Structure preserving numerical methods for the Vlasov equation

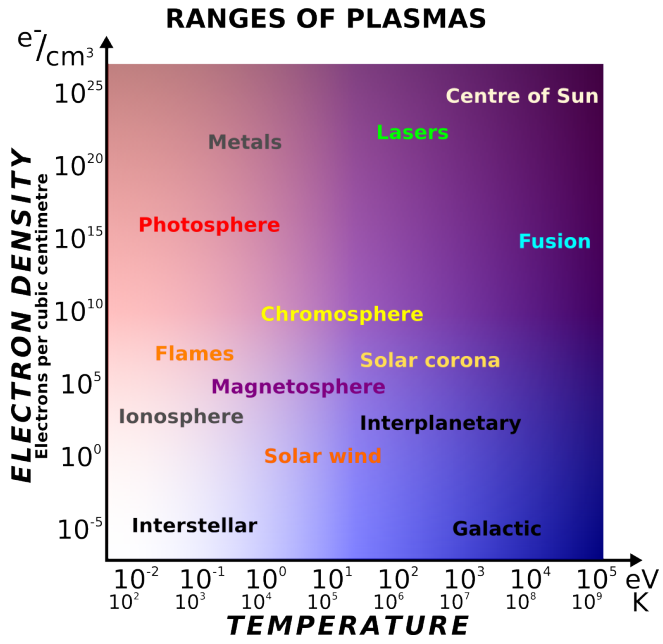
Lukas Einkemmer
University of Innsbruck

Oberwolfach, 2016

Vlasov equation

Main simplifying assumption

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



Kinetic equation describes evolution of particle density f

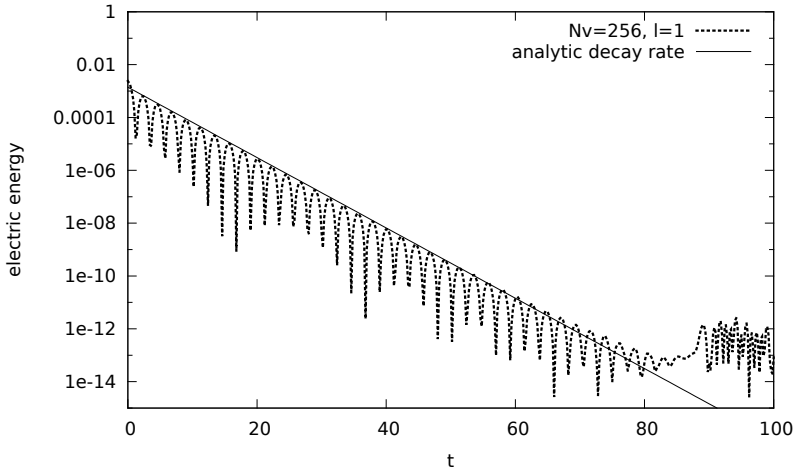
$$\partial_t f(t,x,v) + v \cdot \nabla_x f(t,x,v) + F \cdot \nabla_v f(t,x,v) = 0$$
$$F = q(E + v \times B),$$

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

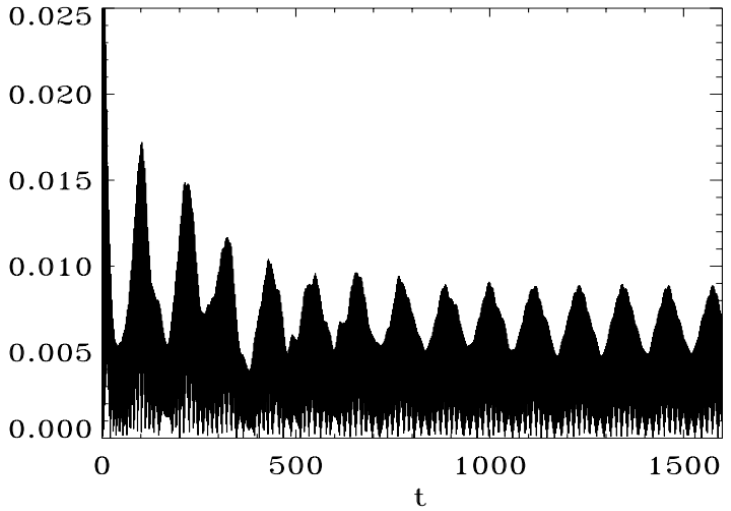
Applications include plasma physics, radiative heat transfer, astrophysics.

