Entropy stable discontinuous Galerkin methods with arbitrary bases and quadratures

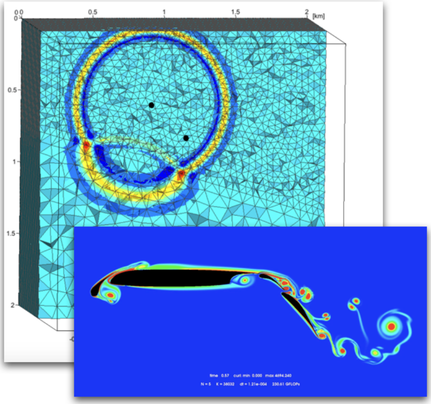
Jesse Chan

[1]Department of Computational and Applied Math

ICOSAHOM 2018
July 25, 2018

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).

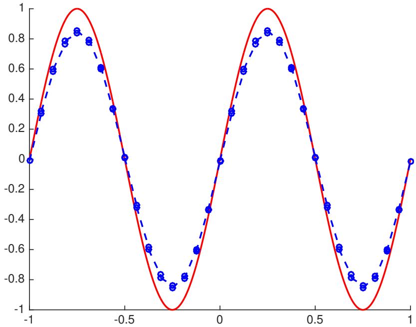


Figures courtesy of T. Warburton, A. Modave.

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).

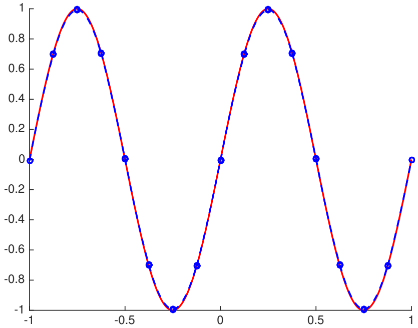


Fine linear approximation.

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).



Coarse quadratic approximation.

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).

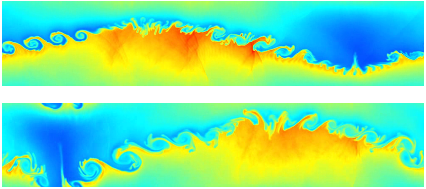


Figure from Per-Olof Persson.

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).

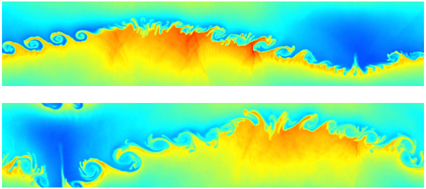
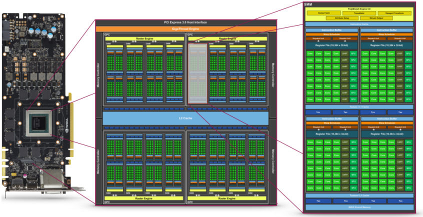


Figure from Per-Olof Persson.

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).

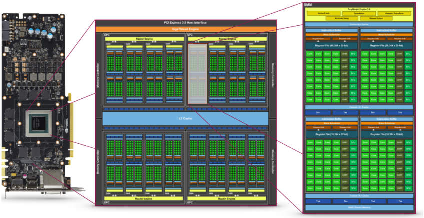


A graphics processing unit (GPU).

Goal: address instability of high order methods!

High order methods for time-dependent hyperbolic PDEs

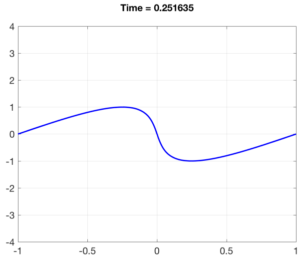
- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).



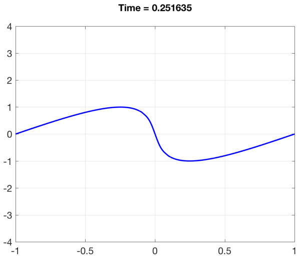
A graphics processing unit (GPU).

Goal: address instability of high order methods!

Why are high order methods for nonlinear PDEs unstable?



(a) $N = 7, K = 8$ (aligned mesh)



(b) $N = 7, K = 9$ (non-aligned mesh)

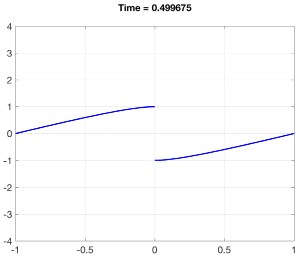
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0, \qquad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$

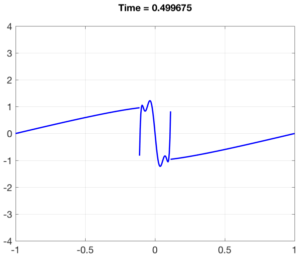
- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

$$\int_{D^k} \left(\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x} P_N u^2 \right) v \, \mathrm{d}x = 0, \qquad \frac{1}{2}\frac{\partial P_N u^2}{\partial x} \neq P_N \left(u \frac{\partial u}{\partial x} \right)$$

Why are high order methods for nonlinear PDEs unstable?



