Simulation Techniques & Scientific Computing

UNIVERSITÄT SIEGEN

Locally Linearized Euler Equations in Discontinuous Galerkin with Legendre Polynomials

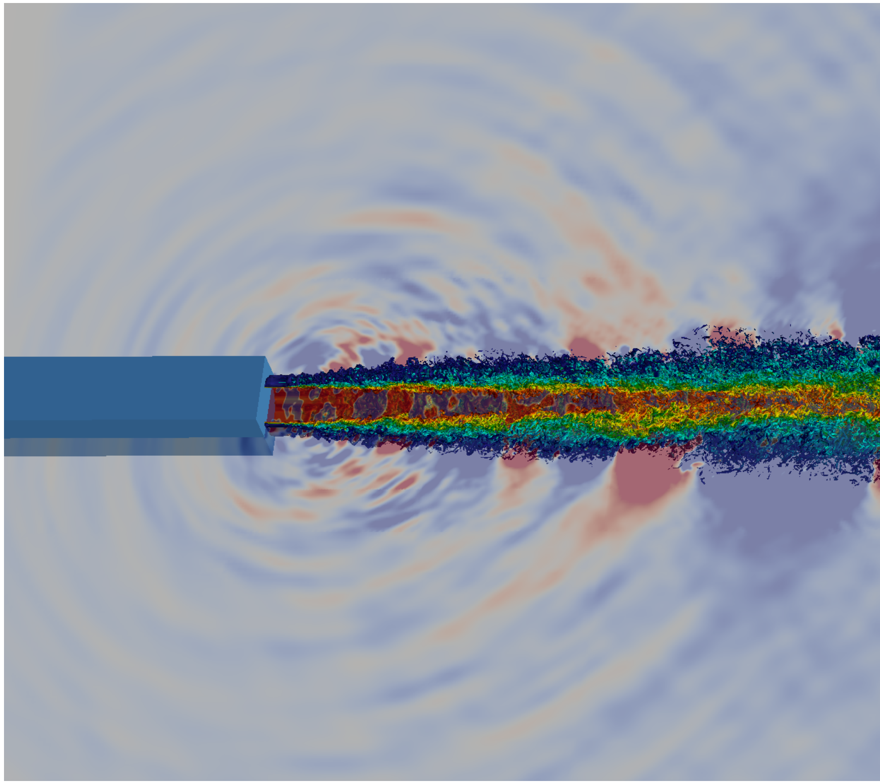
Harald Klimach, Michael Gaida, Sabine Roller

harald.klimach@uni-siegen.de

26th WSSP 2017

Motivation

- Fluid-Dynamic simulations with regions of different behavior



- Modal DG scheme
- Efficient computation

Inviscid Flow

- Nonlinear Euler equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial m}{\partial x} = 0$$

$$\frac{\partial m}{\partial t} + \frac{\partial (\frac{m^2}{\rho} + p)}{\partial x} = 0$$

$$\frac{\partial e}{\partial t} + \frac{\partial \frac{m \cdot (e+p)}{\rho}}{\partial x} = 0$$

$$p = (\gamma - 1)\left(e - \frac{m^2}{2 \cdot \rho}\right)$$

Vector Notation

- In vector notation, the Euler equations can be written as

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0$$

- F is the flux and nonlinear
- With the Jacobian J we can also write

$$\frac{\partial u}{\partial t} + J(u)\frac{\partial u}{\partial x} = 0$$

Linearization

- For the linearization we split u into a constant mean state u_0 and perturbations of that state u':

$$u = u_0 + u'$$

- With u_0 constant in space and neglecting products of perturbations, a linear formulation is obtained:

$$\frac{\partial u}{\partial t} + J(u_0)\frac{\partial u'}{\partial x} = 0$$

Discontinuous Galerkin

- Mesh discretization
- Approximation of the solution by functions within elements
- Flux exchange between elements

