A numerical investigation of error propagation for semi-Lagrangian methods

Lukas Einkemmer
joint work with Alexander Ostermann

Department of Mathematics
University of Innsbruck

Numerical Analysis of Evolution Equations, 8th NAI Workshop

Advection equation

A simple model problem

$$\partial_t u(t,x) + v\partial_x u(t,x) = 0$$

that numerically poses significant challenges.

We will consider numerical methods here that (at least for a class of problems) make no time discretization error.

- Fourier methods
- semi-Lagrangian methods

Such methods are important building block for splitting methods, exponential integrators, ...

Fast Fourier transform

Fourier transform gives

$$\partial_t \hat{u}(t,k) + ikv\hat{u}(t,k) = 0$$

with exact solution

$$\hat{u}(t,k) = \mathrm{e}^{-ikvt}\hat{u}(0,k).$$

After truncation we get (for $u(t,\cdot) \in \mathcal{C}^m$)

$$\|u(n\tau,x) - u_n(x)\| \leq Ch^{m-1}.$$

Application: almost everywhere where periodic boundary conditions are appropriate (Vlasov equation, KP equation, linear diffusion equation, ...).

Semi-Lagrangian methods

Follow the characteristics back in time

$$u_n(x_i) = u_{n-1}(X_{x_i}(\tau)).$$

For the advection we have $X_x(\tau) = x - v\tau$.

Interpolation to reconstruct the function value. We usually have

$$\|u(n\tau, x) - u_n(x)\| \leq C\left(h^q + \frac{h^q}{\tau}\right).$$

Application: almost everywhere where advection dominates (Vlasov equation, fluid dynamics, atmospheric simulations, ...).

Parallelization

On the road to exascale computing

- memory per core decreases
- accelerators (GPUs, Intel Xeon Phi) will be essential

Arithmetic operations are essentially free (high-order methods).

Moving to single-precision at least for parts of the computation.

- 2.5 speedup on modern GPUs
- decreases memory and communication by a factor of two

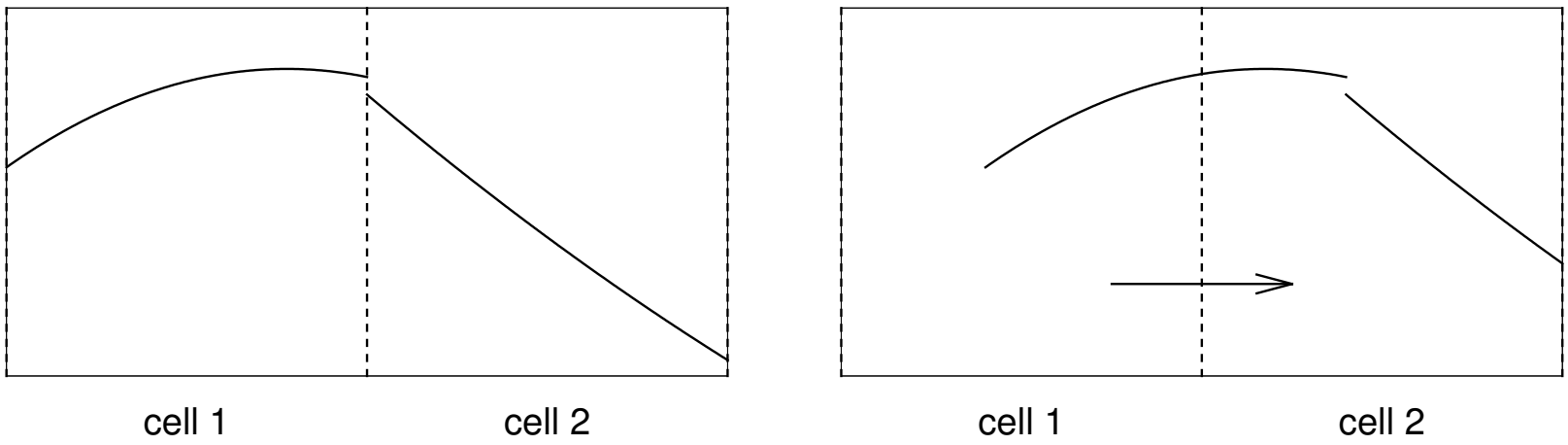
But high precision is also necessary in fields ranging from quantum physics to climate modeling.

$\Longrightarrow$ Numerical methods need to perform well close to machine precision. This is not a question of round-off errors but of error propagation.

ASCAC Subcommittee on Exascale Computing, 2010
D. Bailey, Comput. Sci. Eng., 7(3):54–61, 2005

Semi-Lagrangian discontinuous Galerkin method

Error estimates include a term proportional to the number of time steps.

