Comparison of Numerical Flow Iteration to Lagrangian and Semi-Lagrangian Schemes for the Vlasov System

Rostislav-Paul Wilhelm
Jan Eifert
Manuel Torrilhon

November 28, 2023

Vlasov–Poisson equation

The Vlasov–Poisson equation

Kinetic model for collective electron behaviour

$$\partial_t f + \underbrace{\frac{p}{m\gamma}\partial_x f}_{\text{relativistic terms}} + \underbrace{qE\partial_p f}_{\substack{\text{Neglect magnetic} \\ \text{field.}}} = \underbrace{0}_{\substack{\text{Neglect collisions:} \\ Q(f,f)=0.}}$$

The Poisson equation for E reads:

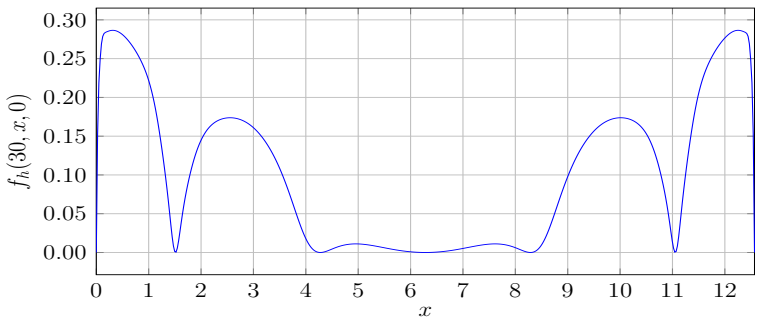
$$\begin{aligned} E(t,x) &= -\nabla_x \varphi(t,x), \\ \Delta_x \varphi(t,x) &= -\rho(t,x) \\ \rho(t,x) &= \int_{\mathbb{R}^d} f_{ion}(t,x,v)\mathrm{d}v - \int_{\mathbb{R}^d} f_{electron}(t,x,v)\mathrm{d}v. \end{aligned}$$

Problems with solving Vlasov equation numerically

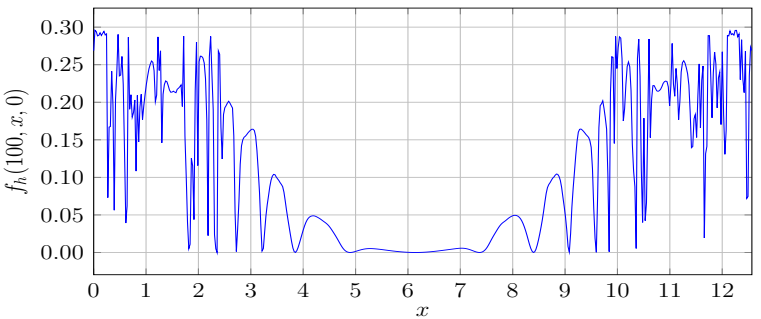
$$\partial_t f + \frac{p}{m\gamma}\partial_x f + qE\partial_p f = 0$$

$$f : \mathbb{R}_{\geq 0} \times \mathbb{T}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$$

- **High-dimensional** equation: Full Vlasov equation for 1 species is a 7-dimensional PDE.
  - ➢ Grid-based solvers suffer from extensive memory-usage.