Physics

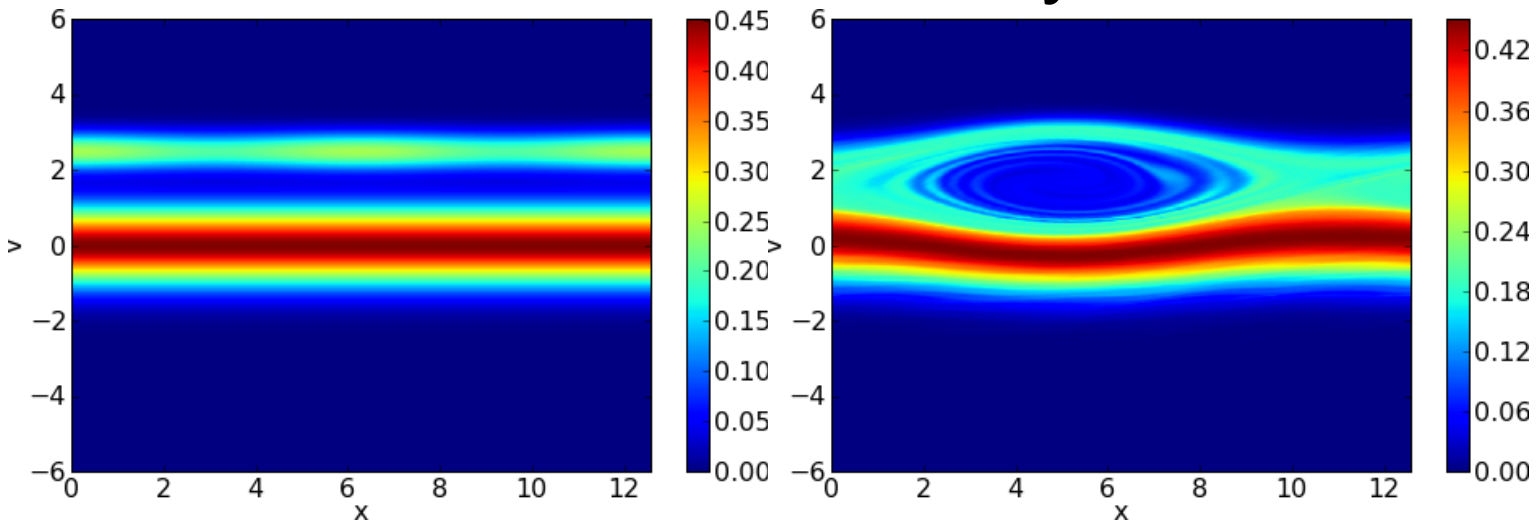
Landau damping



Nonlinear Landau damping and **BGK** equilibria



Two-stream instability



G. Manfredi, Phys. Rev. Lett. 1997

Time splitting for the Vlasov–Poisson equations

Free streaming part

$$\partial_t f(t, \boldsymbol{x}, \boldsymbol{v}) + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f(t, \boldsymbol{x}, \boldsymbol{v}) = 0$$

Solution:

$$f(\tau, \boldsymbol{x}, \boldsymbol{v}) = f(0, \boldsymbol{x} - \tau \boldsymbol{v}, \boldsymbol{v})$$

Acceleration part

$$\partial_t f(t, \boldsymbol{x}, \boldsymbol{v}) + \boldsymbol{F} \cdot \nabla_{\boldsymbol{v}} f(t, \boldsymbol{x}, \boldsymbol{v}) = 0$$

Solution for constant $\boldsymbol{F}$:

$$f(\tau, \boldsymbol{x}, \boldsymbol{v}) = f(0, \boldsymbol{x}, \boldsymbol{v} - \tau \boldsymbol{F})$$

But: $\boldsymbol{F}$ 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.

C.Z. Cheng, G. Knorr, J. Comput. Phys. (1976)
N. Crouseilles, L.E., E. Faou, J. Comput. Phys. (2015)