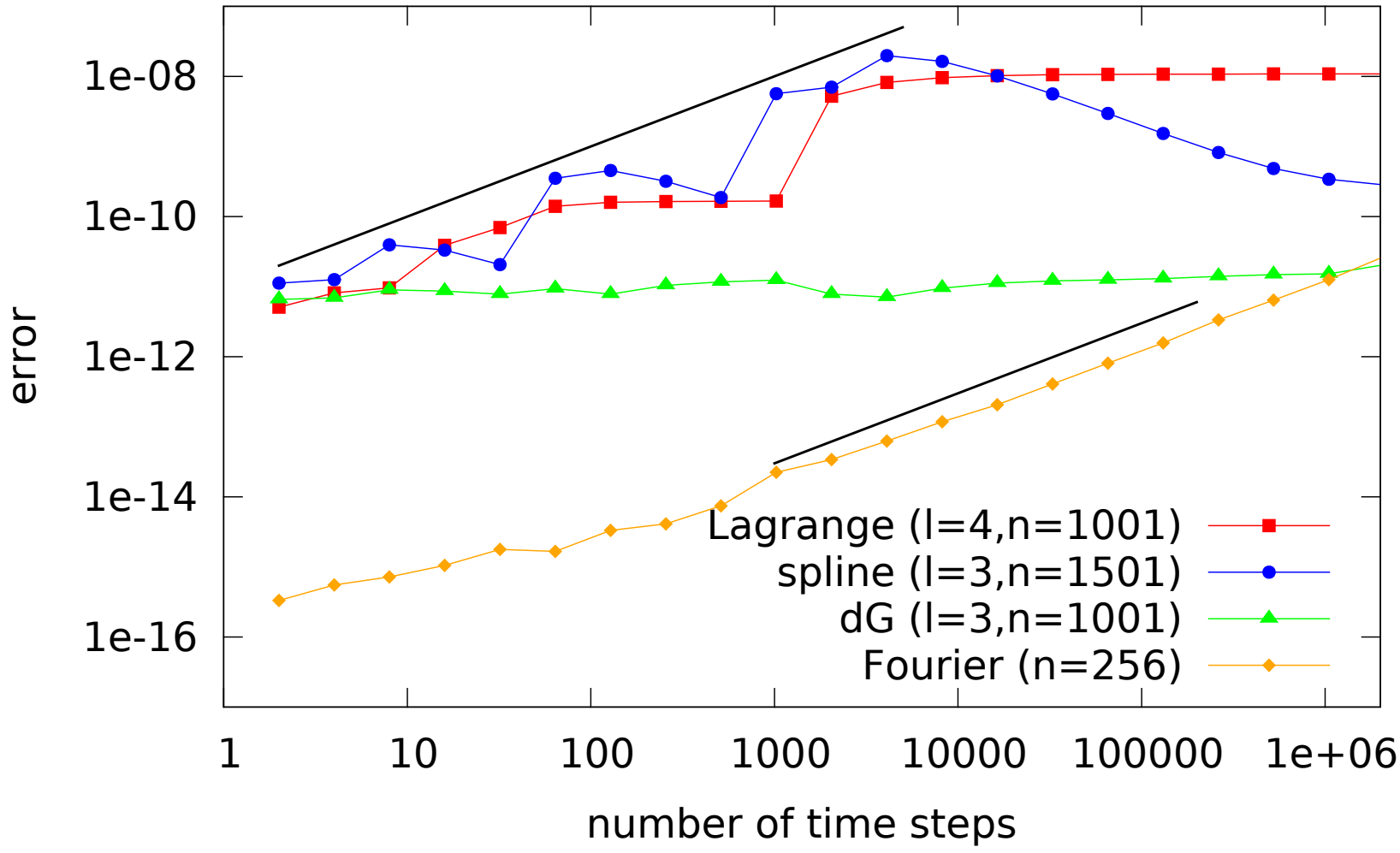


This yields a (mass) conservative method by construction.