(a) $N = 7, K = 8$ (aligned mesh)



(b) $N = 7, K = 9$ (non-aligned mesh)

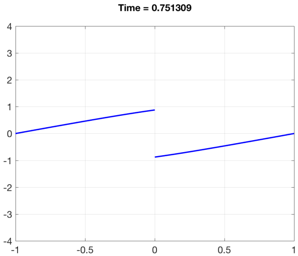
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0, \qquad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$

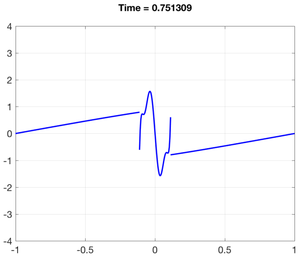
- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

$$\int_{D^k} \left(\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x} P_N u^2 \right) v \,\mathrm{d}x = 0, \qquad \frac{1}{2}\frac{\partial P_N u^2}{\partial x} \neq P_N\left(u \frac{\partial u}{\partial x} \right)$$

Why are high order methods for nonlinear PDEs unstable?



(a) $N = 7, K = 8$ (aligned mesh)



(b) $N = 7, K = 9$ (non-aligned mesh)

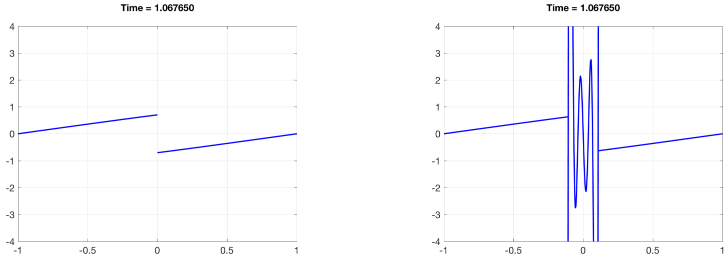
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0, \qquad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$

- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

$$\int_{D^k} \left(\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x} P_N u^2 \right) v \, \mathrm{d}x = 0, \qquad \frac{1}{2}\frac{\partial P_N u^2}{\partial x} \neq P_N \left(u \frac{\partial u}{\partial x} \right)$$

Why are high order methods for nonlinear PDEs unstable?



(a) $N = 7, K = 8$ (aligned mesh) (b) $N = 7, K = 9$ (non-aligned mesh)

- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0, \qquad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$

- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

$$\int_{D^k} \left(\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x} P_N u^2 \right) v \, \mathrm{d}x = 0, \qquad \frac{1}{2}\frac{\partial P_N u^2}{\partial x} \neq P_N \left(u \frac{\partial u}{\partial x} \right)$$

Entropy stability for nonlinear conservation laws

- Analogue of energy stability for nonlinear systems of conservation laws (Burgers', shallow water, compressible Euler, MHD).

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = 0.$$

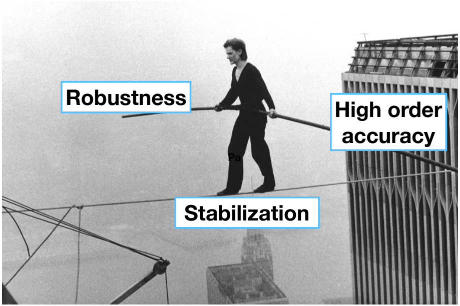
- Continuous entropy inequality: convex entropy function $S(\boldsymbol{u})$ and "entropy potential" $\psi(\boldsymbol{u})$.

$$\int_\Omega \boldsymbol{v}^T \left(\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x}\right) = 0, \qquad \boldsymbol{v} = \frac{\partial S}{\partial \boldsymbol{u}}$$

$$\Longrightarrow \int_\Omega \frac{\partial S(\boldsymbol{u})}{\partial t} + \left(\boldsymbol{v}^T \boldsymbol{f}(\boldsymbol{u}) - \psi(\boldsymbol{u})\right)\Big|_{-1}^{1} \leq 0.$$

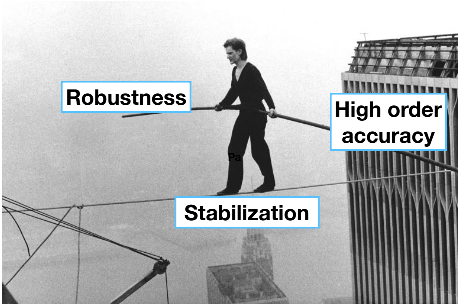
- Proof of entropy inequality relies on integration by parts, chain rule.

Why discretely entropy stable (ES) schemes?



- Existing discrete stability theory: regularization, viscosity, TVD, etc.
- Can result in a balancing act between high order accuracy, stability, and robustness.
- Goal: aim for stability independently of artificial viscosity, limiters, and quadrature accuracy.

Why discretely entropy stable (ES) schemes?



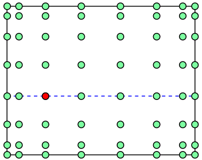
- Existing discrete stability theory: regularization, viscosity, TVD, etc.
- Can result in a balancing act between high order accuracy, stability, and robustness.
- Goal: aim for stability independently of artificial viscosity, limiters, and quadrature accuracy.

