Structure preserving numerical methods for the Vlasov equation

Lukas Einkemmer
University of Innsbruck

Oberwolfach, 2016
Main simplifying assumption

- collisionless plasma
- described by a (classic) particle density function

**Kinetic equation** describes evolution of particle density $f$

$$\partial_t f(t, x, v) + v \cdot \nabla_x f(t, x, v) + F \cdot \nabla_v f(t, x, v) = 0$$

$$F = q(E + v \times B),$$

coupled to Gauss’s law or Maxwell’s equations.

**Applications** include plasma physics, radiative heat transfer, astrophysics.
Landau damping

Nonlinear Landau damping and BGK equilibria

Two-stream instability

Time splitting for the Vlasov–Poisson equations

Free streaming part

\[ \partial_t f(t, x, v) + v \cdot \nabla_x f(t, x, v) = 0 \]

Solution:

\[ f(\tau, x, v) = f(0, x - \tau v, v) \]

But: \( F \) is not constant – nonlinear equation

Idea: Freeze force term at half the time step, i.e. at \( \tau/2 \).

- only translations have to be computed

Generalized to the Vlasov–Maxwell equations.

---

Acceleration part

\[ \partial_t f(t, x, v) + F \cdot \nabla_v f(t, x, v) = 0 \]

Solution for constant \( F \):

\[ f(\tau, x, v) = f(0, x, v - \tau F) \]

---

Cubic spline interpolation

Follow the **characteristics** backward in time

- semi-Lagrangian approach

The feet of the characteristics do not necessarily coincide with the numerical grid

- Interpolation using **cubic splines**

Results in a **mass conservative** scheme.
Discontinuous Galerkin semi-Lagrangian method

**Idea:** translation and projection to a subspace of piecewise polynomials (without any continuity constraint).

Yields a **conservative**, **local**, and only **slightly-diffusive** scheme.

Well suited for **parallelization**.

---

Weak scaling for the two-dimensional Vlasov–Poisson equations on the VSC-2.

What structure to preserve

**Infinite number** of conserved quantities for Vlasov–Poisson

- charge, all $L^p$ norms, current, energy, entropy, ...

Focus here on the **physical relevant quantities**

- charge, current, energy

and two measures of dissipation

- $\| f \|_2^2$, entropy

<table>
<thead>
<tr>
<th>conserved var.</th>
<th>splitting</th>
<th>dG</th>
<th>spline</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>charge</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>current</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>energy</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>entropy</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>$L^1$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>$L^2$</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

additional term for $p = 0$

additional term for $p = 0$, $p = 1$
A common way to analyze the conservation of a numerical scheme is to consider a modified equation.

- Cubic spline interpolation violates the second law of thermodynamics
- and increases the $L^2$ norm.

Both of these help to conserve the quantity in question but are completely unphysical.
Theorem
The $L^2$ norm is a decreasing function of time for the semi-Lagrangian discontinuous Galerkin scheme.

Proof.
Let us consider a translated function $g(x)$. Expanding in a Legendre series gives

$$g(x) = \sum_{k=0}^{\infty} g_k p_k(x).$$

Therefore

$$\|Pg\|_2^2 - \|g\|_2^2 = \sum_{k=0}^{\ell} g_k^2 - \sum_{k=0}^{\infty} g_k^2 \leq 0.$$
Nonlinear Landau damping – low resolution

- **dG4 (32), T=400**
- **spline (128), T=400**

- **Electric energy**
- **Momentum error**
- **Energy error**
- **Entropy**
- **L1 norm**
- **L2 norm**

Legend:
- **spline (128)**
- **dG2 (64)**
- **dG3 (43)**
- **dG4 (32)**
- **dG6 (21)**
Nonlinear Landau damping – high resolution

![Graphs showing electric energy, momentum error, energy error, entropy, L1 norm, and L2 norm for different resolutions and time steps.](image-url)
Theorem

A numerical algorithm that produces negative values and is mass conservative can not conserve the $L^1$ norm.

Proof.

\[
\int |f(0, x, v)| \, d(x, v) = \int f(0, x, v) \, d(x, v) \\
= \int f(t, x, v) \, d(x, v) \\
\leq \int |f(t, x, v)| \, d(x, v).
\]

The inequality becomes strict if the particle density function $f(t, x, v)$ is negative.
To limit or not to limit

The semi-Lagrangian discontinuous Galerkin scheme can be made $L^1$ conservative by using positivity limiters.

But that necessarily introduces additional dissipation (or additional overshoot).

Our numerical simulations indicate that positivity is almost always a less severe problem than dissipation

► this is more of an issue for spline interpolation – positivity preservation is more difficult in that context

Long time behavior

- Electric energy
- Momentum error
- Energy error
- Entropy
- L1 norm
- L2 norm

**Graphs:**
- dG2 (64), T=2e4
- dG4 (32), T=2e4

Legend:
- dG2 (64)
- dG4 (32)
- dG6 (21)
Higher order methods

Suppose we start with

\[ f(0, x, v) = e^{-v^2/2}(1 + \alpha \cos kx). \]

and solve \( \partial_t f + v\partial_x f = 0 \) we get

\[ f(t, x, v) = e^{-v^2/2}(1 + \alpha \cos k(x - vt)). \]

Thus,

\[ \partial_v^m f(t, x, v) \propto (kvt)^m. \]

For all intents and purposes \( \partial_v^6 f(t, x, v) = \infty \).

- But the **sixth order method outperforms** the lower order methods.
Bump-on-tail instability
Conservation in the exascale era

On future supercomputers we will most likely see

- less memory per core
- increased use of accelerators

**Single precision** would result in a reduction in memory and an increase in performance by a factor of 2.

- Conservation up to single precision?
Mixed-precision algorithm

In each cell we store \( c_0, \ldots, c_p \) where

\[
u(x) = \sum_{k=0}^{p} u_k p_k(x).
\]

Store \( c_0 \) in double $\Rightarrow$ Conservation of mass up to double

- Reduction in memory use by 1.6 (1D), 1.8 (2D), 1.97 (3D).

<table>
<thead>
<tr>
<th>order</th>
<th># double</th>
<th>Intel Xeon Phi</th>
<th>NVIDIA K80</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bandwidth</td>
<td>speedup</td>
<td>memorydown</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>67.9 GB/s</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>76.3 GB/s</td>
<td>1.80</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>68.6 GB/s</td>
<td>2.02</td>
</tr>
</tbody>
</table>
Conclusion

Thank you for your attention

http://arxiv.org/abs/1601.02280
http://arxiv.org/abs/1603.07008