Performance comparison of the Numerical Flow Iteration to Lagrangian and Semi-Lagrangian approaches for solving the Vlasov equation in the six-dimensional phase-space

Rostislav-Paul Wilhelm[1], Fabian Orland[2], Manuel Torrilhon[1]

[1]Institute for Applied and Computational Mathematics, RWTH Aachen University
[2]IT Center Aachen, RWTH Aachen University

NHR for Computational Engineering Science

Introduction

High-temperature plasmas require **modelling via the Vlasov equation arising from kinetic theory** [2]. Consider the Vlasov–Poisson system to model the dynamics of the electron probability distribution $f = (t, x, v)$ ($t \in \mathbb{R}$, $x \in \mathbb{R}^d$ and $v \in \mathbb{R}^d$ with $d \in \{1, 2, 3\}$):

$$\partial_t f + v \cdot \nabla_x f - E \cdot \nabla_v f = 0 \tag{1}$$

$$E = -\nabla_x \varphi, \tag{2}$$

$$-\Delta_x \varphi = \rho = \int_{\mathbb{R}^d} f(t, x, v)\,\mathrm{dv}. \tag{3}$$

- **Curse of dimensionality** ⇒ Direct phase-space discretization inefficient.
- Particle-In-Cell (PIC) and Semi-Lagrangian (SL) methods **heavily memory-bound**.
- **Turbulent** model with **filamented** solutions.

NuFI: Backwards Iteration

Algorithm 1: Evaluate f with Störmer–Verlet backwards in time

At $t = t_n$: $x_h^n = x$, $v_h^n = v$ and $i = n$.
while $i > 0$ **do**
  $v_h^{i-\frac{1}{2}} = v_h^i + \frac{\Delta t}{2} E(t_i, x_h^i)$.
  $x_h^{i-1} = x_h^i - \Delta t v_h^{i-\frac{1}{2}}$.
  $v_h^{i-1} = v_h^{i-\frac{1}{2}} + \frac{\Delta t}{2} E(t_{i-1}, x_h^{i-1})$.
  $i = i - 1$.
end
Return $f_0\left(x_h^0, v_h^0\right)$.

- **Trace positions backwards in time to evaluate at initial data f_0 analytically.**

NuFI: Forwards-time loop

An **alternative approach to solving the Vlasov–Poisson system is NuFI** [5]:

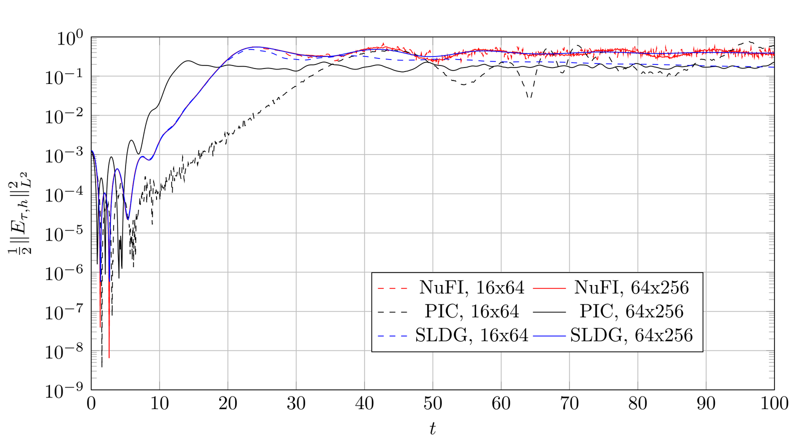
- Indirect evaluation of f through method of characteristics.
- Store electric potential φ instead of f.