Talk outline

1. "Decoupled" summation by parts operators
2. Entropy stable formulations and flux differencing
3. Numerical experiments: triangles and tetrahedra
4. Entropy stable Gauss collocation methods: preliminary results

Talk outline

Overview of entropy stable high order SBP schemes



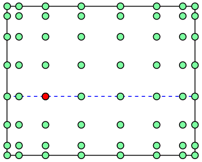
(a) GLL collocation

- Discrete entropy inequality for SBP schemes (e.g. GLL collocation).
- GSBP (e.g. Gauss collocation): higher accuracy, but require non-compact coupling conditions between neighboring elements.
- Tetrahedra, prisms, pyramids, etc (over-integration, dense norms)?

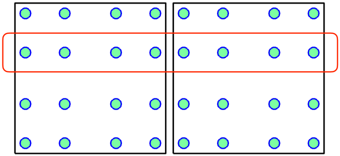
Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher, Carpenter, Nordström, Yamaleev, Swanson (2013), Fisher, Carpenter (2013), Gassner, Winters, and Kopriva (2016), Wintermeyer et al. (2017), Chen and Shu (2017), Crean, Hicken, DCDR Fernandez, et al. (2018), and more ...

Overview of entropy stable high order SBP schemes



(a) GLL collocation



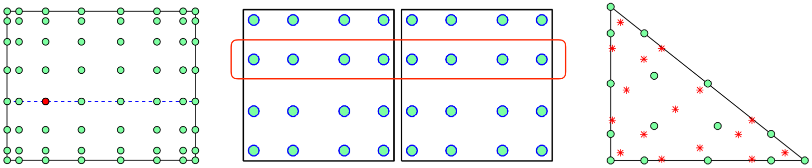
(b) Gauss nodes coupling

- Discrete entropy inequality for SBP schemes (e.g. GLL collocation).
- GSBP (e.g. Gauss collocation): higher accuracy, but require non-compact coupling conditions between neighboring elements.
- Tetrahedra, prisms, pyramids, etc (over-integration, dense norms)?

Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher, Carpenter, Nordström, Yamaleev, Swanson (2013), Fisher, Carpenter (2013), Gassner, Winters, and Kopriva (2016), Wintermeyer et al. (2017), Chen and Shu (2017), Crean, Hicken, DCDR Fernandez, et al. (2018), and more ...

Overview of entropy stable high order SBP schemes



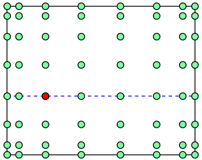
(a) GLL collocation (b) Gauss nodes coupling (c) Nodes vs cubature

- Discrete entropy inequality for SBP schemes (e.g. GLL collocation).
- GSBP (e.g. Gauss collocation): higher accuracy, but require non-compact coupling conditions between neighboring elements.
- Tetrahedra, prisms, pyramids, etc (over-integration, dense norms)?

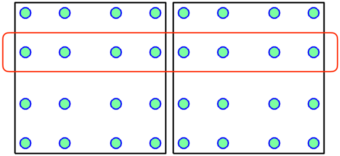
Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher, Carpenter, Nordström, Yamaleev, Swanson (2013), Fisher, Carpenter (2013), Gassner, Winters, and Kopriva (2016), Wintermeyer et al. (2017), Chen and Shu (2017), Crean, Hicken, DCDR Fernandez, et al. (2018), and more . . .

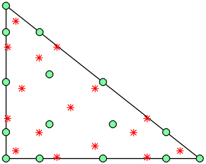
Overview of entropy stable high order SBP schemes



(a) GLL collocation



(b) Gauss nodes coupling



(c) Nodes vs cubature

- Discrete entropy inequality for SBP schemes (e.g. GLL collocation).
- GSBP (e.g. Gauss collocation): higher accuracy, but require non-compact coupling conditions between neighboring elements.
- Tetrahedra, prisms, pyramids, etc (over-integration, dense norms)?

Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher, Carpenter, Nordström, Yamaleev, Swanson (2013), Fisher, Carpenter (2013), Gassner, Winters, and Kopriva (2016), Wintermeyer et al. (2017), Chen and Shu (2017), Crean, Hicken, DCDR Fernandez, et al. (2018), and more . . .

Quadrature-based matrices for polynomial bases

- Volume and surface quadratures $(\boldsymbol{x}_i^q, \boldsymbol{w}_i^q)$, $(\boldsymbol{x}_i^f, \boldsymbol{w}_i^f)$, exact for degree $2N$ polynomials. Define diagonal quadrature weight matrices

$$\boldsymbol{W} = \operatorname{diag}\left(\boldsymbol{w}^q\right), \qquad \boldsymbol{W}_f = \operatorname{diag}\left(\boldsymbol{w}^f\right).$$

- Assume some polynomial basis $\phi_1, \ldots, \phi_{N_p}$. Define the interpolation matrices $\boldsymbol{V}_q$, $\boldsymbol{V}_f$

$$\left(\boldsymbol{V}_q\right)_{ij} = \phi_j(\boldsymbol{x}_i^q), \qquad \left(\boldsymbol{V}_f\right)_{ij} = \phi_j(\boldsymbol{x}_i^f).$$

- Introduce quadrature-based L^2 projection and lifting matrices

$$\boldsymbol{P}_q = \boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}, \qquad \boldsymbol{L}_f = \boldsymbol{M}^{-1}\boldsymbol{V}_f^T\boldsymbol{W}_f.$$

- These matrices map to and from modal and quadrature spaces.

