordinates: $\mathbf{X} = \mathbf{x} - \rho_L$ guiding centre + kind of cylindrical coordinates in $\mathbf{v}$: $v_{\parallel}$, $\mu = \frac{1}{2} m v_{\perp}^2 / \omega_c$, θ.



- Mixes position and velocity variables.
- Perturbative model for slowly varying magnetic field.
- Several small parameters
  - gyroperiod, Debye length
  - Magnetic field in tokamak varies slowly: $\epsilon_B = |\nabla B / B|$
  - Time dependent fluctuating fields are small.
- Gain: Gyrokinetic model posed in 5D phase space including 1 invariant + fastest time scale removed.

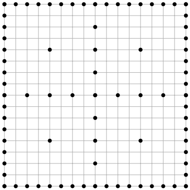
Limits of gyrokinetic model

- Gyrokinetic model has proven useful as well for fusion plasmas as for some magnetised space plasmas.
- However there are regimes, where the hypotheses its derivation is based on are not fully justified. In particular the tokamak edge.
- It is necessary to also consider simulation of the full Vlasov-Maxwell model in 6D at least for verification purposes.
- Highly optimised 6D codes exist as well Eulerian as PIC, but they might be suboptimal considering the storage of data: full particle phase space information or 6D grid data.
  $\rightarrow$ look for reduced data structures
- Techniques developed here could apply for gyrokinetic equations as well, but model more complex.

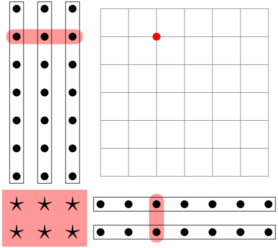
Vlasov solvers using reduced data formats

Use methods designed specifically for high dimensional problems

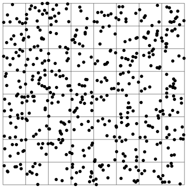
Sparse grids



Hierarchical Tucker Tensors



Monte Carlo (PIC)



- *Sparse grids: K. Kormann, ES: "Sparse Grids for the Vlasov-Poisson Equation." Sparse Grids and Applications-Stuttgart 2014. Springer 2016.*
- *Hierarchical Tensor: K. Kormann, A semi-Lagrangian Vlasov solver in tensor train format, SIAM Journal on Scientific Computing (2015).*

Outline

Hamiltonian structure of Vlasov-Maxwell system

The time evolution of any functional $F[f_s, \mathbf{E}, \mathbf{B}]$ is given by

$$\frac{\mathrm{d}}{\mathrm{d}t} F[f_s, \mathbf{E}, \mathbf{B}] = \{F, \mathcal{H}\} \,,$$

$$\text{with } \mathcal{H} = \sum_s \frac{m_s}{2} \int |\mathbf{v}|^2 \, f_s(\mathbf{x}, \mathbf{v}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{v} + \frac{1}{2} \int \left(|\mathbf{E}(\mathbf{x})|^2 + |\mathbf{B}(\mathbf{x})|^2 \right) \mathrm{d}\mathbf{x}.$$

$$\begin{aligned}
\text{and } \{F, G\} [f_s, \mathbf{E}, \mathbf{B}] = & \sum_s \int \frac{f_s}{m_s} \left[\frac{\delta F}{\delta f_s}, \frac{\delta G}{\delta f_s} \right] \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{v} \\
& + \sum_s \frac{q_s}{m_s} \int f_s \left(\nabla_{\mathbf{v}} \frac{\delta F}{\delta f_s} \cdot \frac{\delta G}{\delta \mathbf{E}} - \nabla_{\mathbf{v}} \frac{\delta G}{\delta f_s} \cdot \frac{\delta F}{\delta \mathbf{E}} \right) \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{v} \\
& + \sum_s \frac{q_s}{m_s^2} \int f_s \, \mathbf{B} \cdot \left(\nabla_{\mathbf{v}} \frac{\delta F}{\delta f_s} \times \nabla_{\mathbf{v}} \frac{\delta G}{\delta f_s} \right) \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{v} \\
& + \int \left(\nabla \times \frac{\delta F}{\delta \mathbf{E}} \cdot \frac{\delta G}{\delta \mathbf{B}} - \nabla \times \frac{\delta G}{\delta \mathbf{E}} \cdot \frac{\delta F}{\delta \mathbf{B}} \right) \mathrm{d}\mathbf{x},
\end{aligned}$$

where $[f, g] = \nabla_{\mathbf{x}} f \cdot \nabla_{\mathbf{v}} g - \nabla_{\mathbf{x}} g \cdot \nabla_{\mathbf{v}} f$.

Structure preserving discretisation

- Find phase space discretisation that preserves the hamiltonian structure
- The finite dimensional version of a non canonical hamiltonian system yields a so-called Poisson structure:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = J(u)\nabla H$$

  where J is a Poisson matrix (skew-symmetric, Jacobi identity).
- Symplectic structure

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

  is a special case corresponding to canonical hamitonian systems.
- Particle discretisation of distribution function: Monte Carlo approximation. Draw particles from initial distribution function.
- Compatible Finite Element (FEEC) discretisation of electric ($H(curl)$ conforming) and magnetic ($H(div)$ conforming) fields.