Algorithm 2: NuFI for VP with periodic boundaries

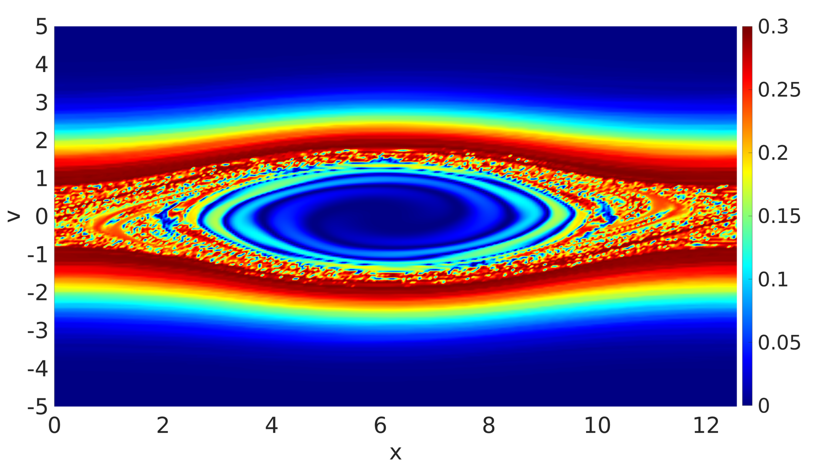
Allocate a array C for the coefficients of φ ($N_t N_x$ floats).
for $n = 0, \ldots, N_t$ **do**
  Compute ρ_n from f_n using Algorithm 1 and the mid-point integration rule.
  Solve Poisson's equation via FFT to obtain φ_n from ρ_n.
  Interpolate φ_n and save the coefficients.
end