Quadrature-based differentiation matrices

- Matrix $\boldsymbol{D}_q^i$: evaluates derivative of L^2 projection at points $\boldsymbol{x}^q$.

$$\boldsymbol{D}_q^i = \boldsymbol{V}_q \boldsymbol{D}^i \boldsymbol{P}_q, \qquad \boldsymbol{D}^i = \text{ modal differentiation matrix.}$$

- Summation-by-parts involving L^2 projection:

$$\boldsymbol{W}\boldsymbol{D}_q^i + \left(\boldsymbol{W}\boldsymbol{D}_q^i\right)^T = \left(\boldsymbol{V}_f \boldsymbol{P}_q\right)^T \boldsymbol{W}_f \operatorname{diag}\left(\boldsymbol{n}_i\right) \boldsymbol{V}_f \boldsymbol{P}_q.$$

- Equivalent to integration-by-parts + quadrature: for $u, v \in L^2\left(\widehat{D}\right)$

$$\int_{\widehat{D}} \frac{\partial P_N u}{\partial x_i} v + \int_{\widehat{D}} u \frac{\partial P_N v}{\partial x_i} = \int_{\partial \widehat{D}} \left(P_N u\right)\left(P_N v\right) \widehat{n}_i$$

- Recovers GSBP, but entropy stable interface terms are expensive.

A "decoupled" block SBP operator

- Approx. derivatives also using boundary traces (compact coupling).
- On an element D^k with unit normal vector $\boldsymbol{n}$: approximate ith derivative (block matrix operating on volume + surface values).

$$\boldsymbol{D}_N^i = \begin{bmatrix} \boldsymbol{D}_q^i - \frac{1}{2}\boldsymbol{V}_q\boldsymbol{L}_f\operatorname{diag}(\boldsymbol{n}_i)\boldsymbol{V}_f\boldsymbol{P}_q & \frac{1}{2}\boldsymbol{V}_q\boldsymbol{L}_f\operatorname{diag}(\boldsymbol{n}_i) \\ -\frac{1}{2}\operatorname{diag}(\boldsymbol{n}_i)\boldsymbol{V}_f\boldsymbol{P}_q & \frac{1}{2}\operatorname{diag}(\boldsymbol{n}_i) \end{bmatrix},$$

- $\boldsymbol{D}_N^i$ satisfies a summation-by-parts (SBP) property + $\boldsymbol{D}_N^i\boldsymbol{1} = 0$

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{W} & \\ & \boldsymbol{W}_f \end{bmatrix}\boldsymbol{D}_N^i, \qquad \boldsymbol{B}_N = \begin{bmatrix} 0 & \\ & \boldsymbol{W}_f\boldsymbol{n}_i \end{bmatrix},$$

$$\boxed{\boldsymbol{Q}_N^i + \left(\boldsymbol{Q}_N^i\right)^T = \boldsymbol{B}_N} \sim \boxed{\int_{D^k} \frac{\partial f}{\partial x_i}g + f\frac{\partial g}{\partial x_i} = \int_{\partial D^k} fg\boldsymbol{n}_i}.$$

Chen and Shu (2017). *ES high order DG methods with suitable quadrature rules for hyperbolic conservation laws.*

Differentiation using decoupled SBP operators

- Note: $\boldsymbol{D}_N^i$ is not a differentiation matrix on its own.
- $\boldsymbol{P}_q, \boldsymbol{L}_f$, and $\boldsymbol{D}_N^i$ produce a high order polynomial approximation of $f\frac{\partial g}{\partial x}$ given data at quadrature points $\boldsymbol{x} = [\boldsymbol{x}^q, \boldsymbol{x}^f]$.

$$f\frac{\partial g}{\partial x} \approx \begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix} \operatorname{diag}(\boldsymbol{f})\, \boldsymbol{D}_N \boldsymbol{g}, \qquad \boldsymbol{f}_i, \boldsymbol{g}_i = f(\boldsymbol{x}_i), g(\boldsymbol{x}_i).$$

- Equivalent to solving variational problem for $u(\boldsymbol{x}) \approx f\frac{\partial g}{\partial x}$

$$\int_{D^k} u(\boldsymbol{x}) v(\boldsymbol{x}) = \int_{D^k} f\frac{\partial P_N g}{\partial x} v + \int_{\partial D^k} (f - P_N f)\frac{(gv + P_N(gv))}{2}.$$

- $\boldsymbol{D}_N^i \mathbf{1} = 0$ holds (necessary for discrete entropy conservation).