Finite Element Exterior Calculus (FEEC)

- In order to preserve the continuous structure at the discrete level, the different unknowns ϕ, $\mathbf{A}$, $\mathbf{E}$ and $\mathbf{B}$ need to be chosen in compatible Finite Element spaces.
- This is provided by Finite Element Exterior Calculus (FEEC) introduced by Arnold, Falk and Winther.
- Continuous and discrete complexes are the following

$$
\begin{array}{ccccccc}
H^1(\Omega) & \overset{\mathbf{grad}}{\longrightarrow} & H(\mathbf{curl}, \Omega) & \overset{\mathbf{curl}}{\longrightarrow} & H(\mathrm{div}, \Omega) & \overset{\mathrm{div}}{\longrightarrow} & L^2(\Omega) \\
\downarrow \Pi_0 & & \downarrow \Pi_1 & & \downarrow \Pi_2 & & \downarrow \Pi_3 \\
V_0 & \longrightarrow & V_1 & \longrightarrow & V_2 & \longrightarrow & V_3
\end{array}
$$

- Faraday and $\mathrm{div}\, B = 0$ are verified strongly as

$$
{}^1\mathbf{E} = -\nabla\, {}^0\phi - \frac{\partial\, {}^1\mathbf{A}}{\partial t}, \qquad {}^2\mathbf{B} = \nabla \times\, {}^1\mathbf{A}.
$$

- Ampere and Gauss' law verified on dual complex.

Spline based De Rham complex

- Several discrete complexes have been discovered.
- A. Buffa, J. Rivas, G. Sangalli, R. Vásquez (2010) introduced a discrete exact sequence for spline Finite Elements.

$$
\begin{array}{ccccccc}
H^1 & \xrightarrow{\textbf{grad}} & H(\textbf{curl}) & \xrightarrow{\textbf{curl}} & H(\mathrm{div}) & \xrightarrow{\mathrm{div}} & L^2 \\
\cup & & \cup & & \cup & & \\
V_0 & \longrightarrow & V_1 & \longrightarrow & V_2 & \longrightarrow & V_3 \\
= & & = & & = & & = \\
\mathcal{S}^{p,p,p} & \longrightarrow & \begin{pmatrix} \mathcal{S}^{p-1,p,p} \\ \mathcal{S}^{p,p-1,p} \\ \mathcal{S}^{p,p,p-1} \end{pmatrix} & \longrightarrow & \begin{pmatrix} \mathcal{S}^{p,p-1,p-1} \\ \mathcal{S}^{p-1,p,p-1} \\ \mathcal{S}^{p-1,p-1,p} \end{pmatrix} & \longrightarrow & \mathcal{S}^{p-1,p-1,p-1}
\end{array}
$$

- Based on tensor product splines on cartesian grid. $\mathcal{S}^{p_1,p_2,p_3}$ denotes tensor product of splines of degree p_1 in first direction, p_2 in second and p_3 in third.

The final semi-discrete system

- Putting all together we get an ODE system involving the particle positions and velocities as well as the FE dofs of the electric and magnetic fields

$$\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}t} = \mathbf{v}_k, \quad m\frac{d\mathbf{v}_k}{dt} = q(\mathbf{E}_h(t, \mathbf{x}_k) + \mathbf{v}_k \times \mathbf{B}_h(t, \mathbf{x}_k)),$$

$$M_1\frac{\mathrm{d}\mathbf{e}}{\mathrm{d}t} = R_{12}^\top M_2\mathbf{b} + \sum_k w_k\mathbf{v}_k\mathbf{\Lambda}^1(\mathbf{x}_k), \quad \frac{\mathrm{d}\mathbf{b}}{\mathrm{d}t} = -R_{12}\mathbf{e}.$$

M_1, M_2 mass matrices in V_1 and V_2, R_{12} discrete curl incidence matrix.

- The following semi-discrete energy is exactly conserved

$$H = \frac{1}{2}\mathbf{v}^\top M_p\mathbf{v} + \frac{1}{2}\mathbf{e}^\top M_1\mathbf{e} + \frac{1}{2}\mathbf{b}^\top M_2\mathbf{b}.$$

denoting by $\mathbf{v}^\top M_p\mathbf{v} = \frac{1}{2}\sum_k w_k v_k^2$

Poisson structure

- Our semi-discrete system fits into a Poisson structure

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \Omega(u)\nabla H,$$

where H is the conserved energy, $u = (\mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{b})^\top$ and the Poisson matrix

$$\Omega(u) = \begin{pmatrix} 0 & M_p^{-1} & 0 & 0 \\ -M_p^{-1} & \mathbf{B}_h^\top(t, x_p)\, M_p^{-1} & (\Lambda^1(x_p))^\top M_1^{-1} & 0 \\ 0 & -M_1^{-1}(\Lambda^1(x_p)) & 0 & M_1^{-1} R^\top \\ 0 & 0 & -R M_1^{-1} & 0 \end{pmatrix}$$

- We see that Ω is anti-symmetric, it remains to show that it satisfies the Jacobi identity to define a Poisson structure, such that

$$\{F, G\} = \nabla F^\top \Omega \nabla G$$

defines a Poisson bracket.

Properties of Poisson structure

- The Jacobi identity can be proved by using a lemma from Hairer-Lubich-Wanner relating the terms of the matrix
- Most terms in the matrix are constant and pose no problem.
- Key identities come from
  1. $\nabla \cdot \mathbf{B}_h = 0$
  2. Derivatives of Λ^1 basis functions related to Λ^2 basis functions and curl incidence matrix.
- Two casimirs (invariants of motion) are found:
  - $\nabla \cdot \mathbf{B}_h$
  - Discrete Gauss law: $\int \mathbf{E}_h \cdot \nabla q \,\mathrm{d}x + \int \rho q \,\mathrm{d}\mathbf{x} \qquad \forall q \in V_0$
  - No projection needed.