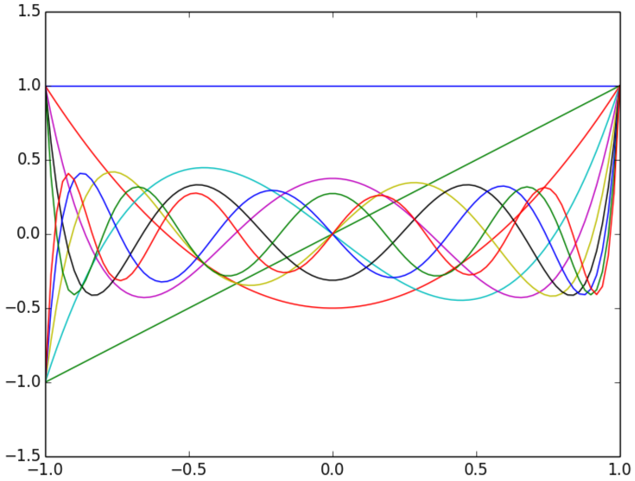
$$\partial_t \int_{\Omega_i} \mathbf{u}\psi \mathrm{d}V - \int_{\Omega_i} \mathbf{F} \cdot \nabla\psi \mathrm{d}V + \int_{\partial\Omega_i} \mathbf{G}\psi \cdot \mathbf{n} \mathrm{d}S = 0$$

Element-local computation

Neighbor dependency

Legendre Polynomials

- Orthogonal polynomial basis
- First mode = integral mean
- Higher dimensions by tensor product -> integral mean still in first mode only



$$L_0(x) = 1$$

$$L_1(x) = x$$

$$L_i(x) = \frac{2i-1}{i} \cdot x \cdot L_{i-1}(x) - \frac{i-1}{i} L_{i-2}(x)$$

Local Linearization within Elements

- State approximated by series of Legendre polynomials

$$u_h = \sum_{i=0}^{n} \hat{u}_i L_i(x)$$

- The first mode is the integral mean in the element, and we use this as u_0 in the linearization:

$$u_{0_h} + u'_h = \hat{u}_0 + \sum_{i=1}^{n} \hat{u}_i L_i(x)$$

Locally Linearized Flux

- We can now linearize the flux for our numerical approximation

$$F(u_h) \approx F(\hat{u}_0) + J(\hat{u}_0)u'_h$$

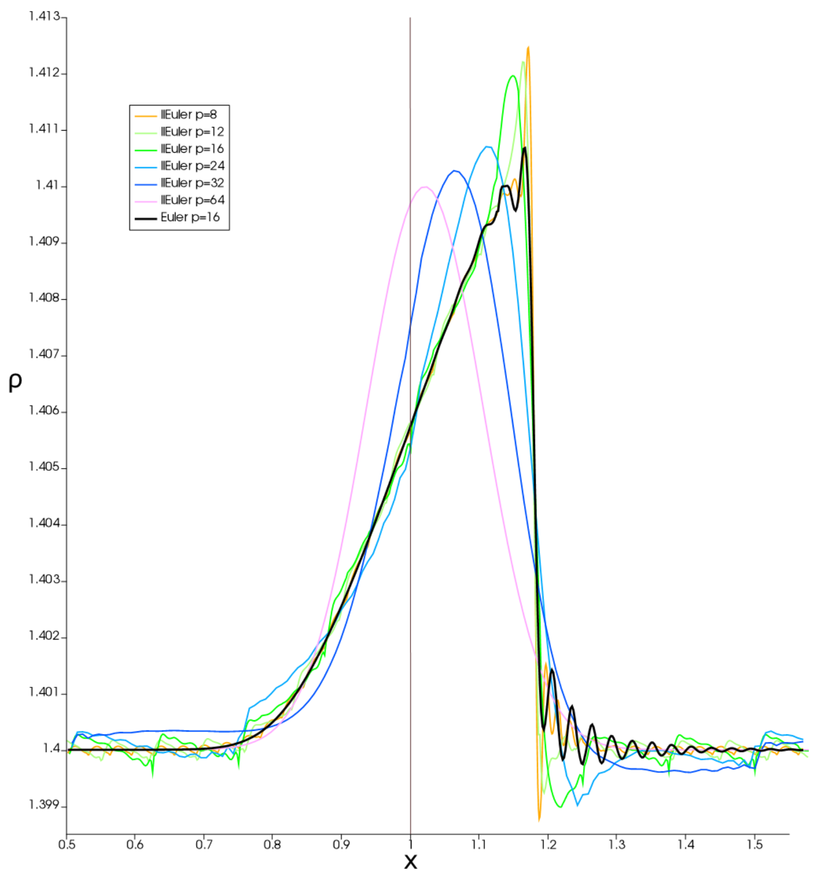
- u_0 now is only spatially constant within the element, it varies from element to element and over time
- Between elements we use the nonlinear flux G