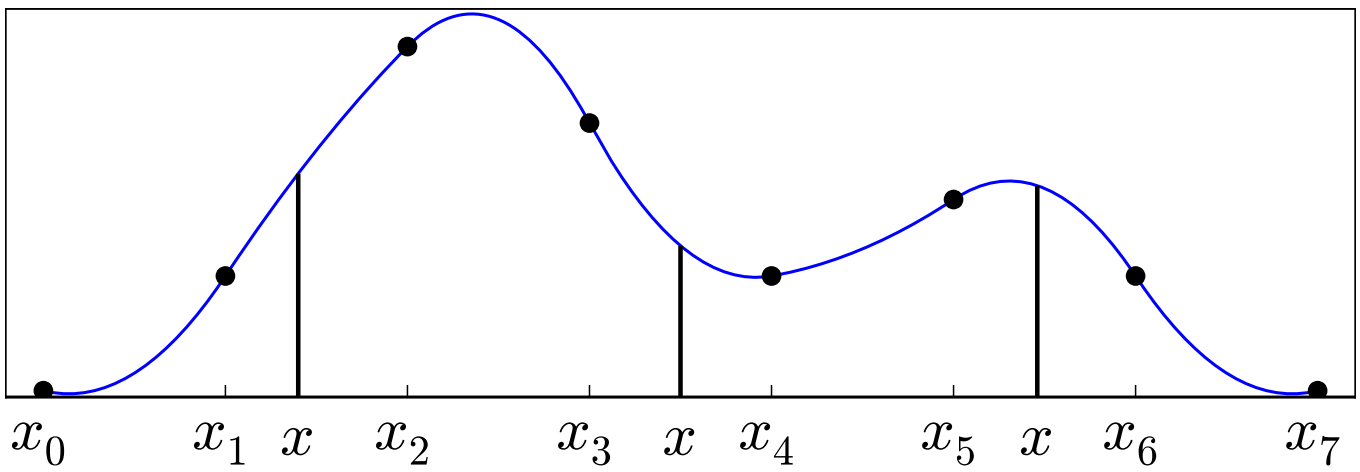
Cubic spline interpolation

Follow the **characteristics** backward in time

- semi-Lagrangian approach

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

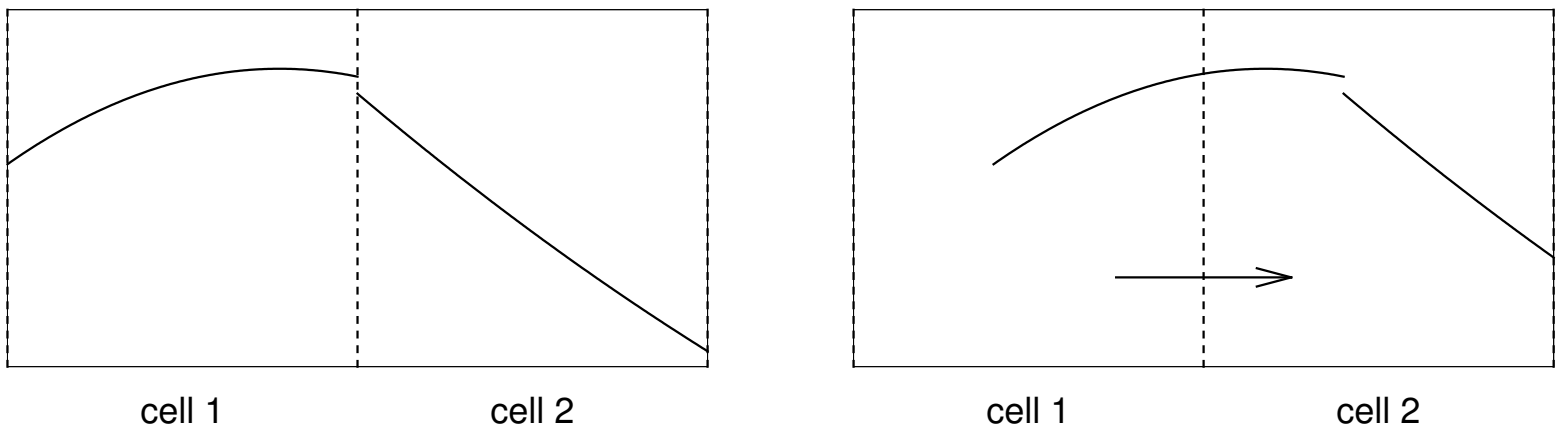
- Interpolation using **cubic splines**



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**.

J.A. Rossmanith, D.C. Seal, J. Comput. Phys. (2011)
J.M. Qiu, C.W. Shu, J. Comput. Phys. (2011)
N. Crouseilles, M. Mehrenberger, F. Vecil, ESAIM: Proceedings (2011)
L. Einkemmer, A. Ostermann, SIAM J. Numer. Anal. (2014)