Talk outline

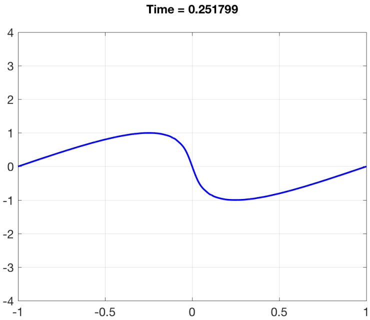
1 “Decoupled” summation by parts operators

2 Entropy stable formulations and flux differencing

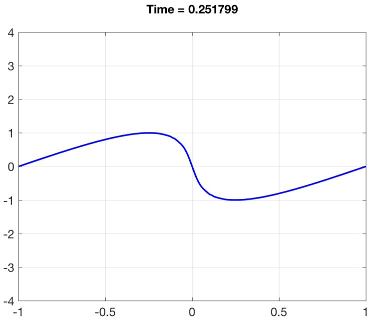
3 Numerical experiments: triangles and tetrahedra

4 Entropy stable Gauss collocation methods: preliminary results

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

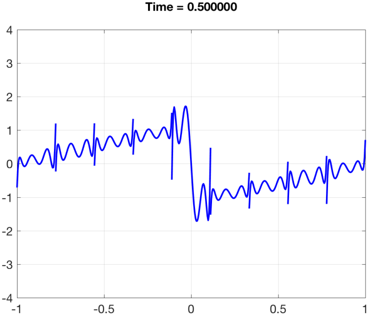


(b) Energy stable

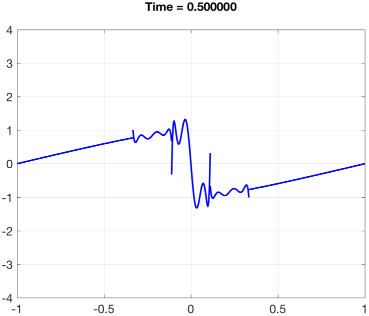
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.}, \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

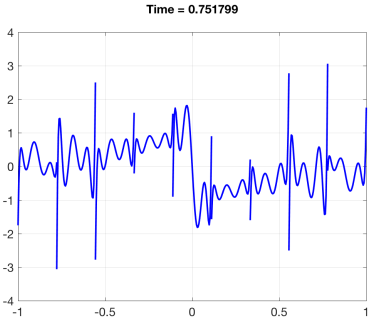


(b) Energy stable

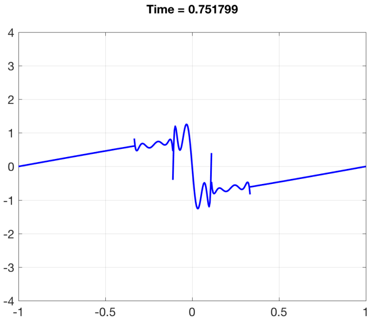
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.}, \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \mathrm{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

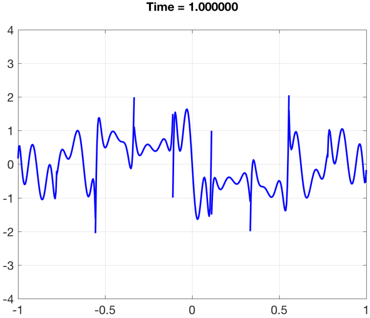


(b) Energy stable

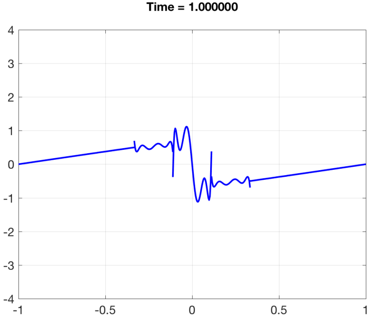
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.}, \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

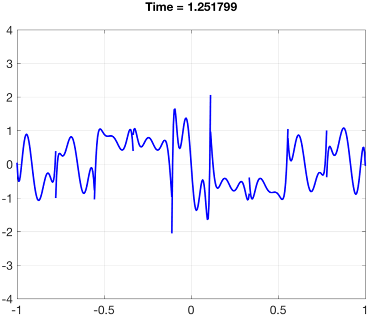


(b) Energy stable

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.,} \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$

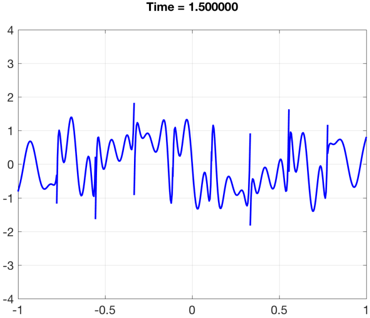


(a) Energy conservative (b) Energy stable

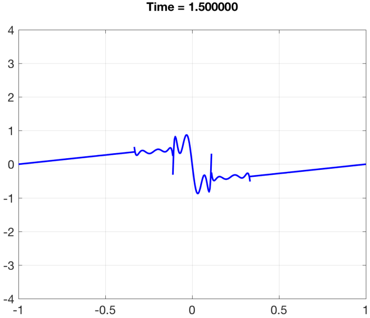
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.}, \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \mathrm{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