Properties of this Approach

- This partial linearization of the physical flux allows us to completely stay in modal space within elements
- Projection to nodal space only on surfaces (reduced dimensionality)
- Avoid aliasing
- Same data structure as for nonlinear
- Degree of linearization depends on order of the scheme

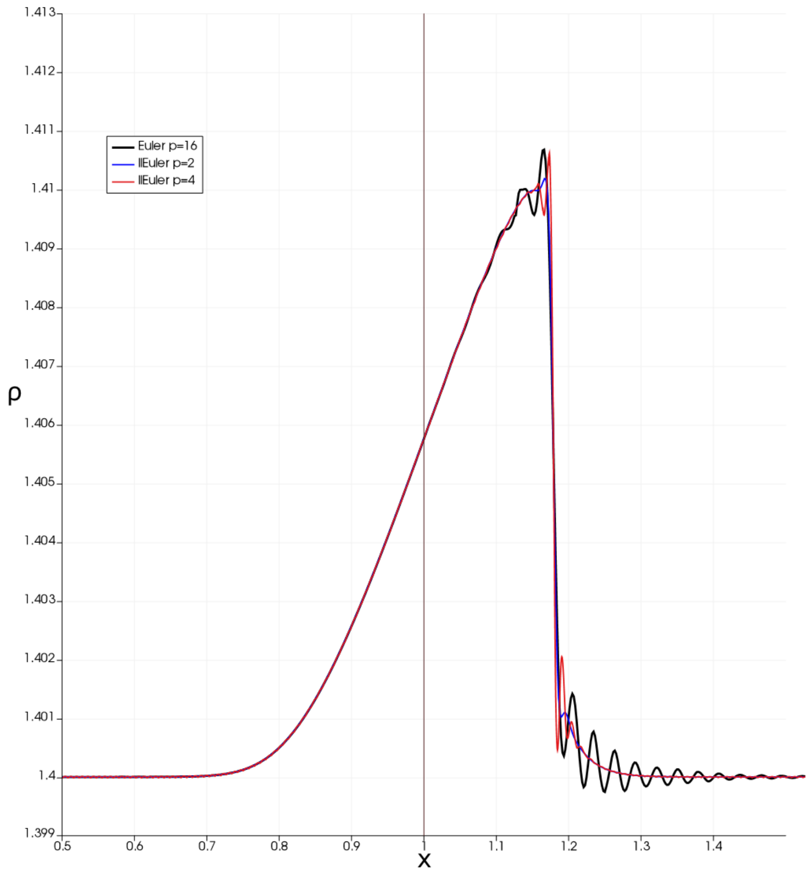
Travelling Wave (Higher Orders)