Weak scaling for Vlasov–Poisson

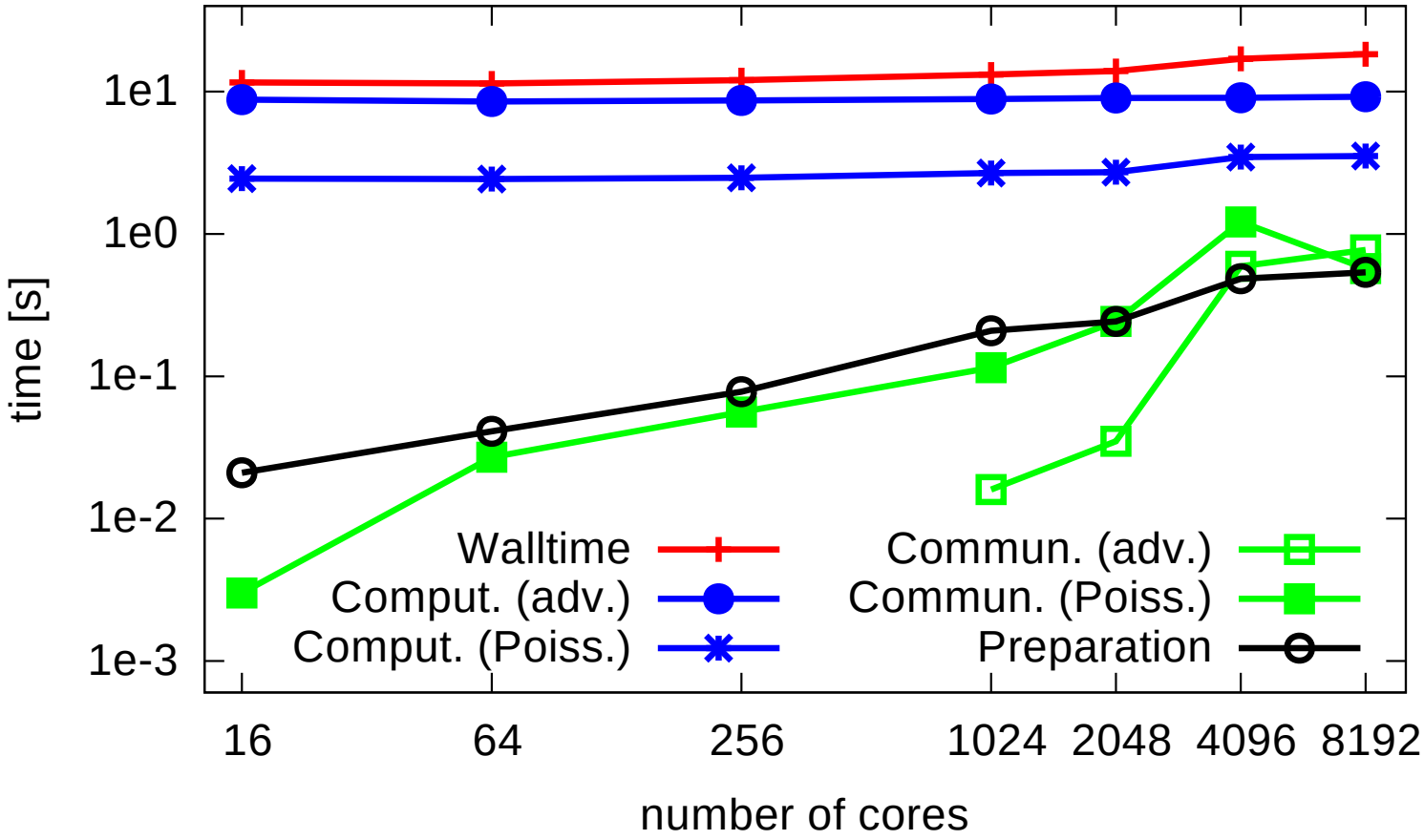


Figure: Weak scaling for the two-dimensional Vlasov–Poisson equations on the VSC-2.