Time advance via Hamiltonian splitting

- Following the prescription of Crouseilles-Einkemmer-Faou (JCP 15) a Hamiltonian splitting can be performed, treating the three terms of the Hamiltonian separately

$$H = \frac{1}{2}\mathbf{v}^\top M_p \mathbf{v} + \frac{1}{2}\mathbf{e}^\top M_1 \mathbf{e} + \frac{1}{2}\mathbf{b}^\top M_2 \mathbf{b} = H_p + H_e + H_b.$$

- Split and solve successively

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \Omega(u)\nabla H_i, \quad i = p, e, b$$

- Lie-Trotter splitting (first order), Strang splitting (second order) or even higher order.
- Exact solution possible for H_e and H_b.
- For H_p split further between the three components. Other possibility: use variational integrator

Discrete solution of H_e

- The equations read

$$\begin{aligned}\dot{x}_p &= 0,\\ \dot{v}_p &= \left(\Lambda^1(x_p(t))\right)^\top e(t),\\ \dot{e} &= 0,\\ \dot{b} &= -Re(t).\end{aligned}$$

- and its exact solution on one time step

$$\begin{aligned}x_p(h) &= x_p(0),\\ v_p(h) &= v_p(0) + \left(\Lambda^1(x_p(0))\right)^T e(0),\\ e(h) &= e(0),\\ b(h) &= b(0) - Re(0).\end{aligned}$$

Discrete solution of H_b

- The equations read

$$
\begin{aligned}
\dot{x}_p &= 0, \\
\dot{v}_p &= 0, \\
M_1 \dot{e} &= R^T M_2 b(t), \\
\dot{b} &= 0.
\end{aligned}
$$

- and its exact solution on one time step

$$
\begin{aligned}
x_p(h) &= x_p(0), \\
v_p(h) &= v_p(0), \\
M_1 e(h) &= M_1 e(0) + R^T M_2 b(0), \\
b(h) &= b(0).
\end{aligned}
$$

Discrete solution of H_p

- The equations read

$$\begin{aligned} \dot{x}_p &= v_p, \\ \dot{v}_p &= v_p \times b(t)\, \Lambda^2(x_p), \\ M_1 \dot{e} &= -\Lambda^1(x_p) \cdot M_p v_p, \\ \dot{b} &= 0, \end{aligned}$$

- For general magnetic field coefficients b, this system cannot be exactly integrated
- As each component of the equation for $\dot{v}_p$ does not depend on v_p, we can split this system once more into

$$H_p = H_{p,1} + H_{p,2} + H_{p,3} \qquad \text{for} \qquad i \in \{1, 2, 3\},$$

with

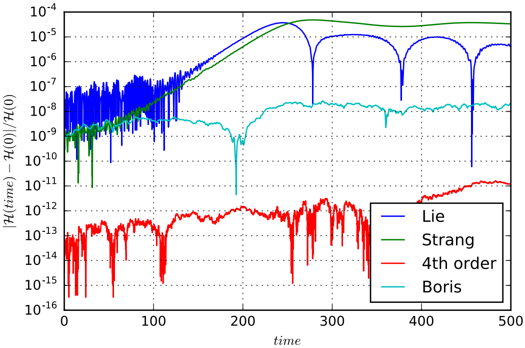
$$H_{p,i} = \frac{1}{2}\, (v_p^i)^T M_p v_p^i.$$

Then an exact solution can be obtained.

GEMPIC: Results for 1d2v Weibel instability

Numerical parameters:

- 100,000 particles.
- 32 grid points.
- $\Delta t = 0.05$.
- Splines of degree 3 and 2.
- Antithetic particle sampling with Sobol numbers.

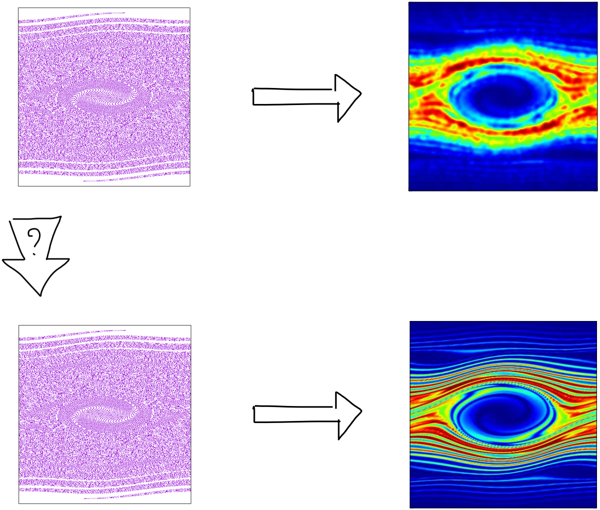


Propagator	total energy	Gauss law
Lie	4.9E-7	8.7E-15
Strang	6.3E-7	1.5E-14
4th order	2.1E-13	3.9E-14
Boris	3.4E-10	1.0E-4

Problems with PIC codes

- Particles follow physical flow. In principle nice but parts of phase space can become over or undersampled.
- Improved with control variate method in tokamaks but still very noisy
- Long time nonlinear simulations not accurate without remapping or resampling:
- Two options:
  1. Remap on phase space grid, partial or full
  2. Particle resampling: (weight smoothing) exchange information between neighbouring particles identified with quad-tree algorithm. Possibly remove or create particles keep global conservation laws within particle group (mass, momentum, energy)

Better remapping by taking into account local flow around particle

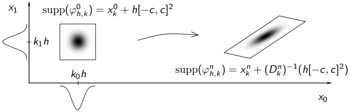


Polynomial approximation of particle shape transportation

- Order 0 (shape stays constant): Standard PIC deposition on phase space grid:

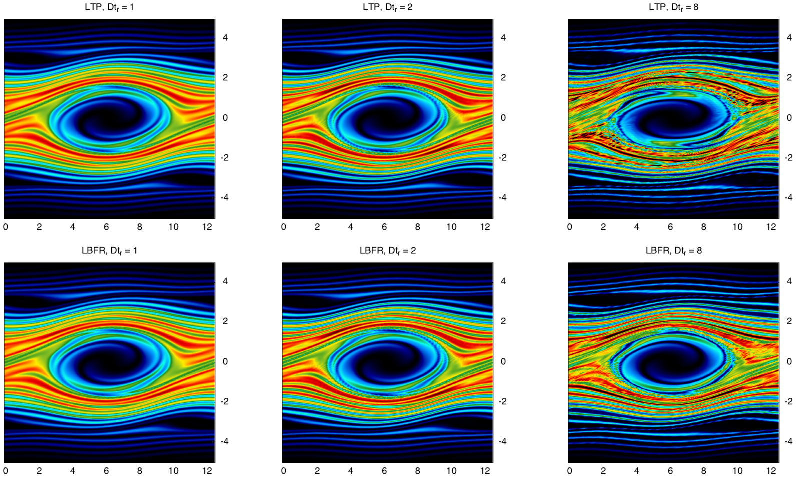
$$f_h(x_i) = \sum_k w_k S(x_i - x_k^n)$$

- Order 1: linearly transported particles (M. Campos Pinto, ES, A. Friedman, S. Lund, D. Grote JCP 14)
  $f_h(x_i) = \sum_k w_k S(D_k^n(x_i - x_k^n))$ $\quad D_k^n$ Jacobian of flow for particle x_k computed using neighbouring particles starting from uniform grid.
  More computation but no storing (free!)



- Order 2: (M. Campos Pinto, even better properties)
- Remapping can be less frequent for higher order. However particle stretching make reconstruction less local

What remapping frequency for LTP?



forward remapping (top), estimation of backward flow (BSL) at grid point using neighbouring particles (bottom)

Outline

An short overview of magnetic fusion models

Kinetic or gyrokinetic model

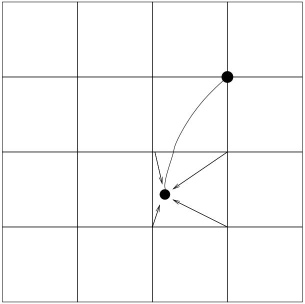
The Geometric Particle-In-Cell method (GEMPIC) (M. Kraus, K. Kormann, P. Morrison, ES)

Semi-Lagrangian Vlasov solvers

Implementation on a sparse grid

Implementation with hierarchical tensor decomposition

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 $\rightarrow$ High order interpolation needed

- Typical interpolation schemes.
  - Cubic spline (Cheng-Knorr)
  - Cubic Hermite with derivative transport (Nakamura-Yabe)
- For Vlasov-Poisson (because of separable Hamiltonian) origin of characteristics can be obtained explicitly with operator splitting.
- Recent hamiltonian splitting method also for Vlasov-Maxwell (Crouseilles et al. 2015).

Split-semi-Lagrangian scheme for Vlasov-Poisson

Given f^n and E^n at time t_n, we compute f^{n+1} at time $t_n + \Delta t$ as follows:

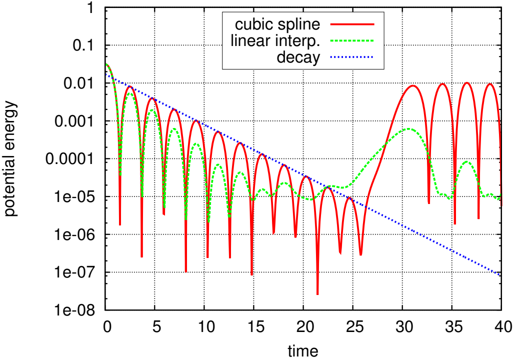
1. Solve $f_t - E^n f_v = 0$ on **half time step**:
   $f^{n,*}(x_i, v_j) = f^n\left(x_i, v_j + E_i^n \frac{\Delta t}{2}\right)$.
2. Solve $f_t + v f_x = 0$ on **full time step**:
   $f^{n,**}(x_i, v_j) = f^{n,*}\left(x_i - v_j \Delta t, v_j\right)$.
3. Compute $\rho(x_i, v_i)$ and solve the **Poisson equation** for E^{n+1}.
4. Solve $f_t - E^{n+1} f_v = 0$ on **half time step**:
   $f^{n+1}(x_i, v_j) = f^{n,**}\left(x_i, v_j + E_i^{n+1} \frac{\Delta t}{2}\right)$.

Constant coefficient advection along given direction at each time step so that the scheme reduces to

- Interpolation along lines
- Integration over velocity to get charge density
- 3D Poisson solver

Need high order interpolation

- **Interpolation** along one-dimensional lines.
  - Displacement constant in some dimensions.
  - One interpolation per grid point.
  - Linear interpolation too diffusive, cubic spline interpolation standard.



Outline

An short overview of magnetic fusion models

Kinetic or gyrokinetic model

The Geometric Particle-In-Cell method (GEMPIC) (M. Kraus, K. Kormann, P. Morrison, ES)

Semi-Lagrangian Vlasov solvers

Implementation on a sparse grid

Implementation with hierarchical tensor decomposition

Semi-Lagrangian solver on sparse grid

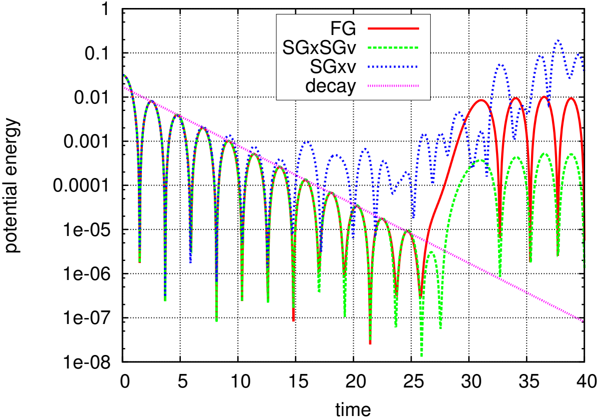
- **Two variants**: sparse grid in 3 dimensions (SGxv) or tensor product of sparse grid in x and v (SGxSGv)
- **Poisson solver**: Pseudo-spectral solver with sparse grid FFT[1]
- **Integration**: Sparse grid trapezoidal rule
- **Interpolation**: Use higher-order sparse grid[2] and exploit structure in interpolation
  - Nodal representation along the dimension with displacement.
  - Nodal one-dimensional interpolation problem can be solved by any interpolator $\rightarrow$ use for instance **cubic spline** interpolation.
  - Use high-order (cubic) sparse grid dehierarchization along the other dimensions.