p	nElems
8	64
12	29
16	16
24	8
32	4
64	1

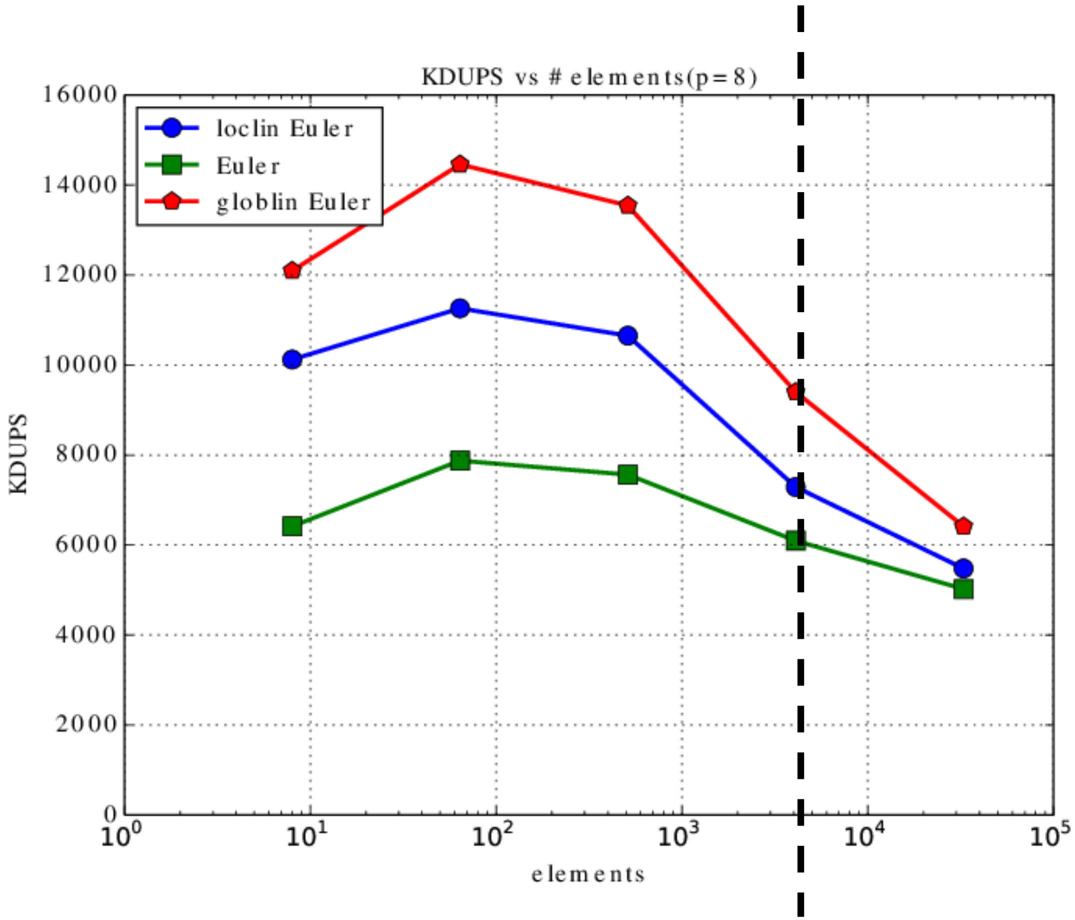


Travelling Wave (Low Order)