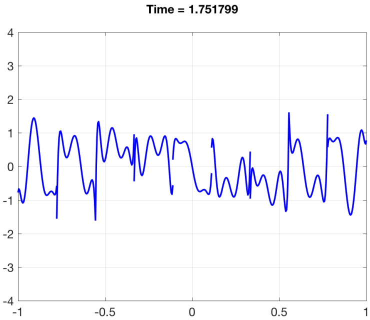


(b) Energy stable

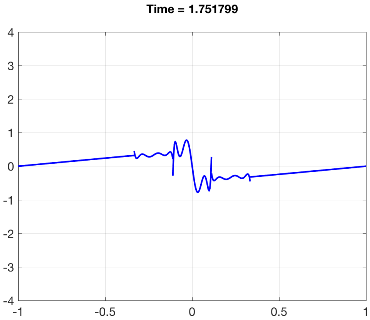
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.,} \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative

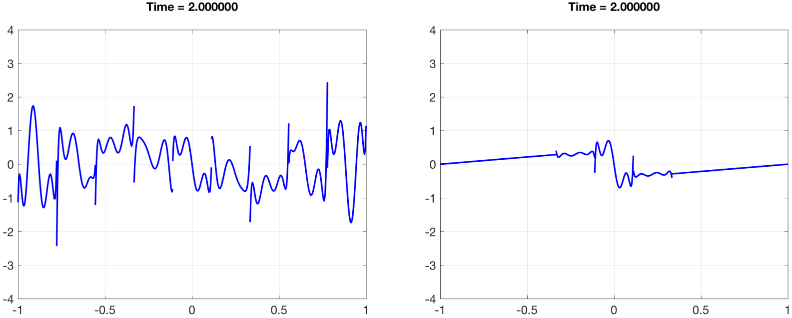


(b) Energy stable

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.,} \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$

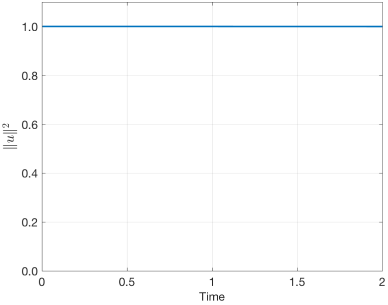


(a) Energy conservative (b) Energy stable

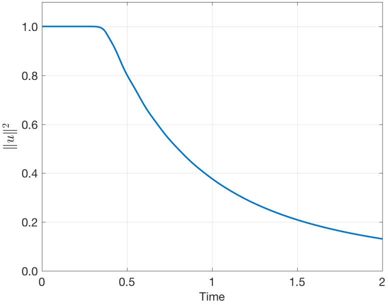
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.}, \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix}\left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \mathrm{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Split form of Burgers': $\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right) = 0$



(a) Energy conservative



(b) Energy stable

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \quad \widehat{\boldsymbol{u}} = \text{modal coeffs.,} \qquad \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{3}\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_f \end{bmatrix} \left(\boldsymbol{D}_N\left(\boldsymbol{u}^2\right) + \operatorname{diag}\left(\boldsymbol{u}\right)\boldsymbol{D}_N\boldsymbol{u}\right) + \boldsymbol{L}_f\left(\boldsymbol{f}^*(u^+, u)\right) = 0.$$

Flux differencing: entropy conservative finite volume fluxes

- Tadmor's entropy conservative (mean value) numerical flux

$$\boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{f}_S(\boldsymbol{v},\boldsymbol{u}), \qquad \text{(consistency, symmetry)}$$

$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \psi_L - \psi_R, \qquad \text{(conservation).}$$

- Flux differencing for Burgers' equation: let $u_L = u(x), u_R = u(y)$

$$f_S(u_L,u_R) = \frac{1}{6}\left(u_L^2 + u_L u_R + u_R^2\right),$$

- Beyond split formulations: mass flux for compressible Euler

$$f_S^{\rho}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \{\{\rho\}\}^{\log}\{\{u\}\}, \qquad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log\rho_L - \log\rho_R}.$$

Tadmor, Eitan (1987). *The numerical viscosity of entropy stable schemes for systems of conservation laws. I.*

Chandrashekar (2013). *Kinetic energy preserving and entropy stable FV schemes for compressible Euler and NS equations.*

Flux differencing: entropy conservative finite volume fluxes

- Tadmor's entropy conservative (mean value) numerical flux

$$\boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{f}_S(\boldsymbol{v},\boldsymbol{u}), \qquad \text{(consistency, symmetry)}$$

$$\left(\boldsymbol{v}_L - \boldsymbol{v}_R\right)^T \boldsymbol{f}\left(\boldsymbol{u}_L,\boldsymbol{u}_R\right) = \psi_L - \psi_R, \qquad \text{(conservation).}$$