[1]K. Hallatschek, Numer. Math. 93, 1992

[2]H.-J. Bungartz, Habilitation thesis, 1998

Test case: Landau damping in 4D

- FG: $32^4 = 1024 \times 1024$.
- SGxSGv: $L_x = 5$, $L_v = 7$ (modified) $\to 112 \times 1024$.
- SGxv: $L = 10$, $M = 66304$, i.e. 6% of memory.



Multiplicative δf method

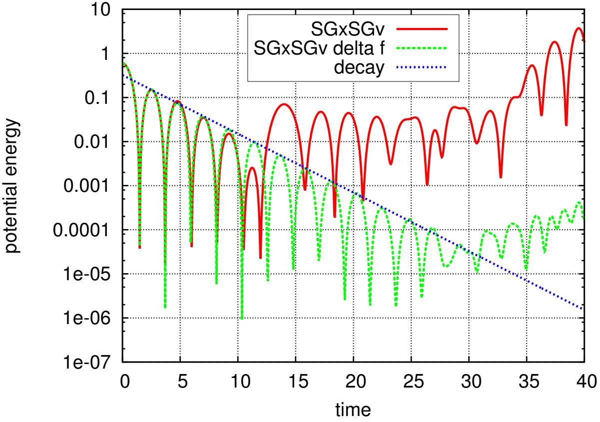
Problem: Poor interpolation of Gaussians on sparse grid.[3]

Ansatz: $f(x,v,t) = g(x,v,t) \cdot h(v)$

- **Initial splitting**: $g(x,v,t) = 1 + 0.01\cos(kx)$, $h(v) = \exp(-0.5v^2)$.
- x-**advection**: $f(x - \Delta t v, v, t) = g(x - \Delta v, v, t) \cdot h(v)$.
- v-**advection**:
  $f(x, v + E(x)\Delta t, t) = g(x, v + E(x)\Delta t, t) h(v + E(x)\Delta t) =$
  $g(x, v + E(x)\Delta t, t) \exp(-0.5\Delta t^2 E(x)^2 - v\Delta t E(x)) \cdot h(v)$
- **Computation of** ρ: Analytically compute $\int h(v)\varphi_{l,k}(v)\,\mathrm{d}v$.

[3] D. Pflüger, PhD Thesis, 2010

Landau damping in 6D



Remarks on sparse grid implementation

- Good compression rates for Landau Damping (0.002 in 6D)
- Complexity can be improved by exploiting structure, mixed interpolation improves accuracy.
- SGxSGv variant more suited than one single sparse grid (better resolution, efficient interpolation and parallelization).
- Linear complexity is achieved for 1D mixed interpolation on fixed sparse grid. Will be lost in any other context, e.g. nD interpolation or adaptive sparse grids.
- Multiplicative δf-method improves accuracy.
- Diffusion necessary for stabilisation.

Outline

Tensor representation

Consider a typical initial distribution function

$$f(\mathbf{x},\mathbf{v}) = \big(1+\varepsilon\cos(k_\perp x_1)\cos(k_\parallel x_3)\big)\exp(-(v_1^2+v_2^2+v_3^2)/2)/(2\pi)^{3/2}$$

- Data stored on $N\times N\times N\times N\times N\times N$ mesh: N^6
- Information can be stored in $9N$ using

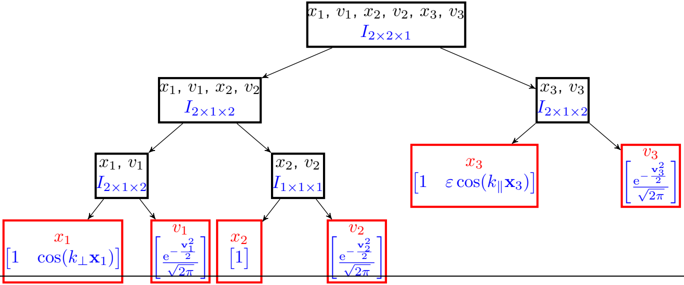
$$f(\mathbf{x},\mathbf{v}) = \big[1\otimes 1\otimes 1+\varepsilon\cos(k_\perp x_1)\otimes 1\otimes\cos(k_\parallel x_3)\big] \otimes\exp(-v_1^2/2)/\sqrt{2\pi}\otimes\exp(-v_2^2/2)/\sqrt{2\pi}\otimes\exp(-v_3^2/2)/\sqrt{2\pi}$$

- Tensor representation offers huge gain in needed memory.
- Can this also be used for more general function?

Hierarchical Tucker format[a]

[a] Hackbusch, Kühn, J. Fourier Anal. Appl. 15, 2009

- Group indices into two sets to get matrix: matricisation.
- Compress based on SVD: $M = U \cdot \Sigma \cdot V^\top$. Only r largest singular values kept.
- This can be done hierarchically in a binary tree.
- Only store kernels
  - $Q_i \in \mathbb{R}^{N_i \times r_i}$ for leaf nodes (number of mesh points N_i, ranks r_i).
  - $B_i \in \mathbb{R}^{r_\text{left} \times r_\text{right} \times r_i}$ for non-leaf nodes.
  - Hierarchical Tucker representation of initial value



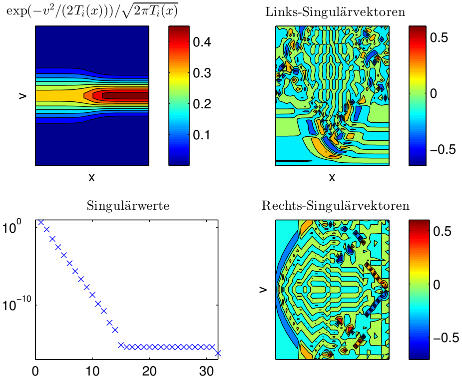
Compression example in 2D

Consider:

$$f(x,v) = \mathrm{e}^{-v^2/(2T_i(x))}/(2\pi T_i(x))^{1/2}$$

No exact representation as sum of Kronecker Products → Approximate representation as compressed Tucker decomposition.