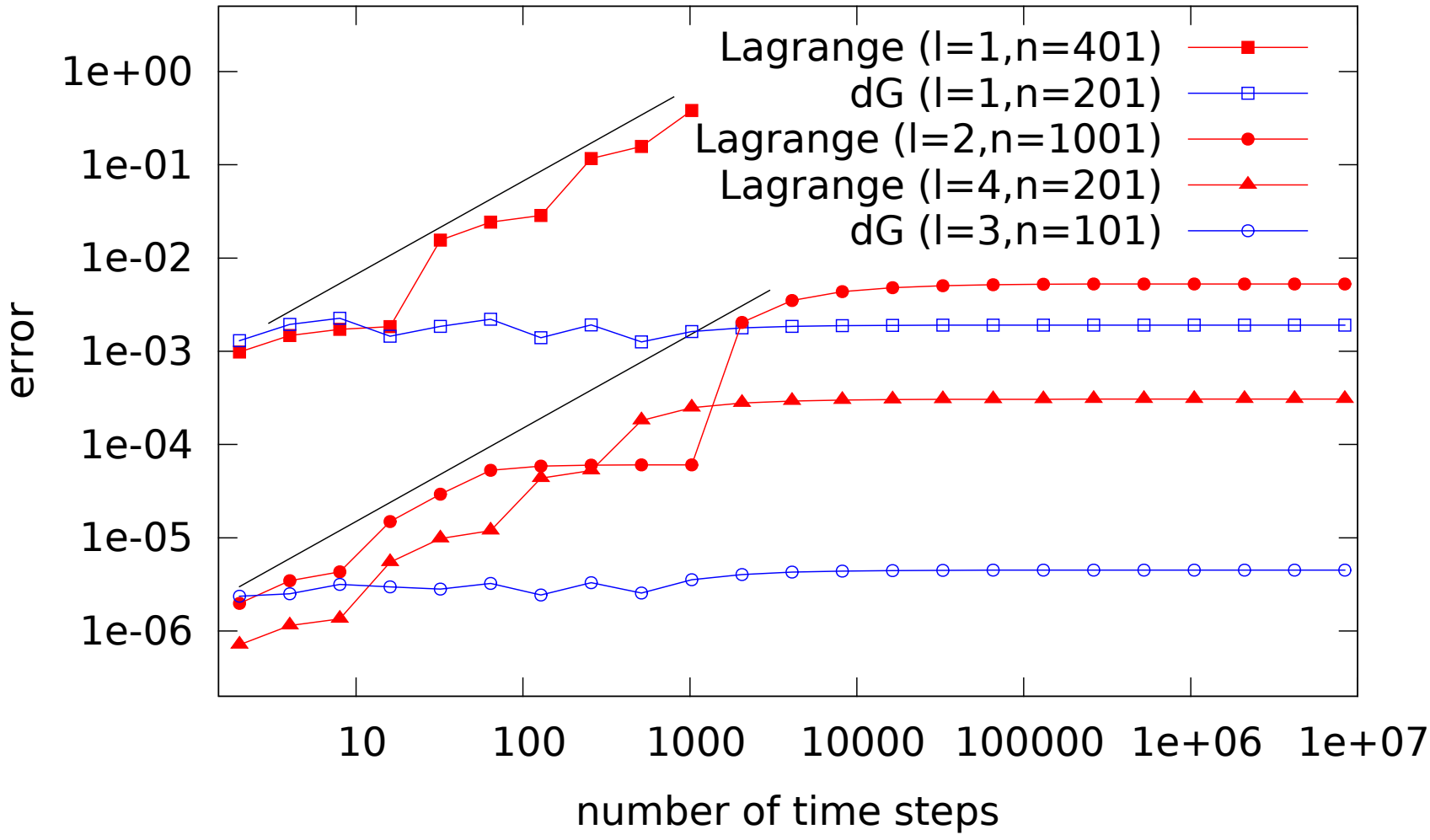
Convergence proofs require the $\mathcal{O}(h^p/\tau)$ term but not observed in simulations for the discontinuous Galerkin method.

L. Einkemmer, A. Ostermann, SIAM J. Numer. Anal., 52(2):757–778, 2014

Numerical simulation



Numerical simulation



Average error

Exact solution

$$\frac{1}{h}\int_0^h \frac{1}{h}\int_0^h u(x_i+\xi-\tau)\,\mathrm{d}\xi\mathrm{d}\tau = \frac{1}{6}\left(u(x_{i-1})+4u(x_i)+u(x_{i+1})\right)$$

Lagrange interpolation based semi-Lagrange method of degree one

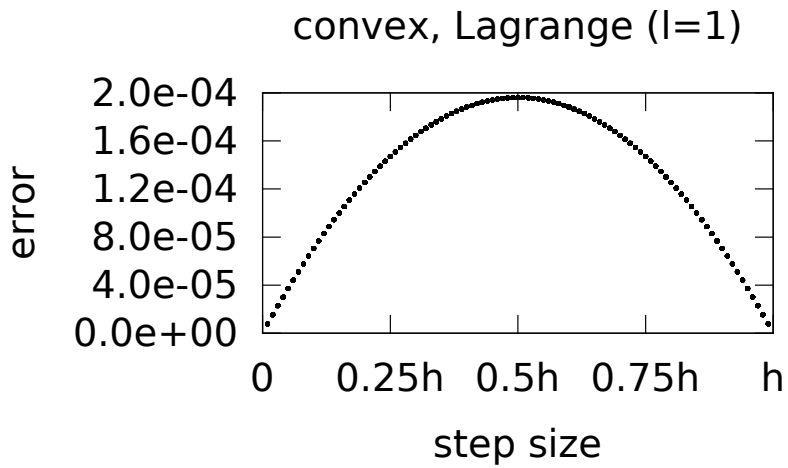
$$\frac{1}{h}\int_0^h \frac{1}{h}\int_0^h \tilde{u}(x_i+\xi)\,\mathrm{d}\xi\mathrm{d}\tau = \frac{1}{4}\left(u(x_{i-1})+2u(x_i)+u(x_{i+1})\right).$$