$t = 30$

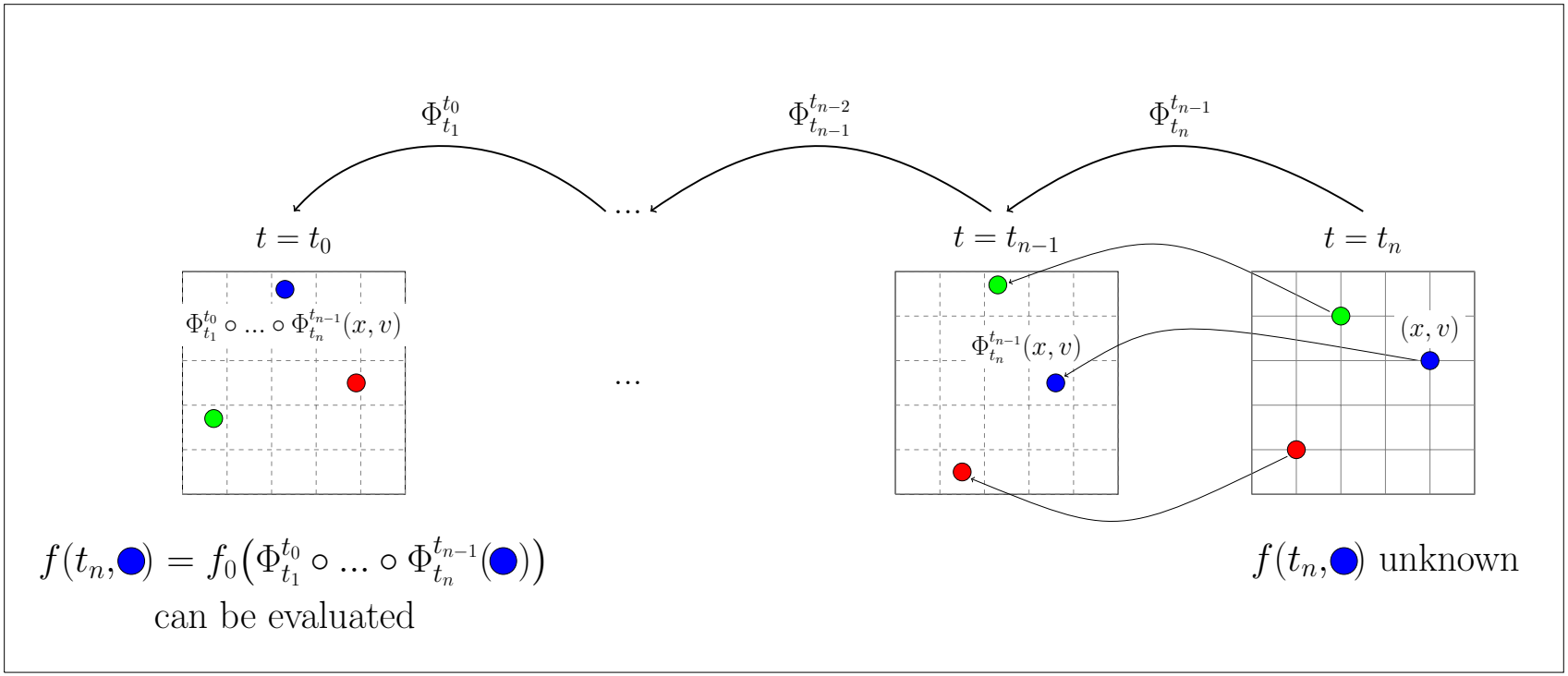


$t = 100$

- **Lack of diffusion** in the model leads to strong filamentation in the solution for long-times.
  - ➢ Steep gradients ⇒ Practically impossible to resolve.
  - ➢ DSMC/SPH/PIC require high resolution and remeshing.

NuFI

Motivation



Note:

$$\partial_t f + \left(\begin{pmatrix} v \\ -E \end{pmatrix} \cdot \nabla_{(x,v)} \right) f = 0 \quad \Rightarrow \quad f(t, x, v) = f_0(\Phi_t^0(x, v))$$

Where:

$$\frac{\mathrm{d}}{\mathrm{d}t} \Phi_t^s(x, v) = \begin{pmatrix} -v \\ E(t, x) \end{pmatrix} \qquad \Phi_s^s(x, v) = (x, v)$$

Approximating the phase-flow

Iterative approximation of $\Phi_{t_k}^{t_{k-1}}$ via **symplectic Euler**:

$$\begin{aligned} x_{k-1} &:= x_k - \gamma \cdot \Delta t \cdot v_k, \\ v_{k-1} &:= v_k + \Delta t \cdot \frac{E\left(t_{k-1}, x_{k-1}\right)}{m}, \end{aligned}$$

Electric field is known at **previous time-step**!

For **nonlinear case**, $t = t_k$:

1. Compute ρ_k using numerical quadrature.
2. Compute electric potential φ_k from ρ_k using finite differences.
3. Save coefficients of φ_k in B-Spline basis $\rightarrow$ incorporate Dirichlet boundary conditions.