Principle of compression algorithm: Perform SVD at each level of hierarchical Tucker decomposition and keep only terms corresponding to highest singular values.

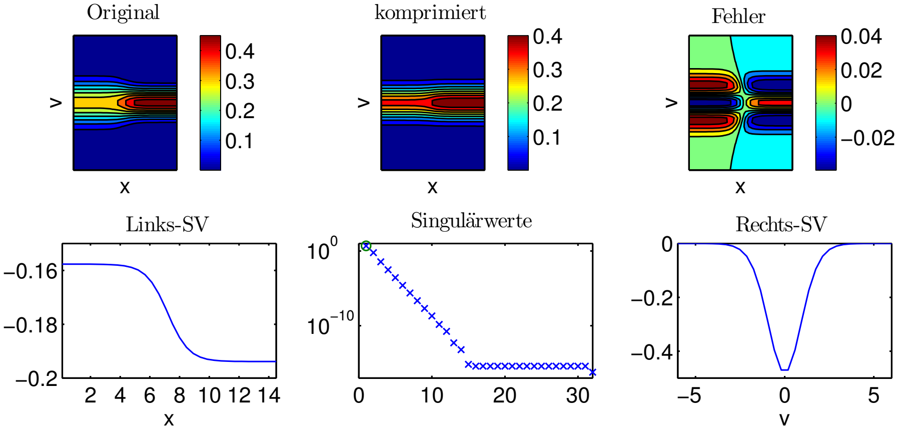


Compression example in 2D

Consider:

$$f(x, v) = e^{-v^2/(2T_i(x))}/(2\pi T_i(x))^{1/2}$$

Representation with one singular value:

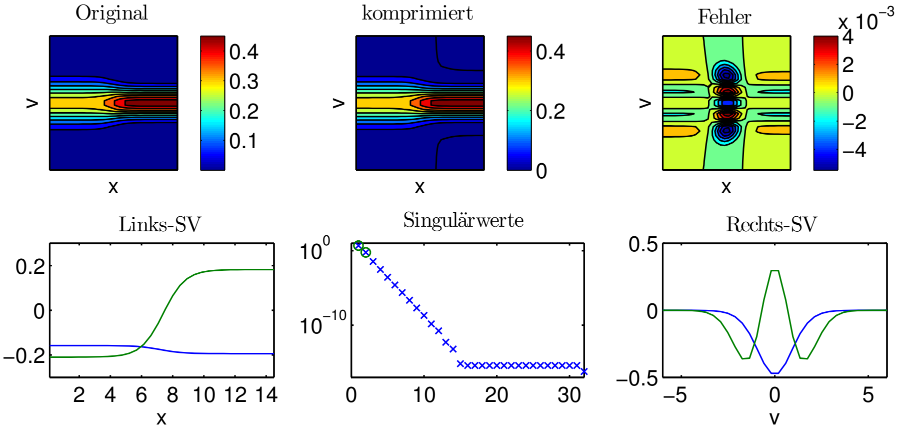


Compression example in 2D

Consider:

$$f(x,v) = e^{-v^2/(2T_i(x))}/(2\pi T_i(x))^{1/2}$$

Representation with two singular values:

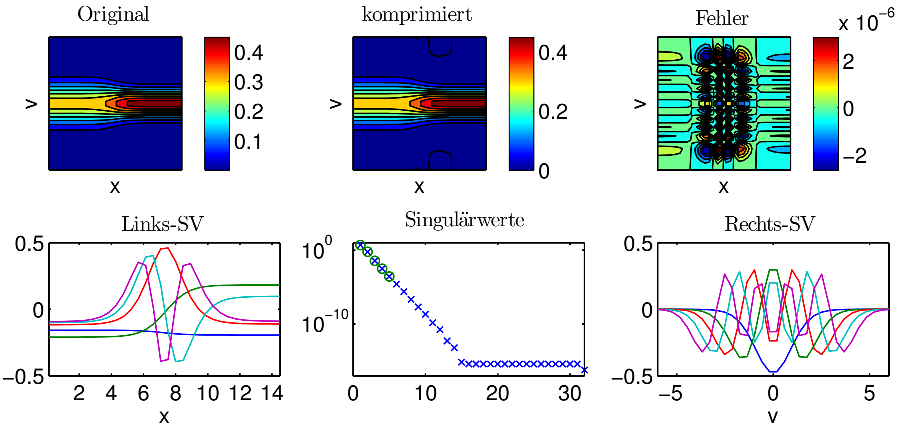


Compression example in 2D

Consider:

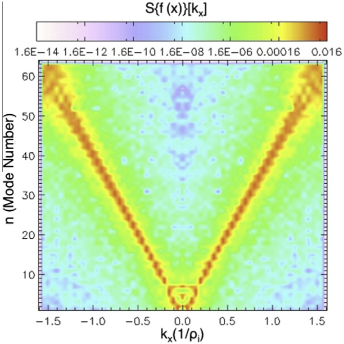
$$f(x,v) = \mathrm{e}^{-v^2/(2T_i(x))}/(2\pi T_i(x))^{1/2}$$