- Flux differencing for Burgers' equation: let $u_L = u(x), u_R = u(y)$

$$f_S(u_L,u_R) = \frac{1}{6}\left(u_L^2 + u_L u_R + u_R^2\right),$$

$$\frac{\partial f(u)}{\partial x} \Longrightarrow 2\frac{\partial f_S\left(u(x),u(y)\right)}{\partial x}\bigg|_{y=x} = \frac{1}{3}\frac{\partial u^2}{\partial x} + \frac{1}{3}u\frac{\partial u}{\partial x} + \frac{1}{3}u^2\cancel{\frac{\partial 1}{\partial x}}.$$

- Beyond split formulations: mass flux for compressible Euler

$$f_S^{\rho}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \{\{\rho\}\}^{\log}\{\{u\}\}, \qquad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log\rho_L - \log\rho_R}.$$

Tadmor, Eitan (1987). *The numerical viscosity of entropy stable schemes for systems of conservation laws. I.*

Chandrashekar (2013). *Kinetic energy preserving and entropy stable FV schemes for compressible Euler and NS equations.*

Flux differencing: entropy conservative finite volume fluxes

- Tadmor's entropy conservative (mean value) numerical flux

$$\boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{f}_S(\boldsymbol{v},\boldsymbol{u}), \qquad \text{(consistency, symmetry)}$$
$$\left(\boldsymbol{v}_L - \boldsymbol{v}_R\right)^T \boldsymbol{f}\left(\boldsymbol{u}_L,\boldsymbol{u}_R\right) = \psi_L - \psi_R, \qquad \text{(conservation).}$$

- Flux differencing for Burgers' equation: let $u_L = u(x), u_R = u(y)$

$$f_S(u(x),u(y)) = \frac{1}{6}\left(u(x)^2 + u(x)u(y) + u(y)^2\right),$$

$$\frac{\partial f(u)}{\partial x} \Longrightarrow 2\frac{\partial f_S\left(u(x),u(y)\right)}{\partial x}\bigg|_{y=x} = \frac{1}{3}\frac{\partial u^2}{\partial x} + \frac{1}{3}u\frac{\partial u}{\partial x} + \frac{1}{3}u^2\cancel{\frac{\partial 1}{\partial x}}.$$

- Beyond split formulations: mass flux for compressible Euler

$$f_S^{\rho}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \{\{\rho\}\}^{\log}\{\{u\}\}, \qquad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log\rho_L - \log\rho_R}.$$

Tadmor, Eitan (1987). *The numerical viscosity of entropy stable schemes for systems of conservation laws. I.*

Chandrashekar (2013). *Kinetic energy preserving and entropy stable FV schemes for compressible Euler and NS equations.*

Flux differencing: entropy conservative finite volume fluxes

- Tadmor's entropy conservative (mean value) numerical flux

$$\boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{f}_S(\boldsymbol{v},\boldsymbol{u}), \qquad \text{(consistency, symmetry)}$$
$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \psi_L - \psi_R, \qquad \text{(conservation).}$$

- Flux differencing for Burgers' equation: let $u_L = u(x), u_R = u(y)$

$$f_S(u(x),u(y)) = \frac{1}{6}\left(u(x)^2 + u(x)u(y) + u(y)^2\right),$$
$$\frac{\partial f(u)}{\partial x} \Longrightarrow 2\frac{\partial f_S(u(x),u(y))}{\partial x}\bigg|_{y=x} = \frac{1}{3}\frac{\partial u^2}{\partial x} + \frac{1}{3}u\frac{\partial u}{\partial x} + \frac{1}{3}u^2\cancel{\frac{\partial 1}{\partial x}}.$$

- Beyond split formulations: mass flux for compressible Euler

$$f_S^\rho(\boldsymbol{u}_L,\boldsymbol{u}_R) = \{\{\rho\}\}^{\log}\{\{u\}\}, \qquad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log\rho_L - \log\rho_R}.$$

Tadmor, Eitan (1987). *The numerical viscosity of entropy stable schemes for systems of conservation laws. I.*

Chandrashekar (2013). *Kinetic energy preserving and entropy stable FV schemes for compressible Euler and NS equations.*

Flux differencing: entropy conservative finite volume fluxes

- Tadmor's entropy conservative (mean value) numerical flux

$$\boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}), \qquad \boldsymbol{f}_S(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{f}_S(\boldsymbol{v},\boldsymbol{u}), \qquad \text{(consistency, symmetry)}$$

$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \psi_L - \psi_R, \qquad \text{(conservation)}.$$

- Flux differencing for Burgers' equation: let $u_L = u(x), u_R = u(y)$

$$f_S(u(x),u(y)) = \frac{1}{6}\left(u(x)^2 + u(x)u(y) + u(y)^2\right),$$

$$\frac{\partial f(u)}{\partial x} \Longrightarrow 2\left.\frac{\partial f_S(u(x),u(y))}{\partial x}\right|_{y=x} = \frac{1}{3}\frac{\partial u^2}{\partial x} + \frac{1}{3}u\frac{\partial u}{\partial x} + \frac{1}{3}u^2\cancel{\frac{\partial 1}{\partial x}}.$$