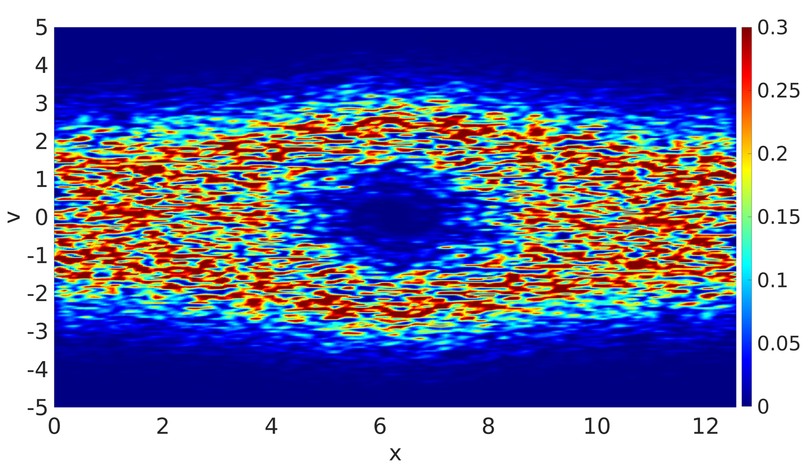
- **Embarassingly parallel algorithm!**

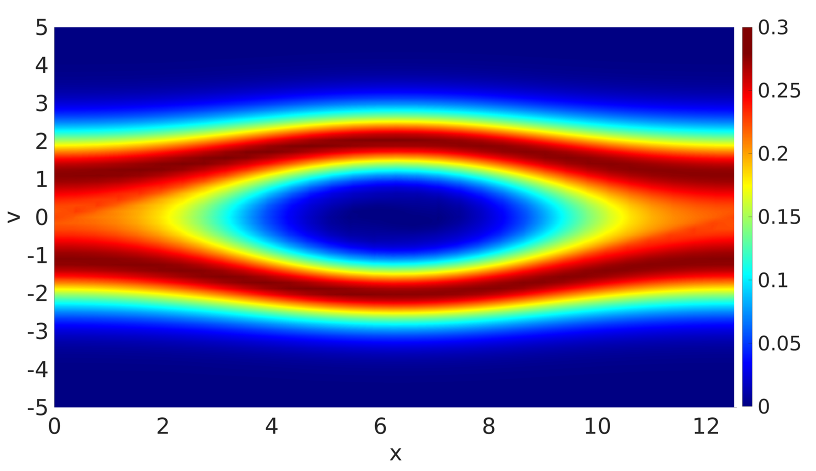
Two stream instability benchmark



(a) Electric Energy over time

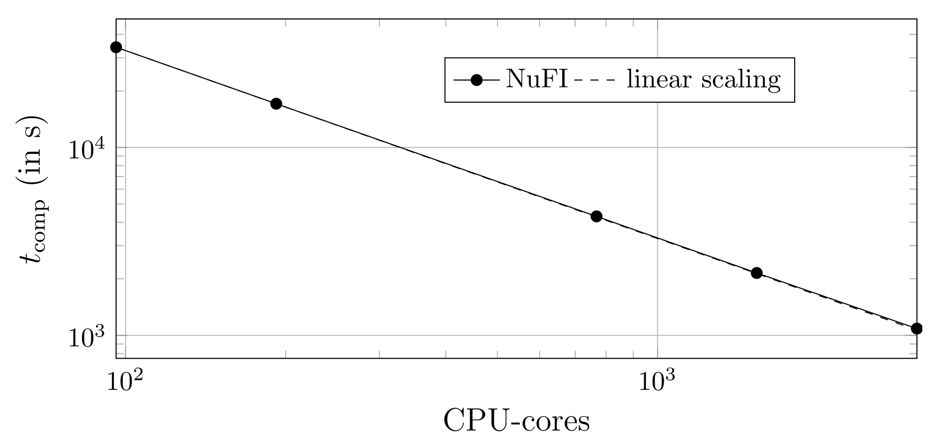

(b) NuFI [5]


(c) PIC [7]

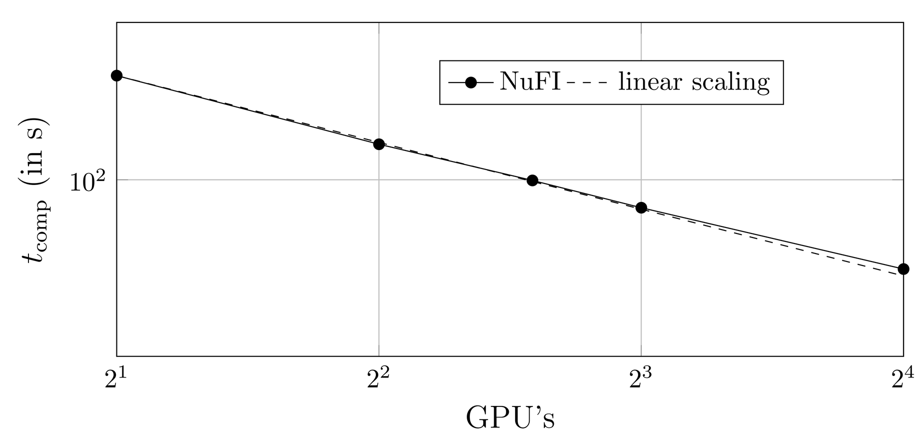

(d) SLDG [3]

⇒ PIC is unable to capture the dynamics and resolve f for long simulation times.
⇒ SLDG is unable to resolve f for long simulation times.
⇒ NuFI is able to resolve the fine structures while reproducing the right dynamics.

Strong scaling of NuFI: CPU vs GPU



(a) Strong Scaling on CPU's: Parallel efficiency $\geq 98.1\%$. Time given for a single time-step in $d = 3$ with $N_x = N_v = 64$ at $n_T = 100$.



(b) Strong Scaling on GPU's: Parallel efficiency $\geq 86.2\%$. Time given for full simulation in $d = 2$ with $N_x = 64$, $N_v = 256$, $n_T = 480$.

- For SeLaLib parallel efficiency reduces to ≈ 50 % for runs with 64 cores with a problem size of 32^6 DoFs (using Intel Xeon Phi nodes on DRACO) [6] .
- Using SLDG parallel efficiency reduces to 50 % with 16 GPU's and to between 33 and 37 % for 64 to 1024 GPU's with a problem size of 40^6 DoFs (using JUWELS with NVidia A100) [4].