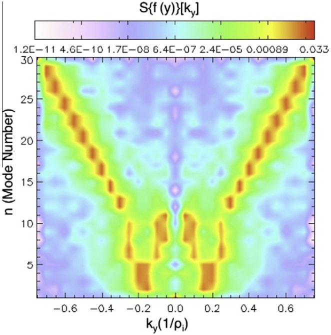
Representation with five singular values:



HOSVD of GENE output

- HOSVD compression of turbulence GENE simulation performed in Hatch, del-Castillo-Negrete, Terry, J. Comput. Phys. 231(11), 2012
- HOSVDx and HOSVDy show clear dominating modes





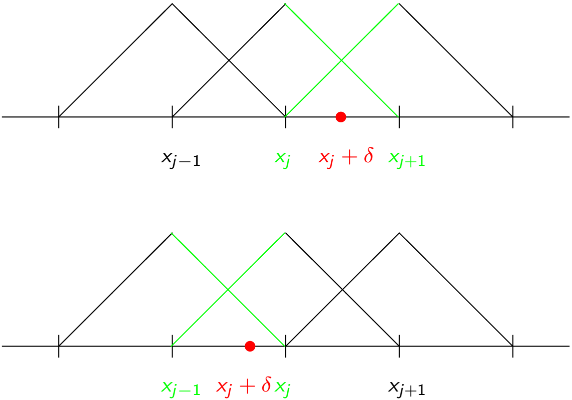
- Compression to 4% of initial size achieved at 10% error.

Operations for Semi-Lagrangian method in Hierarchical Tucker (HT) format

- Our aim: implement algorithm with data in HT format.
- Semi-Lagrangian method with 1D splitting. Essential piece is 1D interpolation.
- Use 5 points Lagrange interpolation. Stencil identical on whole line. Assemble sparse tensor and compress.
- Basic operations (matrix-vector product, addition, inner product,...) are defined in HT format but generally increase the rank more than necessary $\rightarrow$ Recompress data with SVD.
- SVD needed to find a suitable compression in hierarchical Tucker format. Good algorithms exist for shared memory. Might be expensive for very large problems. Alternative algorithms exist which might be more efficient on many nodes.

SLHT algorithm: linear interpolation in x-direction

Condition: $|\delta| = |-\Delta t v_k| \leq \Delta x$ to limit stencil.



$$f^*(x_j, v_k) = f^n(x_j - \Delta t v_k, v_k) \approx \max(0, \Delta t v_k) f^n(x_{j-1}, v_k) + (1 - \text{abs}(-\Delta t v_k))\, f^n(x_j, v_k) + \max(0, -\Delta t v_k) f^n(x_{j+1}, v_k)$$

SLHT algorithm: linear interpolation in x-direction

Condition: $|\delta| = |-\Delta t v_k| \leq \Delta x$ to limit stencil.

$$
\begin{aligned}
f^*(x_j, v_k) = f^n(x_j - \Delta t v_k, v_k) &\approx \max(0, \Delta t v_k) f^n(x_{j-1}, v_k) \\
&+ (1 - \mathrm{abs}(-\Delta t v_k))\, f^n(x_j, v_k) + \max(0, -\Delta t v_k) f^n(x_{j+1}, v_k)
\end{aligned}
$$

In HT format

$$
\begin{aligned}
&Q_1^*(x_j, \alpha) B(\alpha, \beta) Q_2^*(\beta, v_k) = \\
&\quad \approx Q_1^{(n)}(x_{j-1}, \alpha) B(\alpha, \beta) \left(Q_2^{(n)}(\beta, v_k) \max(0, \Delta t v_k) \right) \\
&\quad + Q_1^{(n)}(x_j, \alpha) B(\alpha, \beta) \left(Q_2^{(n)}(\beta, v_k)\, (1 - \mathrm{abs}(-\Delta t v_k)) \right) \\
&\quad + Q_1^{(n)}(x_{j+1}, \alpha) B(\alpha, \beta) \left(Q_2^{(n)}(\beta, v_k) \max(0, -\Delta t v_k) \right).
\end{aligned}
$$

Involves only simple operations on existing HT decomposition.

Some comments on the complexity

- Ranks multiply for matrix-vector product and sum up for addition of HT tensors.
- Complexity dominated by SVD to recompress the data.
- Complexity of HT truncation: $\mathcal{O}(dnr^2 + dr^4)$.
- Complexity can be reduced when using special structure and applying intermediate rounding.

Test on landau damping

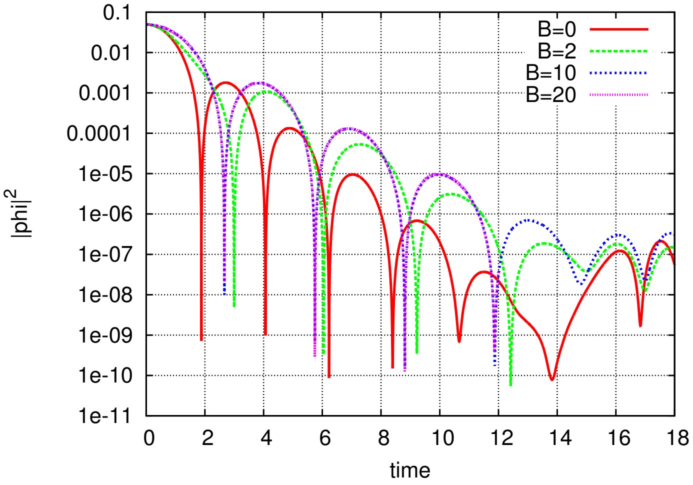
- Verify algorithm and check performance on a Landau damping problem in a magnetized plasma
- Initial condition:

$$f_0(\mathbf{x}, \mathbf{v}) = \frac{1}{\pi^{3/2} v_{T_i}^3} \mathrm{e}^{-\frac{\|\mathbf{v}\|_2^2}{v_{T_i}^2}} \left(1 + \varepsilon \cos(k_\perp x_1) \cos(k_\| x_3)\right)$$