L. Einkemmer, Comput. Phys. Commun. (2016)

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**

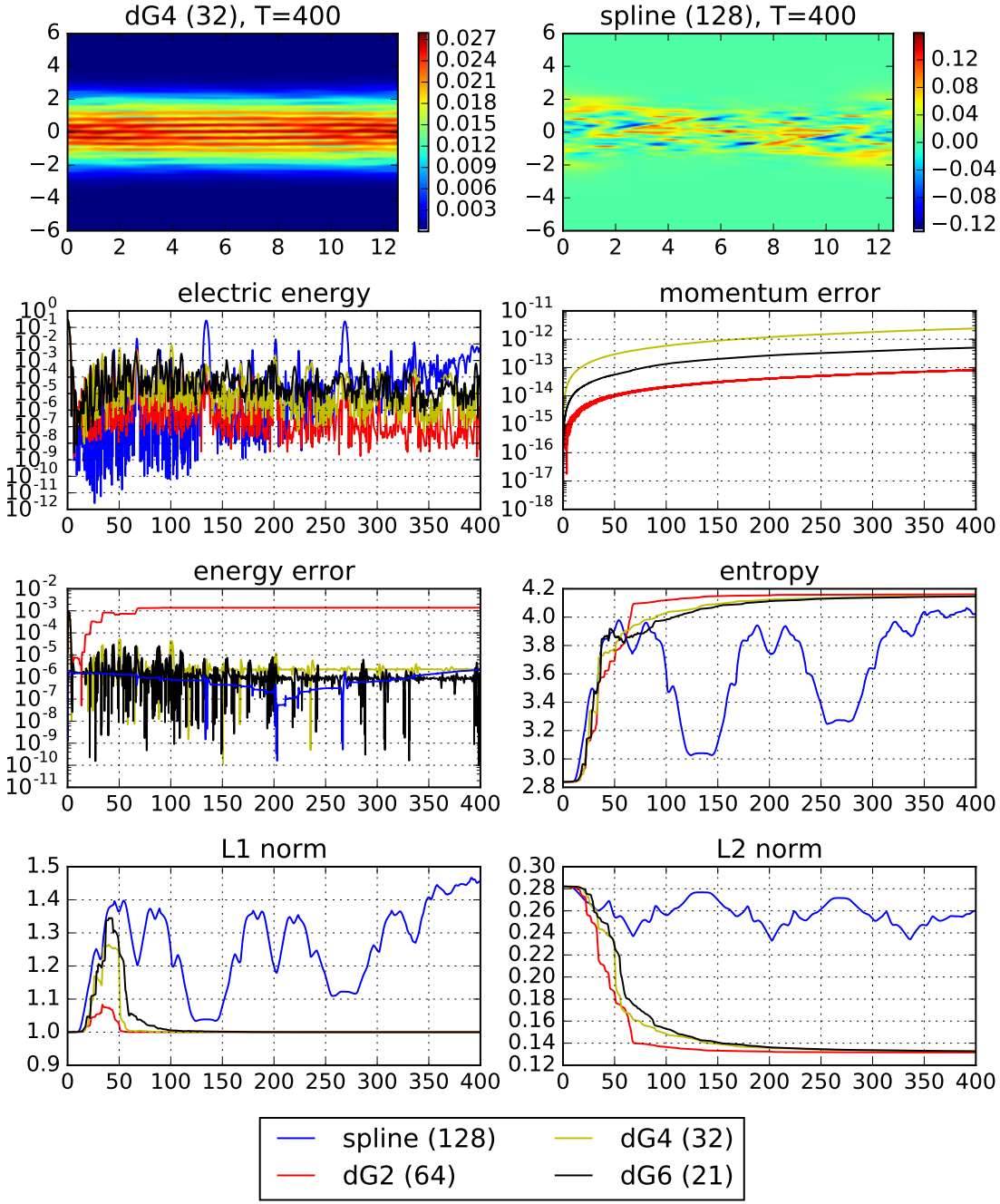
- charge, current, energy

and two measures of dissipation

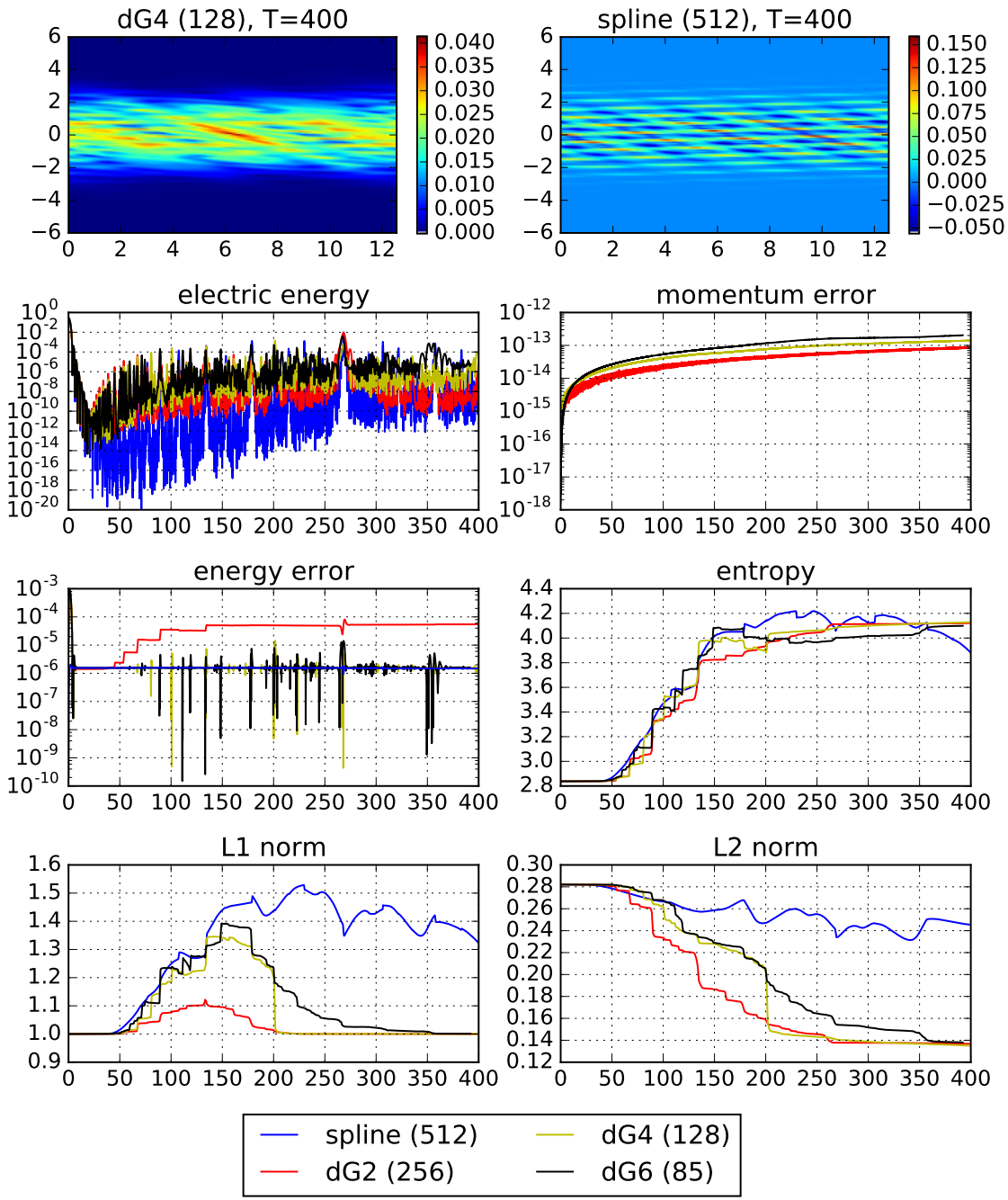
- $\|f\|_2^2$, entropy

conserved var.	splitting	dG	spline	remarks
charge	yes	yes	yes	
current	yes	yes†	yes†	additional term for $p = 0$
energy	no	no	no	additional term for $p = 0$, $p = 1$
entropy	yes	no	no	
L^1	yes	no*	no	
L^2	yes	no	no	

Expansion – low resolution



Expansion – high resolution



Dissipation

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.

L^2 is strictly decreasing for dG

Theorem

The L^2 norm is a decreasing function of time for the semi-Lagrangian discontinuous Galerkin scheme.

Proof.