Comparing memory complexity

NuFI reduces its memory complexity by a factor $\mathcal{O}(N_v^d / N_t)$ compared to "classical approaches" (e.g. spline-based Semi-Lagrangian solver) via having **quadratic runtime complexity instead of linear**:

- Classical approach:
  ⇒ Comp. complexity: $\mathcal{O}(N_t)$.
  ⇒ Memory complexity: $\mathcal{O}(N_x^d \cdot N_v^d)$.
- NufI:
  ⇒ Comp. complexity: $\mathcal{O}(N_t^2)$.
  ⇒ Memory complexity: $\mathcal{O}(N_t \cdot N_x^d)$.

4-dim. phase-space ($N_t = 1000$):

$N_x = N_v$	NuFI	Classic	Savings
32	7.8 MiB	8 MiB	2.5%
128	0.12 GiB	2 GiB	94 %
512	1.95 GiB	512 GiB	99.62 %

6-dim. phase-space ($N_t = 1000$):

$N_x = N_v$	NuFI	Classic	Savings
8	3.9 MiB	8 MiB	51.25 %
32	0.244 GiB	8 GiB	96.95 %
128	0.015 TiB	32 TiB	99.95 %

- **With NuFI efficient use of cache possible even for large problems**!

Comparison of computational efficiency

	48	96	192	384	768	1536
Global Efficiency	0.9874	1.0005	0.9921	0.9855	0.9551	0.9023
Parallel Efficiency	0.9874	0.987	0.9796	0.9688	0.9489	0.8962
Process_Efficiency	0.9904	0.9907	0.9842	0.9733	0.9525	0.9014
MPI_Communication_Efficiency	1.0	1.0	1.0	0.9999	0.9997	0.9995
Process_Load_Balance	0.9904	0.9907	0.9842	0.9734	0.9528	0.9019
Thread_Efficiency	0.997	0.9963	0.9955	0.9956	0.9965	0.9948
Amdahl_Efficiency	1.0	1.0	1.0	1.0	1.0	1.0
OpenMP_Efficiency	0.997	0.9963	0.9955	0.9956	0.9965	0.9948
Computational_Scaling	1.0	1.0137	1.0127	1.0172	1.0065	1.0067

(a) Strong scaling of NuFI on 1-32 nodes (48 - 1536 cores) on CLAIX-2018.

- Performance analysis using *additive hybrid POP metrics* for NuFI [1]: Shows **parallel efficiency over** 90 % for all cases, equal distribution of work and negligible communication cost.

Break-Even point

- **There exists a break-even point n_B until which NuFI is faster and after which a classical approach becomes faster.** It depends on:
  ⇒ Memory bandwidth & Flop/s of hardware,
  ⇒ Dimension and degrees of freedom.

References

- POP Standard Metrics for Performance Analysis of Hybrid Parallel Applications. https://pop-coe.eu/. Accessed: 2023-01-10.
- F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer Cham, 2015.
- L. Einkemmer. High performance computing aspects of a dimension independent semi-Lagrangian discontinuous Galerkin code. *Comput. Phys. Commun.*, 202:326–336, 2016.
- L. Einkemmer and A. Moriggl. Semi-Lagrangian 4d, 5d, and 6d kinetic plasma simulation on large-scale GPU-equipped supercomputers. *The International Journal of High Performance Computing Applications*, 37(2):180–196, 2023.
- M. Kirchhart and R.-P. Wilhelm. The Numerical Flow Iteration for the Vlasov-Poisson equation. *arXiv*, 2023.
- K. Kormann, K. Reuter, and M. Rampp. A massively parallel semi-Lagrangian solver for the six-dimensional Vlasov–Poisson equation. *The International Journal of High Performance Computing Applications*, 33(5):924–947, 2019.
- B. Wang, G. H. Miller, and P. Colella. A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas. *SIAM J. Sci. Comput.*, 33(6):3509–3537, 12 2011.

NuFI preprint

POP metrics