Repeat for $t_k \rightarrow t_{k+1}$!

Memory usage

No need to save f!

- Saving φ_h.
- Shift workload from "memory access" to "computation on the fly":
  - ➢ High flop/Byte ratio.
  - ➢ Cache can store coefficients of φ_h for several time-steps.

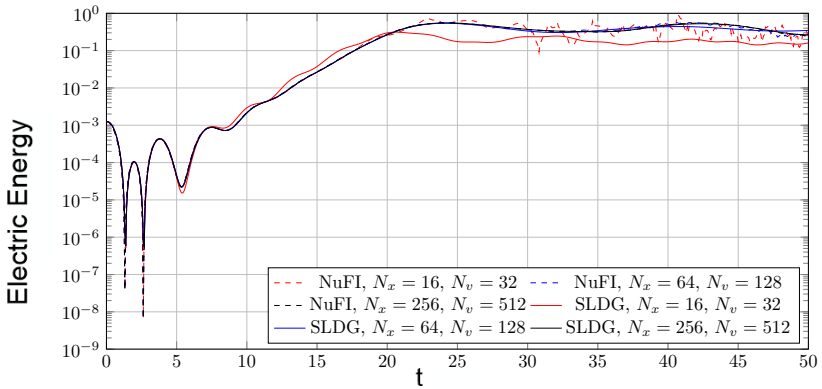
Memory Requirement **Grid** $= N_x^d \times N_v^d \times$ size of one floating point value
Memory Requirement **NuFI** $= n \times N_x^d \times$ size of one floating point value

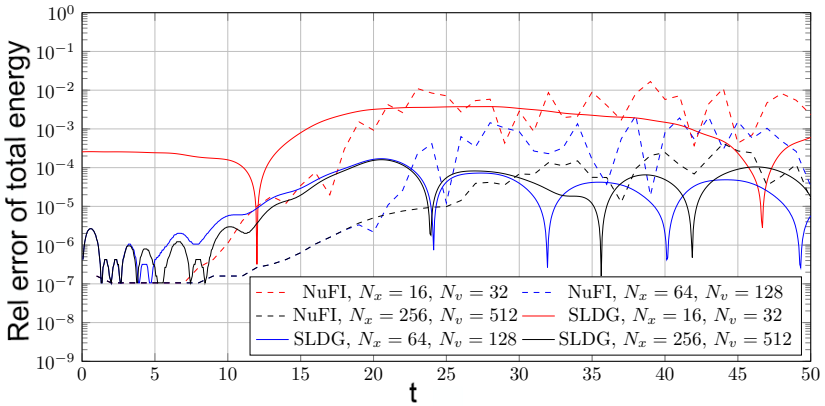
Where n number of time-steps: $n \ll N_v^d$.

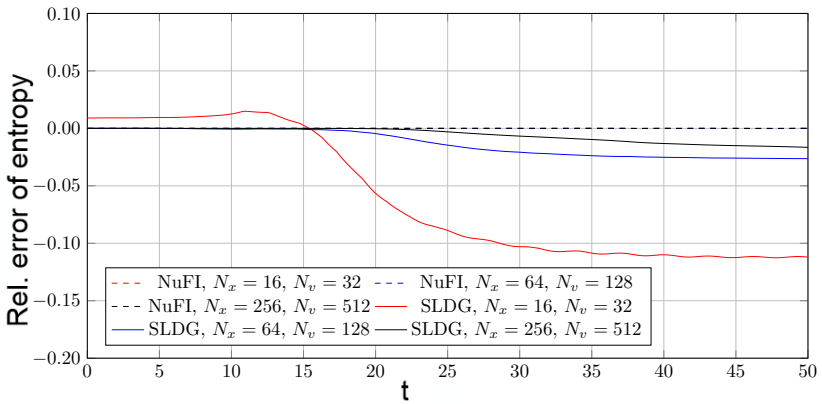
d	$N_x = N_v$	Memory of φ_h per step	Memory for storing f
2	128	0.125 MiB	$\approx 2.0 \cdot 10^3$ MiB
2	256	0.5 MiB	$\approx 3.3 \cdot 10^4$ MiB
2	512	2 MiB	$\approx 5.2 \cdot 10^5$ MiB
3	128	0.015 GiB	$\approx 3.3 \cdot 10^4$ GiB
3	256	0.125 GiB	$\approx 2.1 \cdot 10^6$ GiB
3	512	1 GiB	$\approx 1.3 \cdot 10^8$ GiB