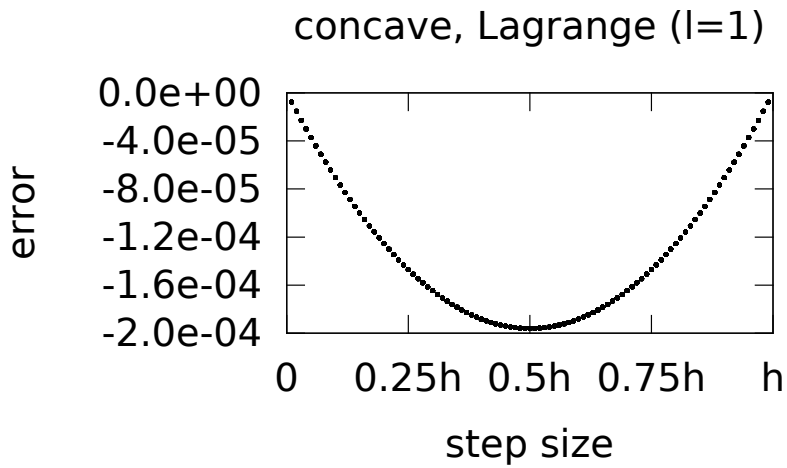
Average error (over one cell and all time step sizes)

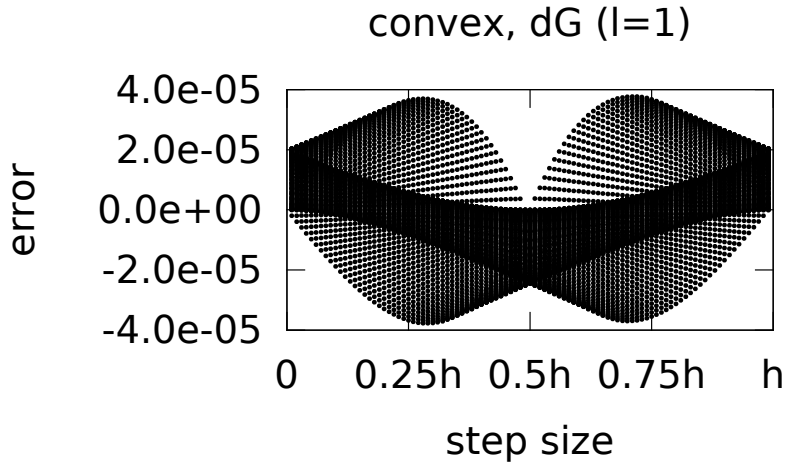
$$\overline{\overline{e}} = \frac{1}{12}\left(u(x_{i-1})-2u(x_i)+u(x_{i+1})\right) \approx \frac{h^2}{12}u''(x_i).$$

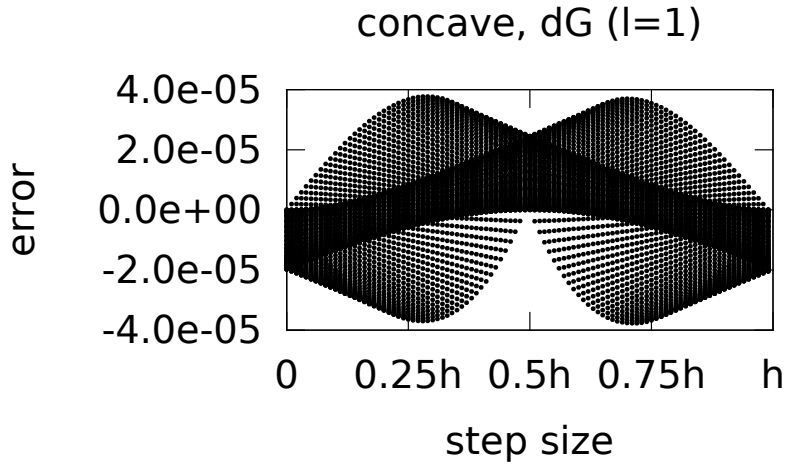
The average error of the discontinuous Galerkin scheme is zero by construction.