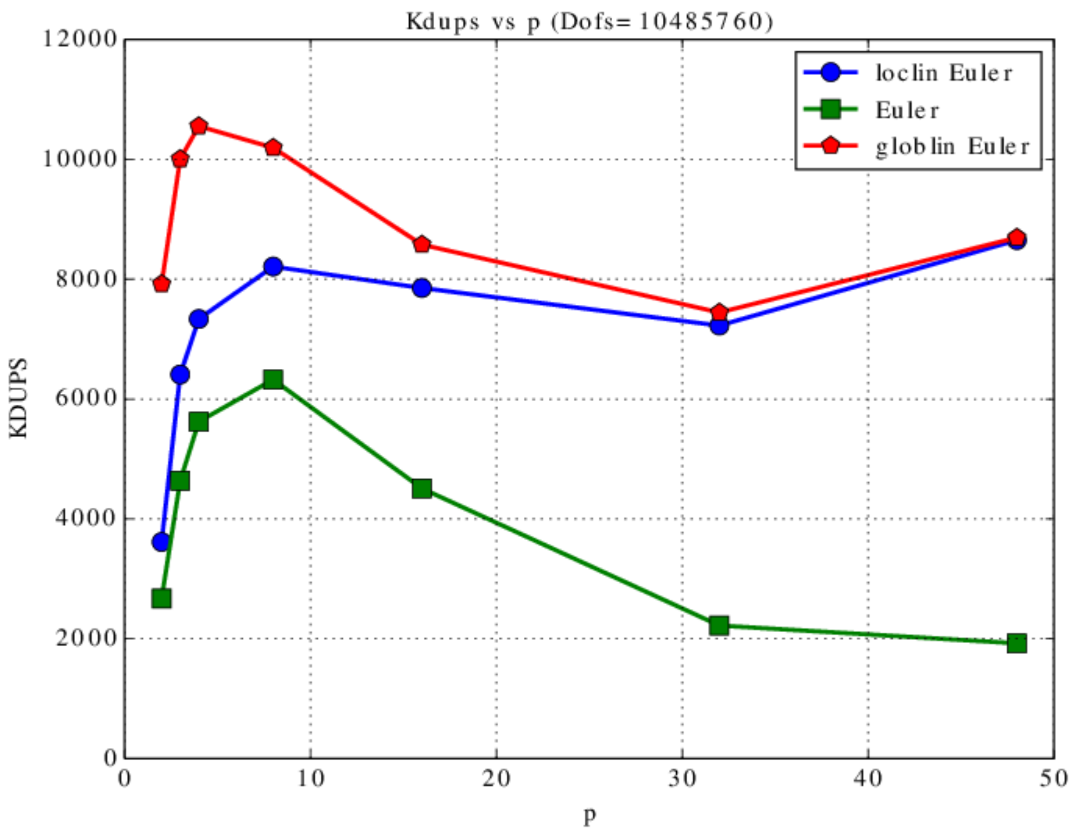
p	nElems
2	1024
4	256



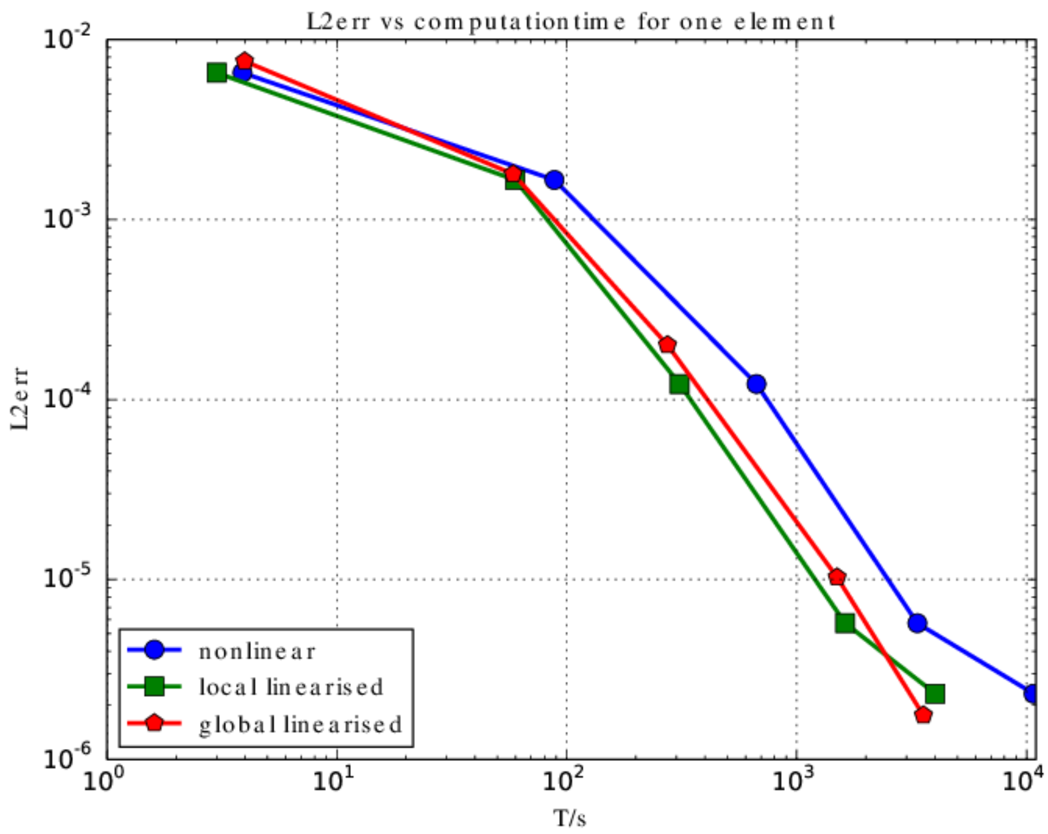
Performance for 8th Order



Performance over Scheme Order



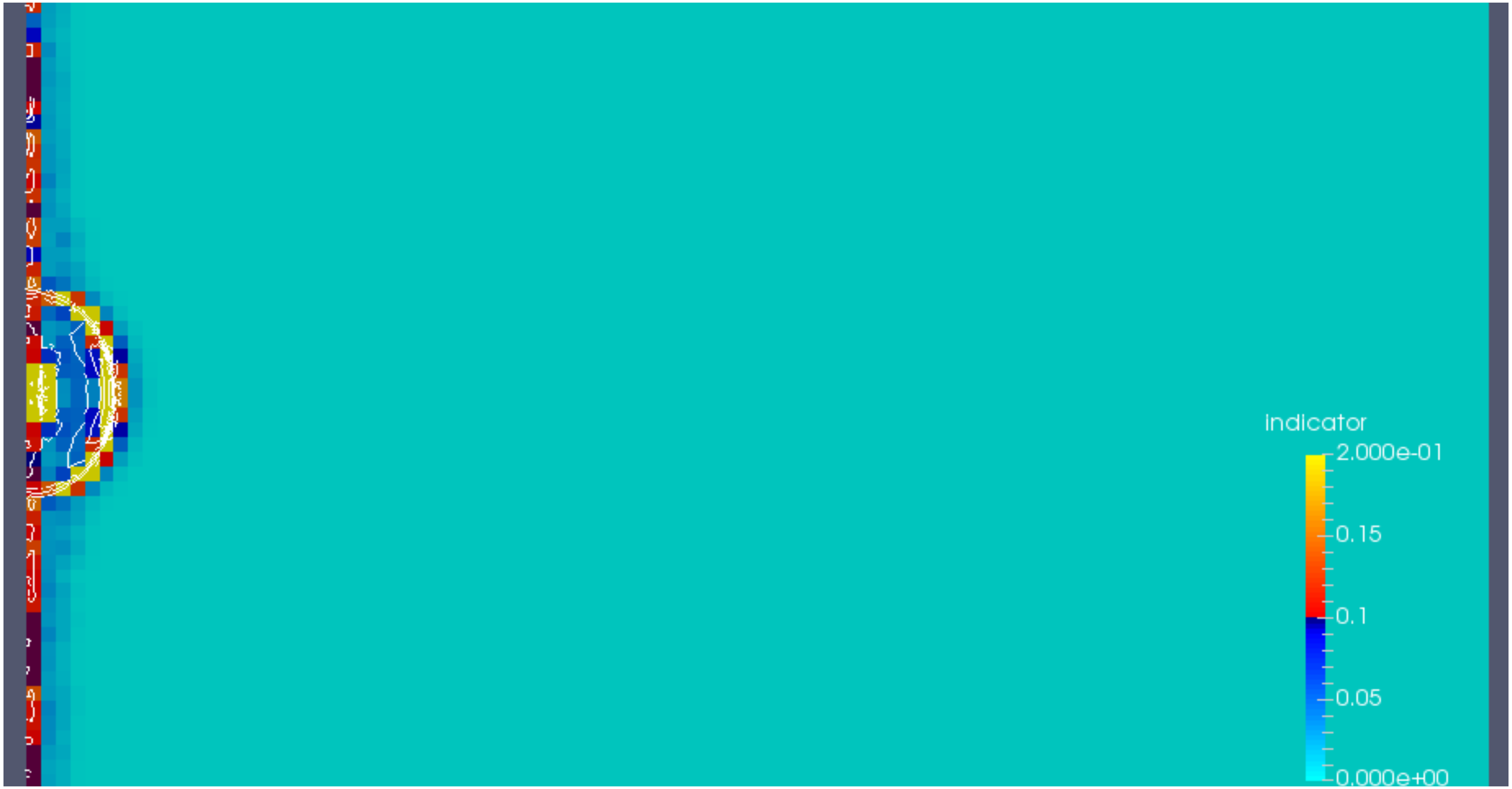
Spectral Convergence for Linear Problem