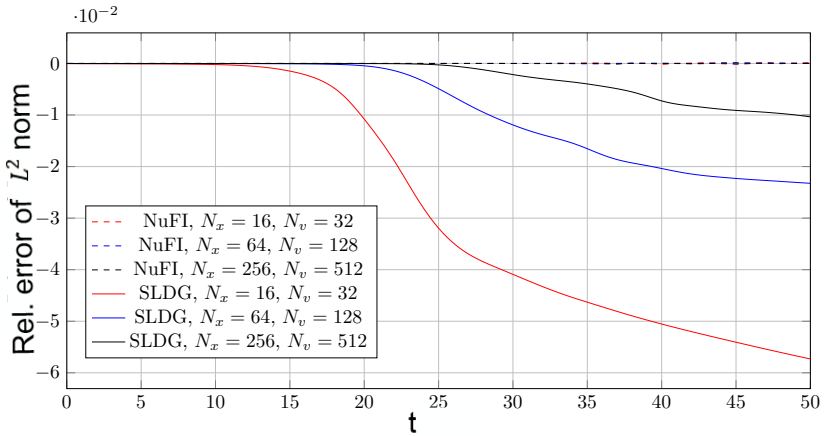
Conservation properties of NuFI

Comparison of different conservation properties over time between NuFI and SLDG.