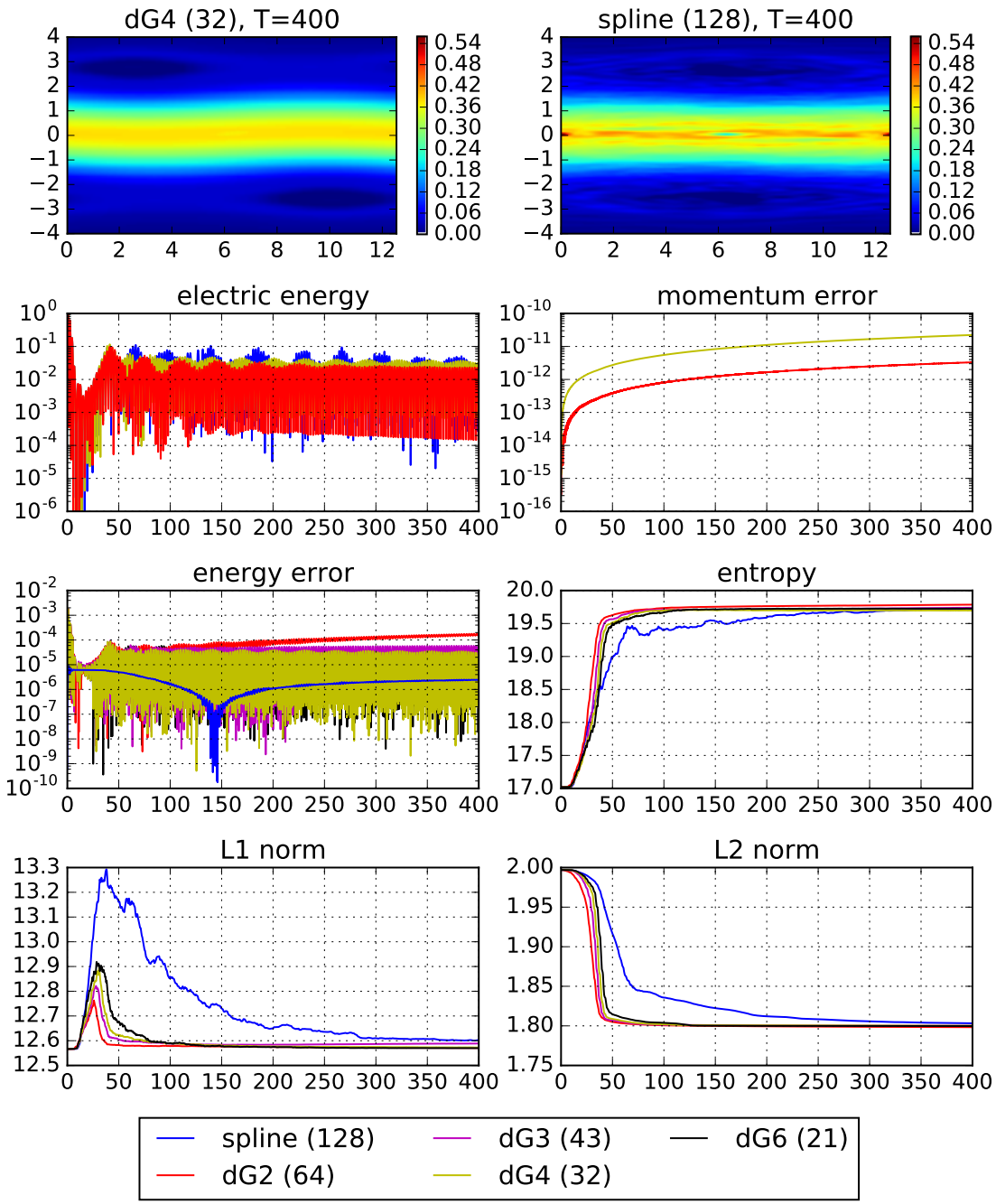
Let us consider a translated function $g(x)$. Expanding in a Legendre series gives

