NuFl: The numerical flow iteration for the Vlasov–Poisson equation

Rostislav-Paul Wilhelm
Matthias Kirchhart

December 12, 2022
1 Vlasov–Poisson equation

2 NuFl

3 Numerical experiments

4 Conclusion
Vlasov–Poisson equation
The Vlasov–Poisson equation

**Kinetic model** for collective electron behaviour

\[
\partial_t f + \mathbf{v} \cdot \nabla_x f - \mathbf{E} \cdot \nabla_v f = 0
\]

Neglect magnetic field.

Neglect collisions: \( Q(f, f) = 0 \).

The Poisson equation for \( E \) reads:

\[
E(t, x) = -\nabla_x \varphi(t, x), \\
\Delta_x \varphi(t, x) = -\rho(t, x) \\
\rho(t, x) = 1 - \int_{\mathbb{R}^d} f(t, x, \mathbf{v}) d\mathbf{v}.
\]
Problems with solving Vlasov equation numerically

\[
\partial_t f + \mathbf{v} \cdot \nabla_x f - E \cdot \nabla_v f = 0
\]

\[f : \mathbb{R}_{\geq 0} \times \mathbb{T}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}\]

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

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

NuFl: The numerical flow iteration for the Vlasov–Poisson equation
Motivation

\[ f(t_n, \bullet) = f_0(\Phi_{t_1}^{t_0} \circ \ldots \circ \Phi_{t_n}^{t_{n-1}}(x, v)) \]
can be evaluated

Note:

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

Where:

\[ \frac{d}{dt} \Phi^s_t(x, v) = a(t, x, v), \quad \Phi^s_s(x, v) = (x, v) \]
Approximating the phase-flow

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

\[
\begin{align*}
    x_{k-1} &:= x_k - \Delta t \cdot v_k, \\
v_{k-1} &:= v_k + \Delta t \cdot E(t_{k-1}, x_{k-1}),
\end{align*}
\]

Electric field is known at previous time-step!

For nonlinear case, $t = t_k$:

1. Compute $\rho_k$ using numerical quadrature.
2. Compute electric potential $\varphi_k$ from $\rho_k$ using FFT.
3. Save coefficients of $\varphi_k$ in B-Spline basis.

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

No need to save \( f \)!
- Saving \( \varphi_h \).
- Shift workload from "memory access" to "computation on the fly":
  - High flop/Byte ratio.
  - Cache can store coefficients of \( \varphi_h \) for several time-steps.

<table>
<thead>
<tr>
<th>( d )</th>
<th>( N_x = N_v )</th>
<th>Memory of ( \varphi_h ) per step</th>
<th>Memory for storing ( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>128</td>
<td>0.125 MiB</td>
<td>( \approx 2.0 \cdot 10^3 ) MiB</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>0.5 MiB</td>
<td>( \approx 3.3 \cdot 10^4 ) MiB</td>
</tr>
<tr>
<td>2</td>
<td>512</td>
<td>2 MiB</td>
<td>( \approx 5.2 \cdot 10^5 ) MiB</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>0.015 GiB</td>
<td>( \approx 3.3 \cdot 10^4 ) GiB</td>
</tr>
<tr>
<td>3</td>
<td>256</td>
<td>0.125 GiB</td>
<td>( \approx 2.1 \cdot 10^6 ) GiB</td>
</tr>
<tr>
<td>3</td>
<td>512</td>
<td>1 GiB</td>
<td>( \approx 1.3 \cdot 10^8 ) GiB</td>
</tr>
</tbody>
</table>
Conservation properties of NuFI

\[ f_h(t, x, v) = f_0(\Phi^0_{t,h}(x, v)) \]

- Satisfies the maximum principle exactly

\[ 0 \leq f_h(t, x, v) \leq \|f_0\|_{L^\infty(\Omega \times \mathbb{R}^d)} \]

- Exact conservation of all \( L^p \)-norms

\[
\frac{d}{dt} \left( \int \int f_h(t, x, v)^p \, dx \, dv \right) = 0
\]

and the kinetic entropy

\[
\frac{d}{dt} \left( \int \int f_h \log(f_h) \, dx \, dv \right) = 0.
\]

- Numerical proof for conservation of total energy

\[
\frac{d}{dt} \left( \int \int v^2 f_h \, dx \, dv + \int E_h^2 \, dx \right) = 0.
\]

Relative errors - computed using numerical quadrature:
Numerical experiments
Two-Stream instability

\[ f_0(x, v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} (1 + 0.01 \cos(0.5x)) \]

- Dynamics captured correctly.
- Low resolution leads to distortion.
Errors

**Two Stream 1d** \((h \to 0, \Delta t \text{ fixed}): \) Single- (left) & double precision (right).

**Two Stream 2d** (single precision): \(h \to 0 \) left & \(\Delta t \to 0 \) right.
Computational complexity and scaling

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

⇒ Computational complexity **linear** in number of quadrature points!

⇒ Strong scaling.

Until here L1-Cache can contain coefficient vector.

GPU’s are not fully used yet.

\[
\begin{align*}
N_x \cdot N_v & \quad t_{\text{comp}} \quad (\text{in s}) \\
10^2 & \quad 10^{-1} \quad 10^1 \\
10^6 & \quad 10^2 \quad 10^3 \\
10^{10} & \quad 10^5 \quad 10^6
\end{align*}
\]

\[
\begin{align*}
N_x \cdot N_y \cdot N_u \cdot N_v & \quad t_{\text{comp}} \quad (\text{in s}) \\
10^5 & \quad 10^{-1} \quad 10^1 \\
10^8 & \quad 10^2 \quad 10^3 \\
10^{10} & \quad 10^5 \quad 10^6
\end{align*}
\]

\[
\begin{align*}
2^0 & \quad 2^{20} \\
2^2 & \quad 2^{26} \\
2^4 & \quad 2^{32}
\end{align*}
\]
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.

Future research:

• Enabling restart of simulation & starting from experimental data.
• Collision-handling.  Evaluation of $Q(f, f)$ still expensive.

Wilhelm, R.-P., ACoM RWTH Aachen, wilhelm@acom.rwth-aachen.de
Kirchhart, M., ACoM RWTH Aachen and Intel Co., matthias.kirchhart@intel.com