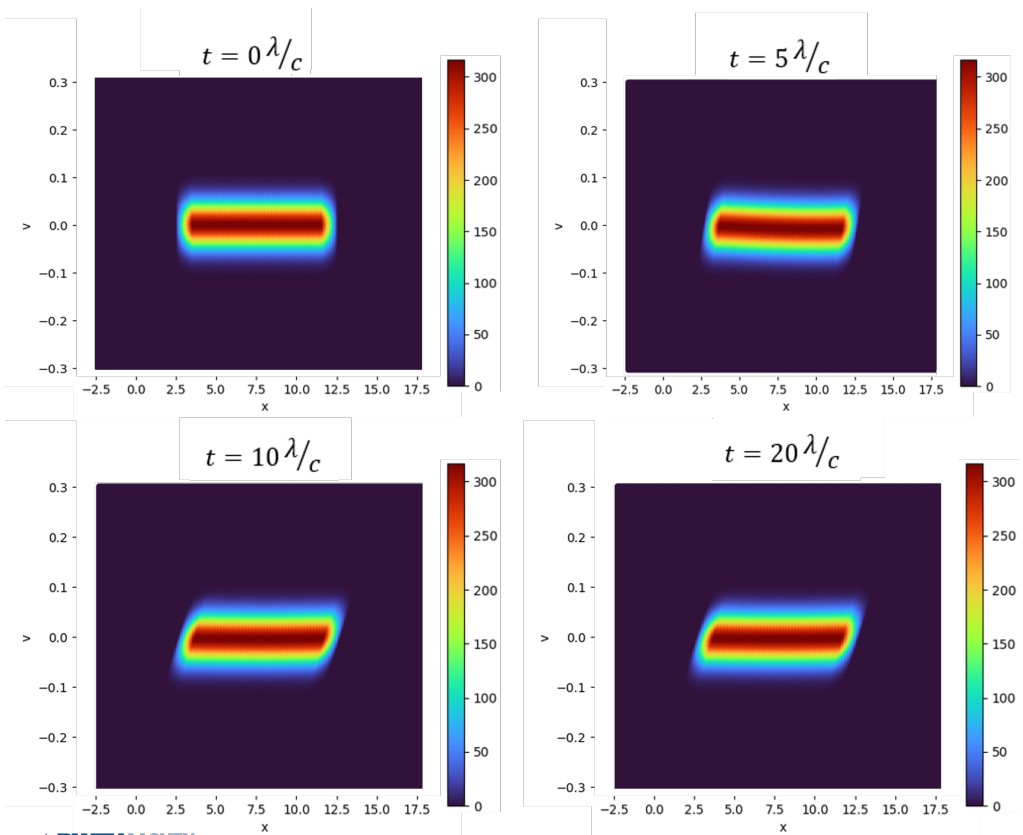




Numerical experiments

Laser-driven shock

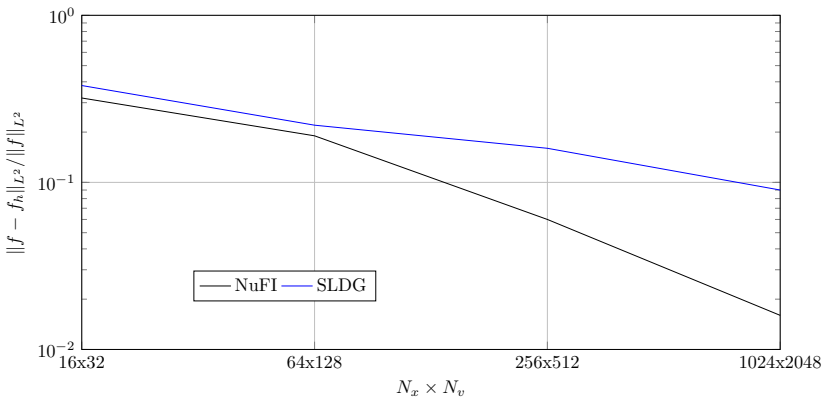
$$f_0(x, v, T, m) = p_0(x) \cdot \frac{1}{\sqrt{2\pi \frac{T}{m}}} e^{-\frac{v^2 m}{2T}}, \quad p_0(x) = \begin{cases} 2n_c x/\lambda - 6n_c, & \text{if } 3\lambda \leq x < 4\lambda, \\ 2n_c, & \text{if } 4\lambda \leq x < 12\lambda, \\ -2n_c x/\lambda + 26n_c, & \text{if } 12\lambda \leq x < 13\lambda, \\ 0, & \text{otherwise.} \end{cases}$$

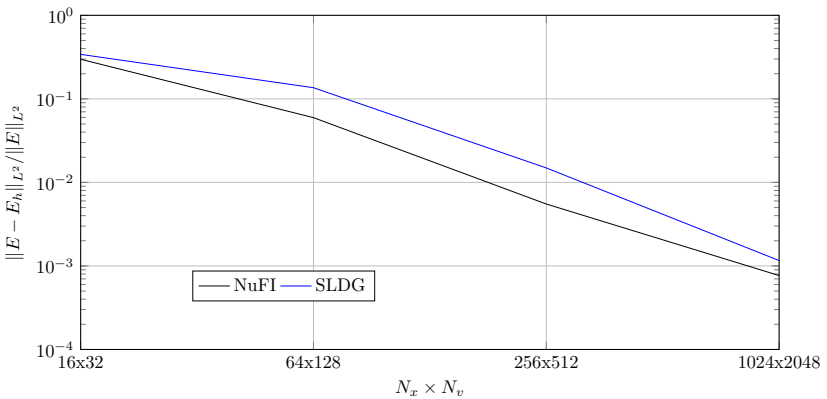


- Ion movement
- Reflecting boundary conditions for f
- Dirichlet boundary conditions for φ

Errors

Two Stream 1d Relative L^2 errors in f and E:

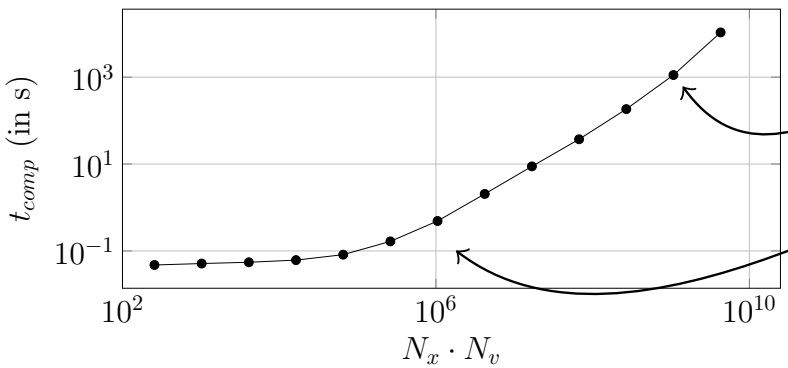


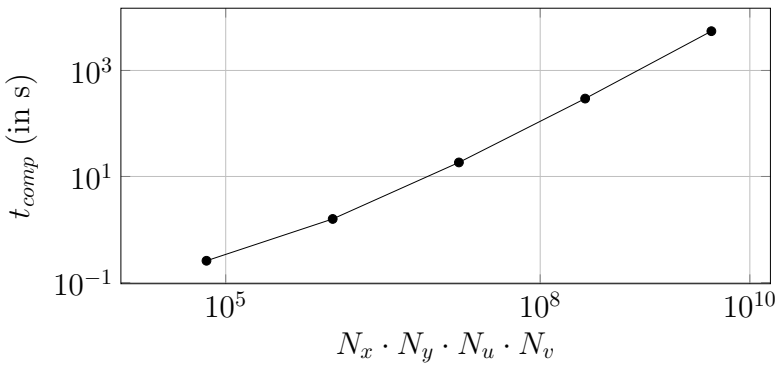


$\Rightarrow$ Decreasing error in f and E for increasing resolution
$\Rightarrow$ NuFI results in faster convergence for E and, in particular, f [Einkemmer, 2015]

Computational complexity and scaling

Two-stream benchmark: $d = 1$ (left) & $d = 2$ (right). $N_T = 481$ time-steps.

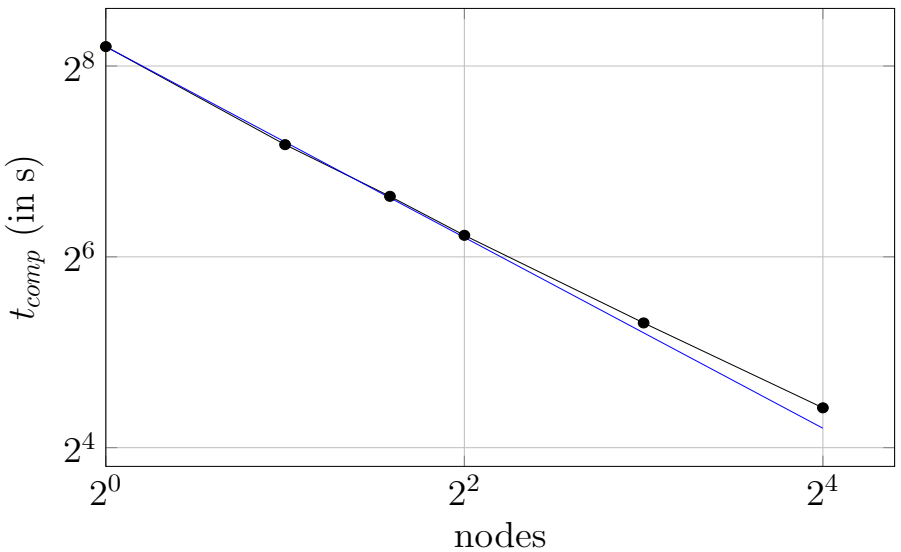




Until here L1-Cache can contain coefficient vector.

GPU's are not fully used yet.

$\Rightarrow$ Computational complexity **linear** in number of quadrature points!



$\Rightarrow$ **Strong scaling.**

Conclusion

Summary

Conclusion:

- All DOF in phase-space are independent of each other!
  - ➢ Embarrassingly parallel algorithm.
- Very low memory usage.
  - ➢ **Never save f!**
- Comes at cost of increasing complexity with number of time-steps.
- Exact conservation of properties.
- Can naturally incorporate any boundary conditions.

Future research:

- Enabling restart of simulation & starting from experimental data.

Wilhelm, R.-P., ACoM RWTH Aachen, wilhelm@acom.rwth-aachen.de
Eifert, J., ACoM RWTH Aachen, jan.eifert@rwth-aachen.de