- Parameters:

$$v_{T_i} = \sqrt{2}, \quad k_\perp = k_\| = 0.5, \quad \varepsilon = 0.01$$

Time evolution of $\|\phi\|_2^2$

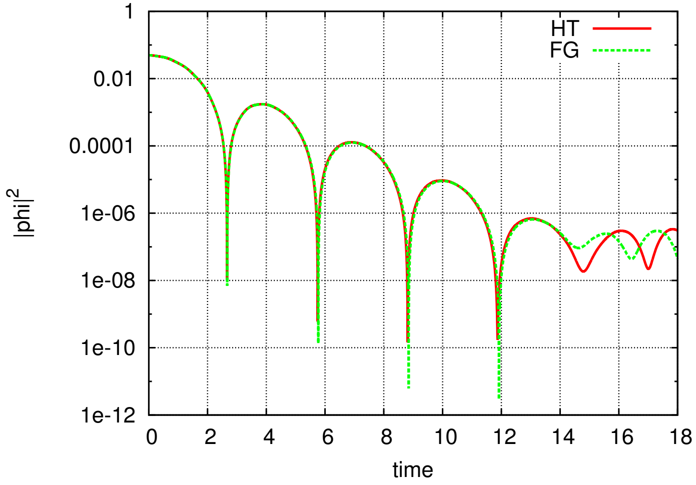


Comparison dispersion relation

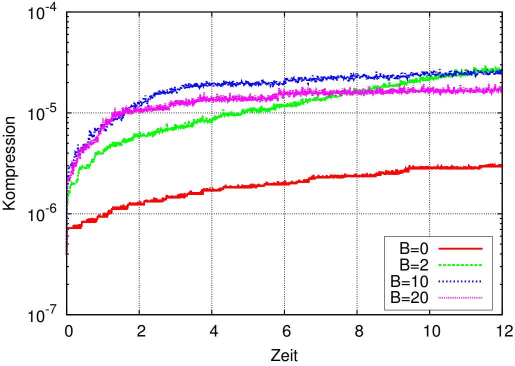
Table: Value of ω for $v_{T_i} = \sqrt{2}$, $k_\perp = k_\| = 0.5$.

B	ω (dispersion)	ω (simulation)
0		1.4455 − 0.6023i
0.1	1.4444 − 0.6042i	1.4477 − 0.6074i
0.8	1.4420 − 0.7306i	1.4215 − 0.7607i
2	0.9851 − 0.4482i	0.9931 − 0.4577i
4	1.0162 − 0.4320i	1.0175 − 0.4348i
10	1.0219 − 0.4267i	1.0287 − 0.4265i
20	1.0228 − 0.4258i	1.0283 − 0.4263i
50	1.0229 − 0.4257i	1.0290 − 0.4260i

Comparison to full grid ($B = 10$)



Comparison to full grid: Compression

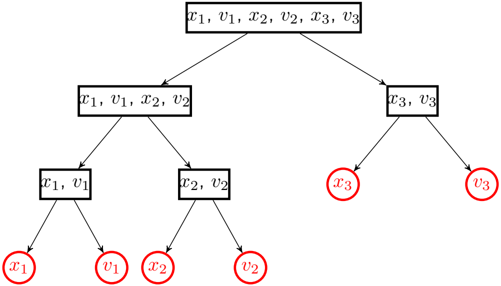


Ordering of variables in HT matters

Where is the coupling?

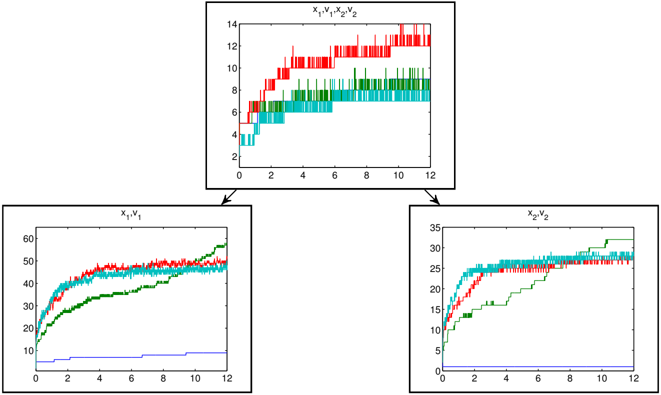
- Coupling within pairs (x_i, v_i) in each x_i-advection step.
- Coupling between $(v_i, \mathbf{x})$ in each v_i-advection step.

Optimized ordering:



Dependence on B of compression

Compression rates of each kernel for different values of B



Color code: blue $B = 0$, green $B = 2$, red $B = 10$, cyan $B = 20$

Computing times

Hierarchical Tucker code:
Hardware: Intel Ivy Bridge notebook processor 3.0 GHz

B	CPU time [s]
0	66
2	1553
10	2376

Full grid code ($B = 10$):
Hardware: Hydra cluster of Max–Planck–Gesellschaft, Intel Sandy Bridge-EP nodes with 16 cores 2.6 GHz

# nodes	wall clock time [h]	total CPU time [h]
1	20:56	327
2	11:48	358

Conclusion

- Structure preserving PIC method has very nice conservation properties and long time stability. But noisy and slow convergence.
- Projection (at least from time to time) on phase space grid helps
- Improve convergence properties when information on local flow, from neighgoring particles is taken into account.
- Hierarchical Tucker very efficient for semi-Lagrangian method.
- Huge gain in memory for simple 6D Vlasov problems. Can be run on laptop.
- Hierarchical Tucker seems better suited to semi-Lagrangian method for Vlasov than sparse grids.
- Next step is to tackle more realistic problems that fit on single node using shared memory SVD, and see how good compression is.