Numerical simulation

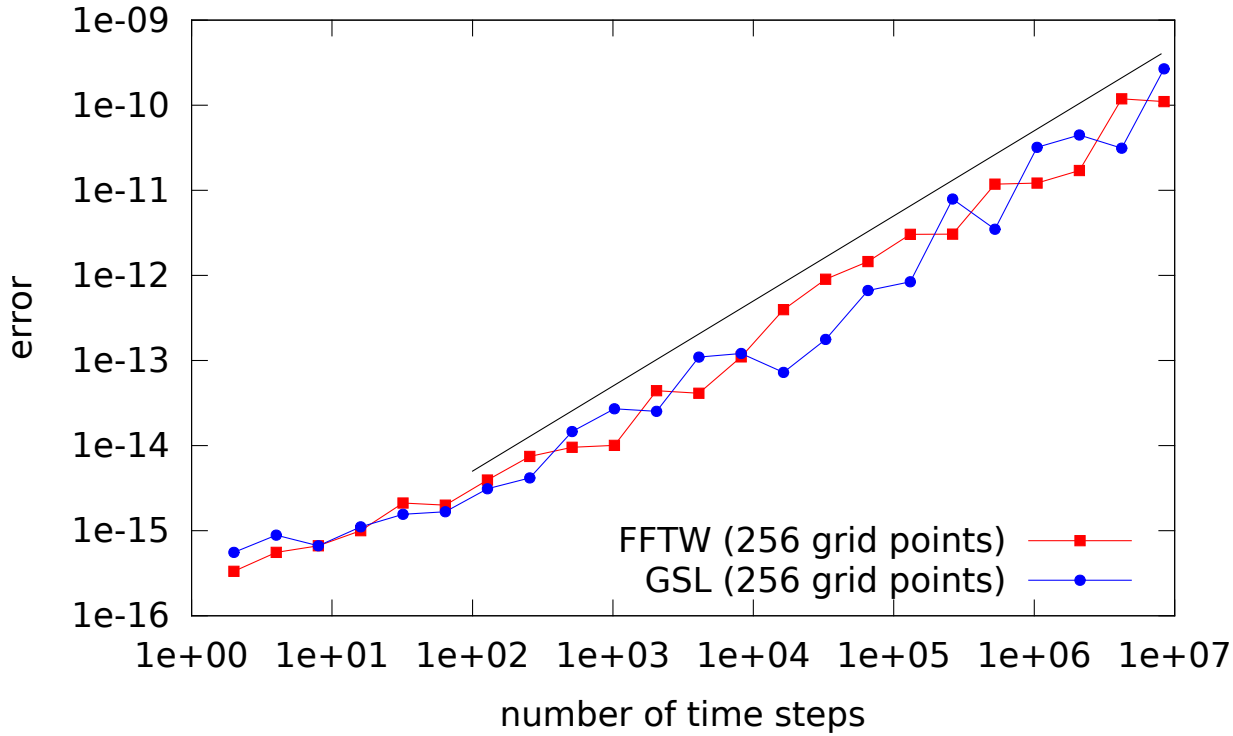






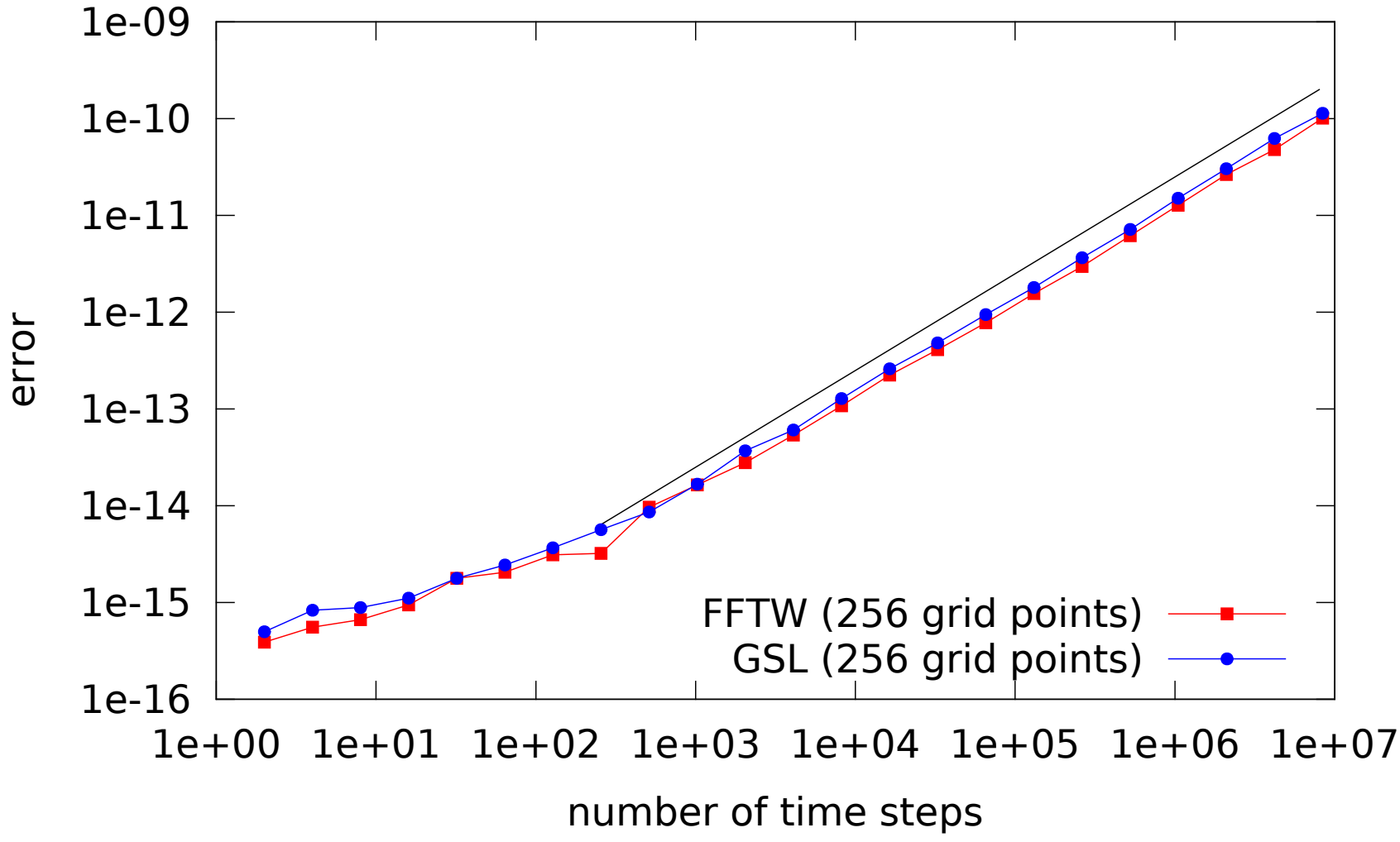


FFT



- Linear growth in error (similar to semi-Lagrangian schemes)
- FFT libraries consider square root scaling as number of points increase