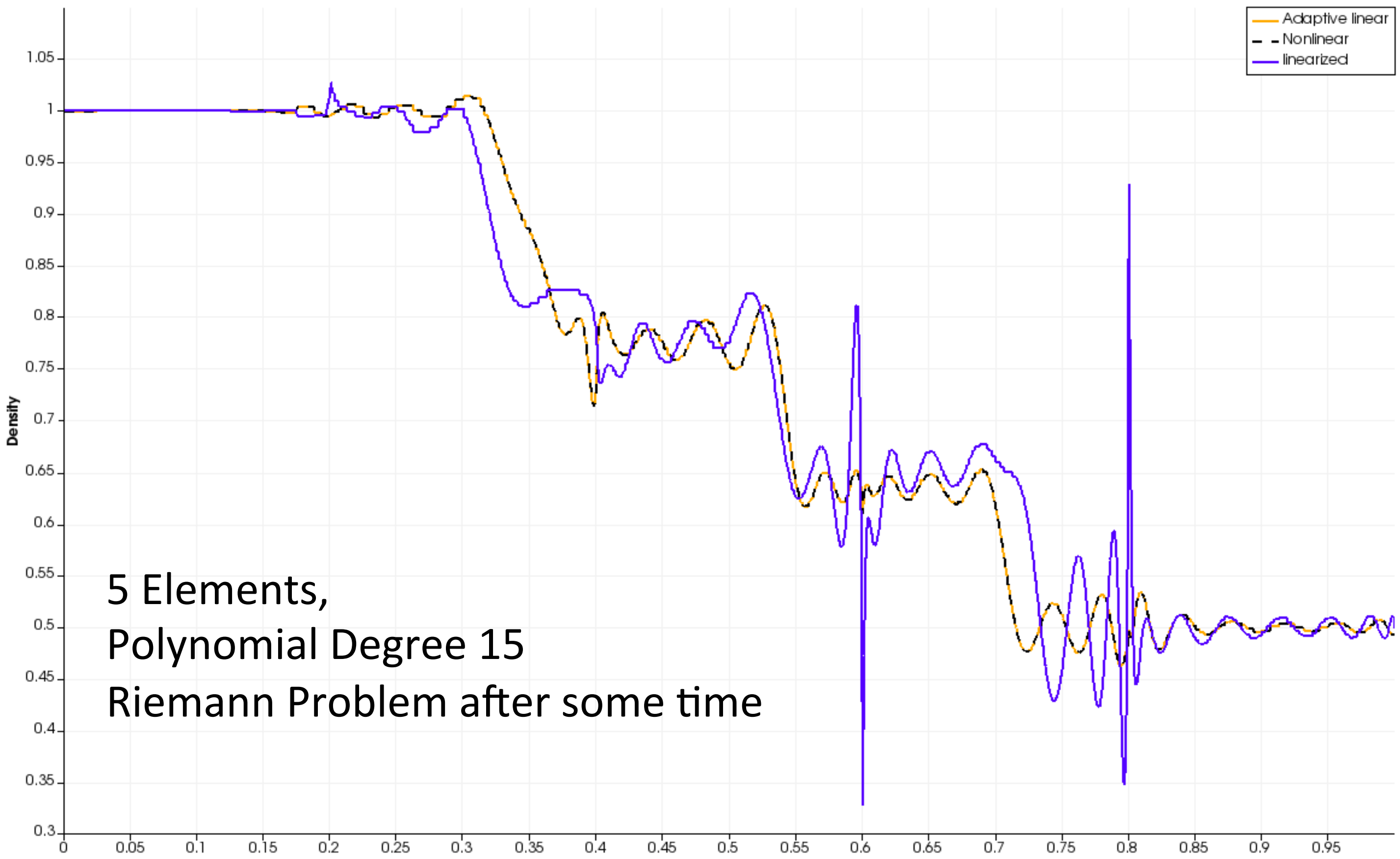
Adaptive Linearization

- With this local linearization approach it is simple to switch to linear flux computations dynamically at runtime
- Just need an indicator to decide which equations to use
  - For now we just use the variation of energy to decide whether to use linearized fluxes is acceptable
- Introduces load imbalance

Adaptive Linearization with Indicator



Riemann: Linear, Adaptively Linear, Nonlinear



Thank you.