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

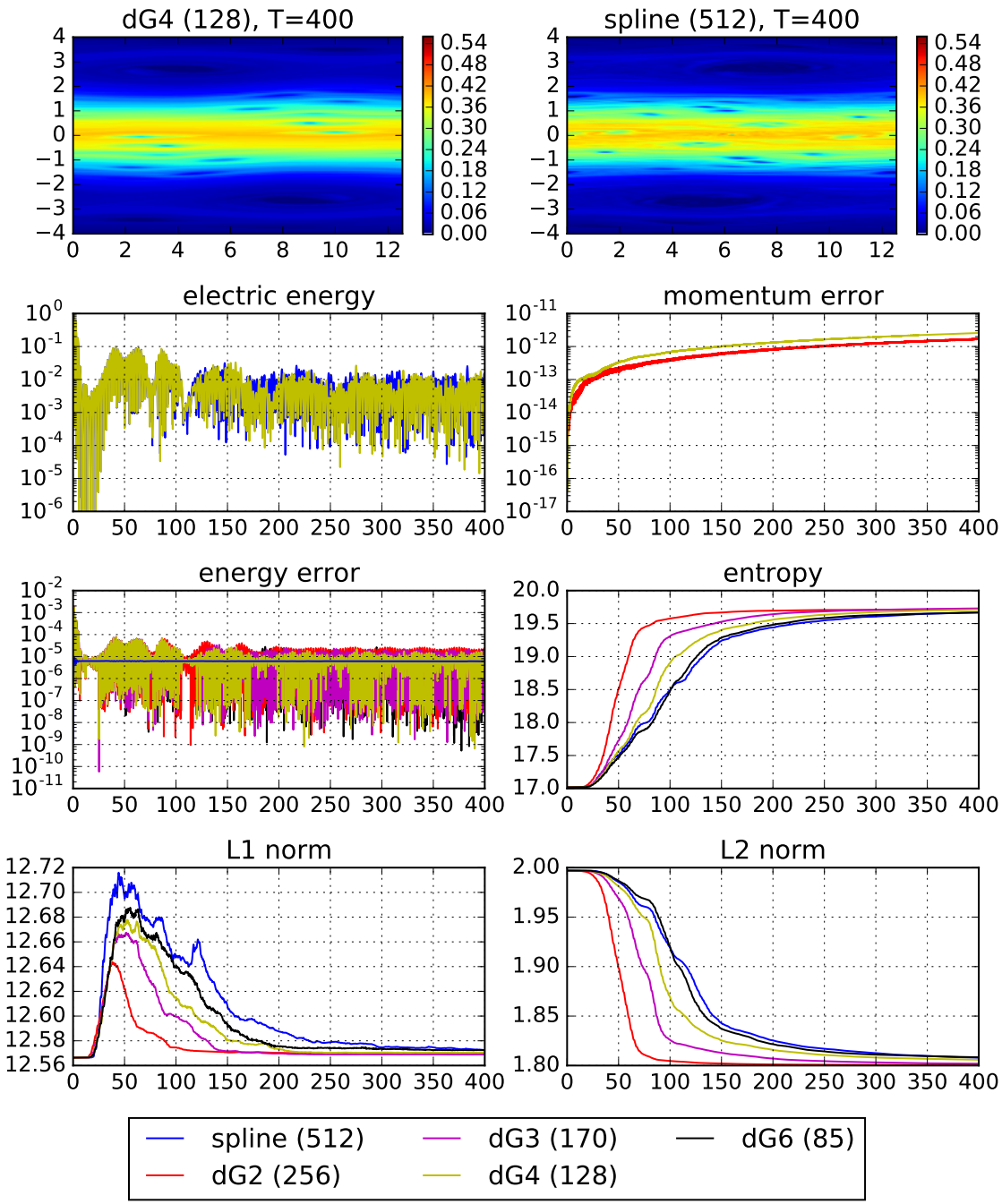
Therefore

$$\|Pg\|_2^2 - \|g\|_2^2 = \sum_{k=0}^{\ell} g_k^2 - \sum_{k=0}^{\infty} g_k^2 \leq 0.$$

Nonlinear Landau damping – low resolution



Nonlinear Landau damping – high resolution



L^1 conservation

Theorem

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

Proof.

$$\begin{aligned}\int |f(0,x,v)|\,\mathrm{d}(x,v) &= \int f(0,x,v)\,\mathrm{d}(x,v)\\ &= \int f(t,x,v)\,\mathrm{d}(x,v)\\ &\leq \int |f(t,x,v)|\,\mathrm{d}(x,v).\end{aligned}$$

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).

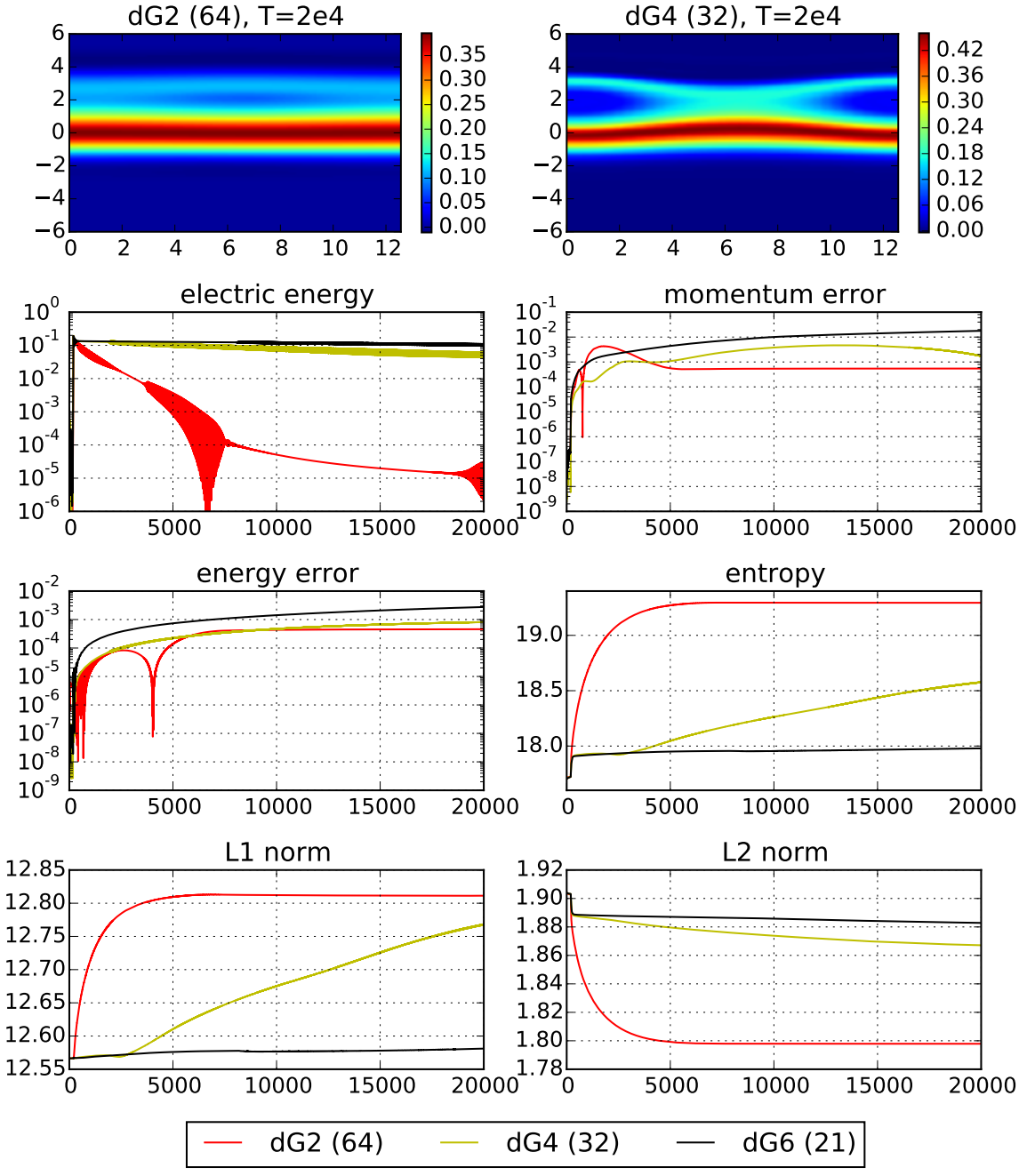
Our numerical simulations indicate that positivity is almost always a less severe problem than dissipation