- Well known that twiddle factors need to be rounded correctly to machine precision

Numerical simulation



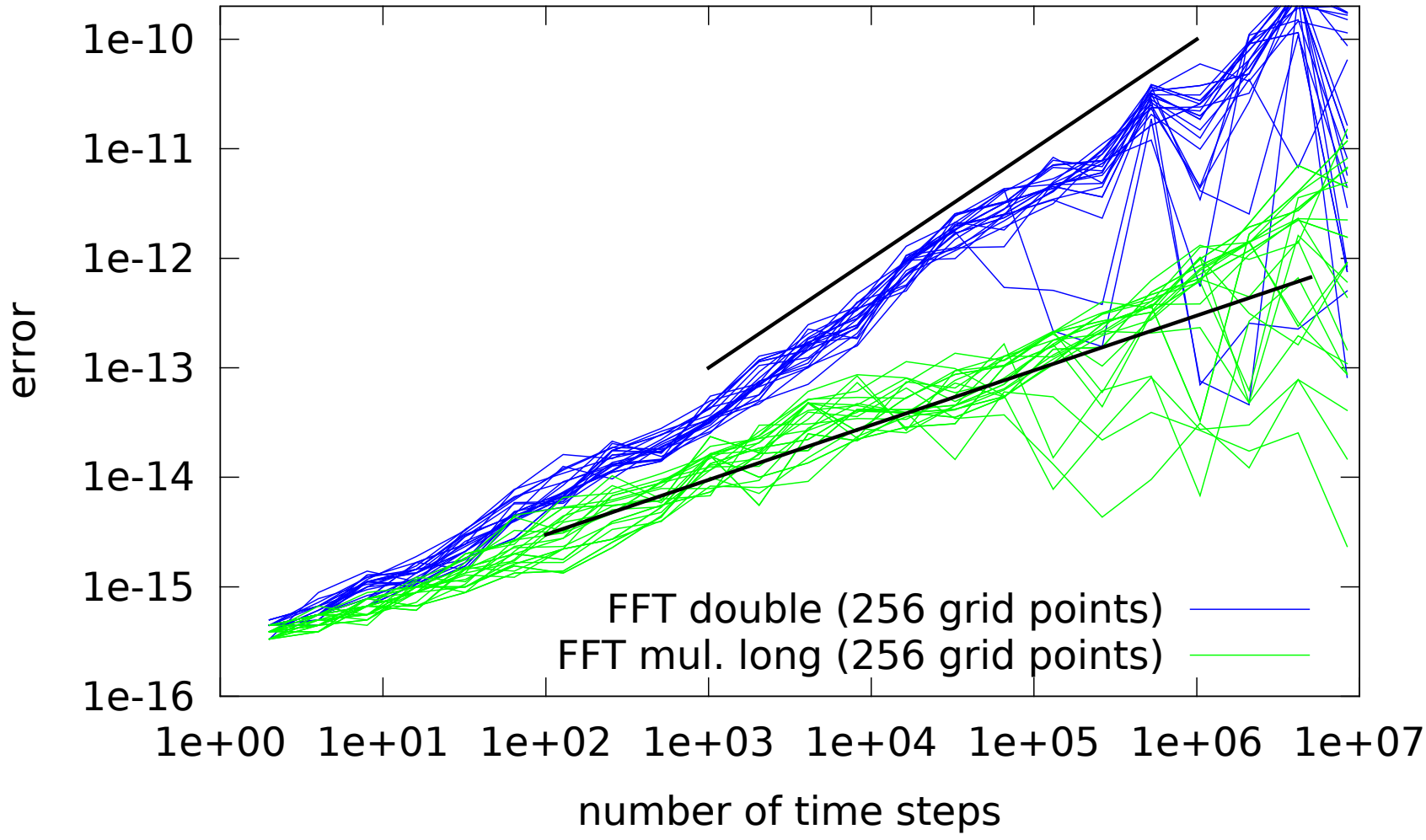
Cooley–Tukey algorithm

The Cooley–Tukey algorithm is often used synonymously with fast Fourier transform. But additional optimizations are performed in practice.

The Cooley–Tukey algorithm where the multiplications are computed in high precision (long double)

- 10-15% performance penalty in naive implementation
- storage requirement and communication is not changed

Numerical simulation



Putting everything together

Advection equation with source term

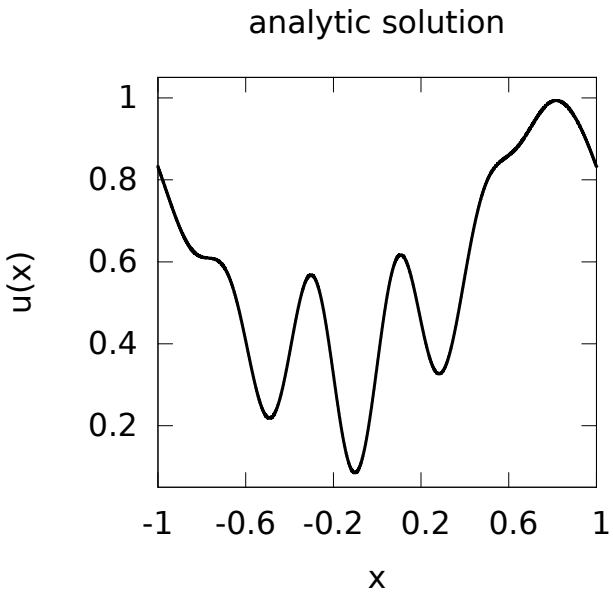
$$\partial_t u(t,x) + v\partial_x u(t,x) = s(x)$$

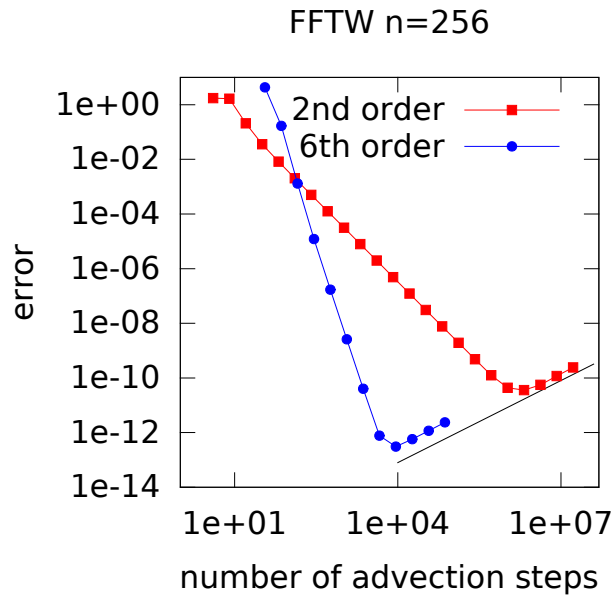
Splitting into an advection equation (computed using a semi-Lagrangian or Fourier approach) and the source term

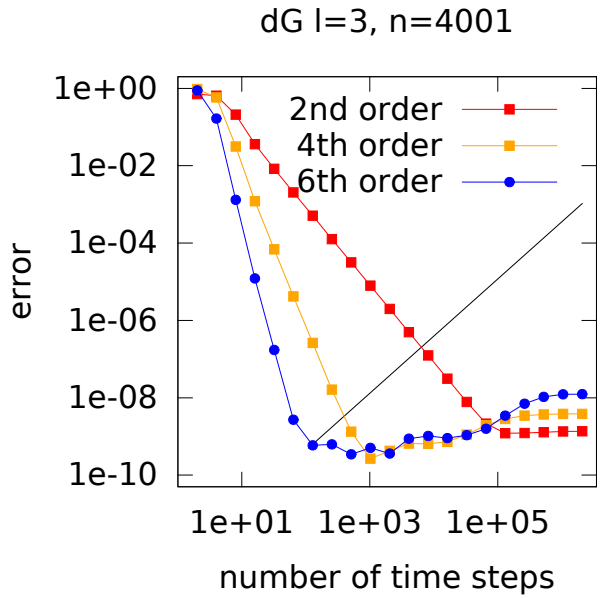
$$s(x) = (1 + \cos \pi x) \cos 5\pi x$$

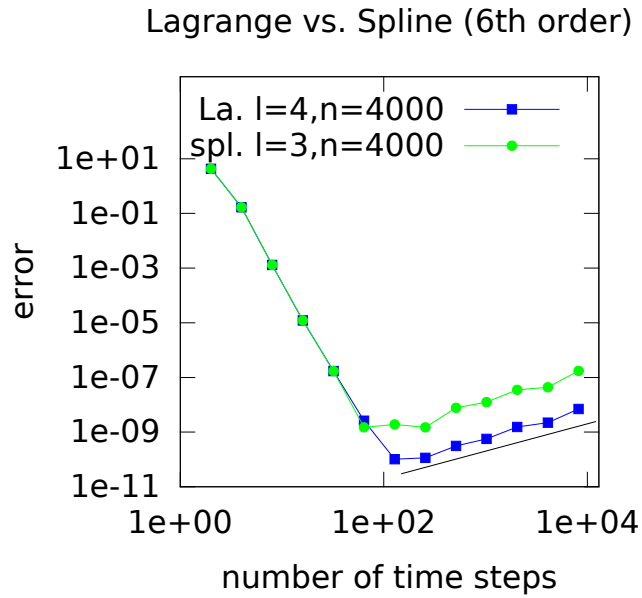
which can be solved analytically.