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

---

J.A. Rossmanith, D.C. Seal, J. Comput. Phys. (2011)
J.M. Qiu, C.W. Shu, J. Comput. Phys. (2011)
N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, J. Comput. Phys. (2010)

Long time behavior



Higher order methods

Suppose we start with

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

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

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

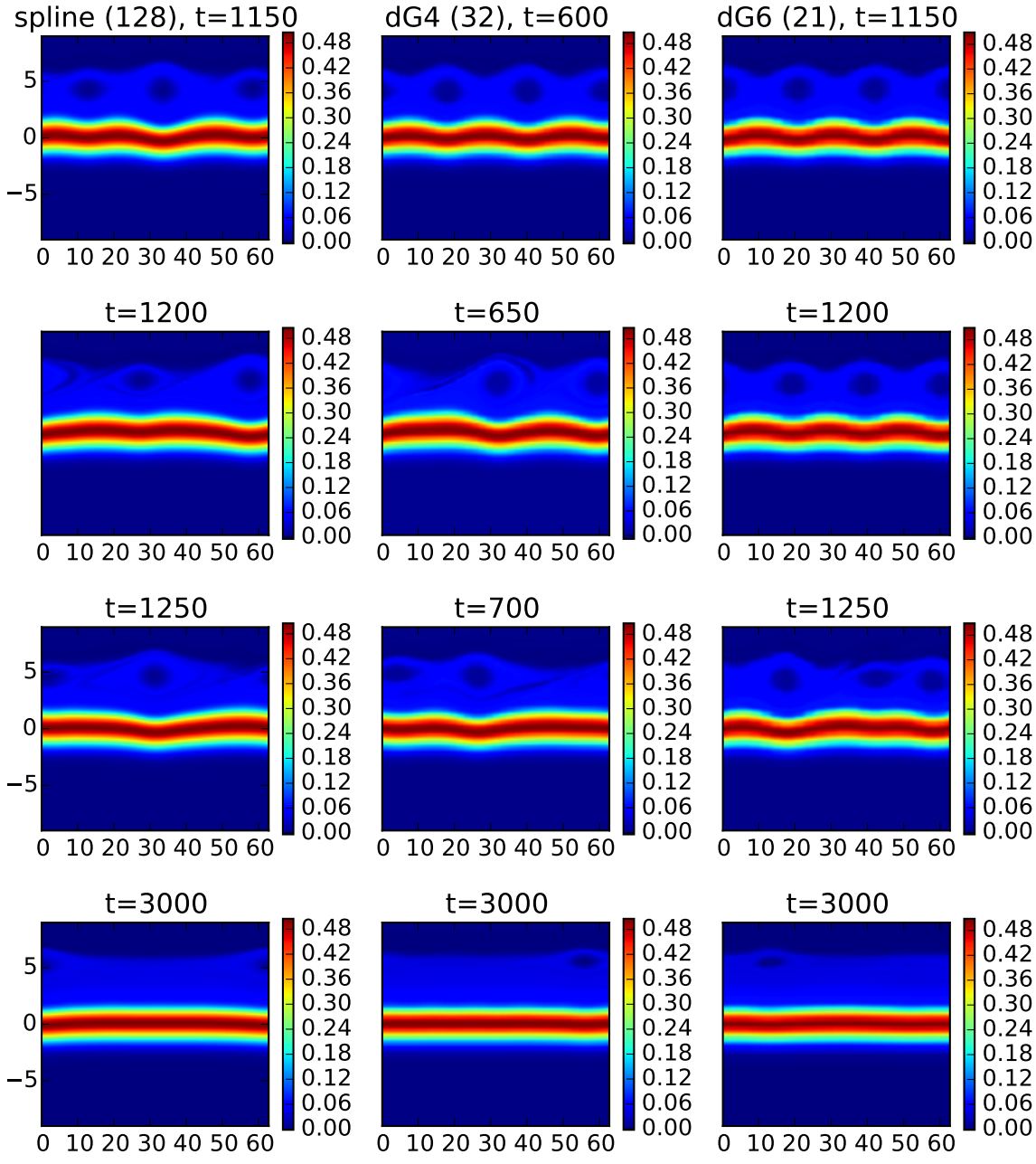
Thus,

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

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).

Intel Xeon Phi

order	# double	bandwidth	speedup	memorydown
4	4	67.9 GB/s	–	–
4	1	76.3 GB/s	1.80	1.60
4	0	68.6 GB/s	2.02	2.00

NVIDIA K80

order	# double	bandwidth	speedup	memorydown
4	4	137.9 GB/s	–	–
4	1	130.1 GB/s	1.51	1.60
4	0	142.5 GB/s	2.07	2.00

Conclusion

Thank you for your attention

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