Numerical simulation



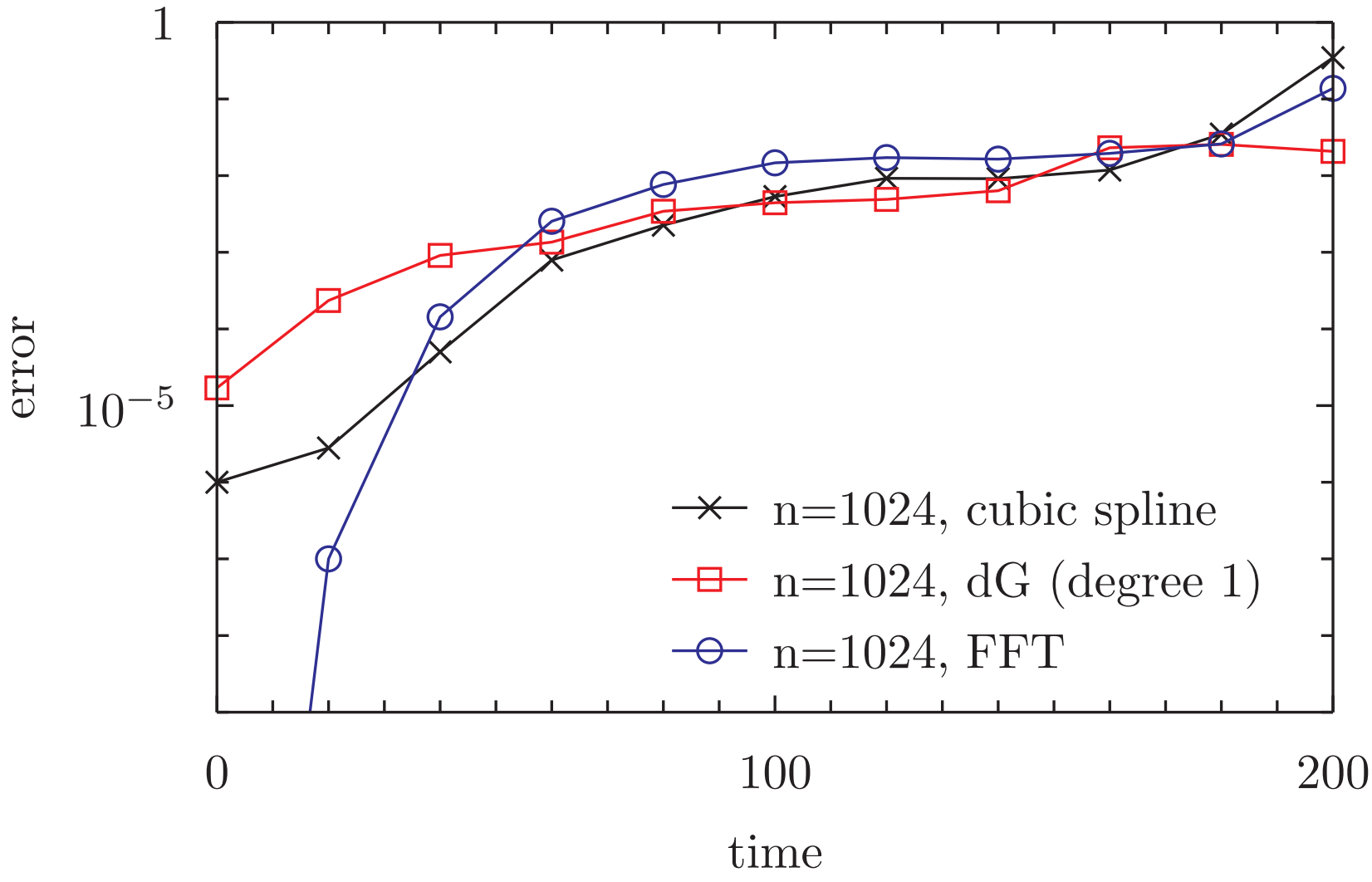






But that is not the end of the story

Vlasov–Poisson is split into two advection equations.



Excellent behavior of the semi-Lagrangian dG method.

Conclusion

- Due to computational constraints performance close to machine precision is becoming more important.
- Error propagation is vital in this context.
- The semi-Lagrangian discontinuous Galerkin method performs quite well in this context.
- The FFT shows a square root error propagation (with modest increase in computational effort).
- Exponential convergence is not everything.

Based on arXiv:1406.1933.