- Beyond split formulations: mass flux for compressible Euler

$$f_S^{\rho}(\boldsymbol{u}_L,\boldsymbol{u}_R) = \{\{\rho\}\}^{\log}\{\{u\}\}, \qquad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log\rho_L - \log\rho_R}.$$

Tadmor, Eitan (1987). *The numerical viscosity of entropy stable schemes for systems of conservation laws. I.*

Chandrashekar (2013). *Kinetic energy preserving and entropy stable FV schemes for compressible Euler and NS equations.*

Flux differencing: implementational details

- Define $\boldsymbol{F}_S$ as evaluation of $\boldsymbol{f}_S$ at all combinations of quadrature points

$$(\boldsymbol{F}_S)_{ij} = \boldsymbol{f}_S\left(u(\boldsymbol{x}_i), u(\boldsymbol{x}_j)\right), \qquad \boldsymbol{x} = \left[\boldsymbol{x}^q, \boldsymbol{x}^f\right]^T.$$

- Replace $\frac{\partial}{\partial x}$ with $\boldsymbol{D}_N$ + projection and lifting matrices.

$$2\frac{\partial f_S\left(u(x), u(y)\right)}{\partial x}\bigg|_{y=x} \Longrightarrow \left[\begin{array}{cc} \boldsymbol{P}_q & \boldsymbol{L}_f \end{array}\right] \operatorname{diag}(2\boldsymbol{D}_N\boldsymbol{F}_S).$$

- Efficient Hadamard product reformulation of flux differencing (efficient on-the-fly evaluation of $\boldsymbol{F}_S$)

$$\operatorname{diag}(2\boldsymbol{D}_N\boldsymbol{F}_S) = (2\boldsymbol{D}_N \circ \boldsymbol{F}_S)\,\mathbf{1}.$$

Flux differencing: avoiding the chain rule

- Test $(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1}$ with entropy variables $\widetilde{\boldsymbol{v}}$, integrate, use SBP:

$$\widetilde{\boldsymbol{v}}^T (2\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1} = \widetilde{\boldsymbol{v}}^T \left(\left(\begin{bmatrix} 0 & \\ & \boldsymbol{W}_f \boldsymbol{n} \end{bmatrix} + \boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right) \circ \boldsymbol{F}_S\right)\mathbf{1}.$$

- Only boundary terms appear in final estimate; volume terms become boundary terms using properties of $(\boldsymbol{F}_S)_{ij} = \boldsymbol{f}_S(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j)$

$$\begin{aligned}\widetilde{\boldsymbol{v}}^T \left(\left(\boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right) \circ \boldsymbol{F}_S\right)\mathbf{1} &= \widetilde{\boldsymbol{v}}^T (\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1} - \mathbf{1}^T (\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\widetilde{\boldsymbol{v}} \\ &= \sum_{i,j} (\boldsymbol{Q}_N)_{ij} \left(\widetilde{\boldsymbol{v}}_i - \widetilde{\boldsymbol{v}}_j\right)^T \boldsymbol{f}_S(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j).\end{aligned}$$

- Applying Tadmor shuffle condition requires $\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}})$; the entropy variables $\widetilde{\boldsymbol{v}}$ must be a function of the conservative variables $\widetilde{\boldsymbol{u}}$.

Flux differencing: avoiding the chain rule

- Test $(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1}$ with entropy variables $\widetilde{\boldsymbol{v}}$, integrate, use SBP:

$$\widetilde{\boldsymbol{v}}^T (2\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1} = \widetilde{\boldsymbol{v}}^T \left(\left(\begin{bmatrix} 0 & \\ & \boldsymbol{W}_f \boldsymbol{n} \end{bmatrix} + \boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right) \circ \boldsymbol{F}_S\right)\mathbf{1}.$$

- Only boundary terms appear in final estimate; volume terms become boundary terms using properties of $(\boldsymbol{F}_S)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right)$

$$\begin{aligned}\widetilde{\boldsymbol{v}}^T \left(\left(\boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right) \circ \boldsymbol{F}_S\right)\mathbf{1} &= \widetilde{\boldsymbol{v}}^T (\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1} - \mathbf{1}^T (\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\widetilde{\boldsymbol{v}} \\ &= \sum_{i,j} (\boldsymbol{Q}_N)_{ij} \left(\psi(\widetilde{\boldsymbol{u}}_i) - \psi(\widetilde{\boldsymbol{u}}_j)\right).\end{aligned}$$

- Applying Tadmor shuffle condition requires $\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}})$; the entropy variables $\widetilde{\boldsymbol{v}}$ must be a function of the conservative variables $\widetilde{\boldsymbol{u}}$.

Flux differencing: avoiding the chain rule

- Test $\left(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1}$ with entropy variables $\widetilde{\boldsymbol{v}}$, integrate, use SBP:

$$\widetilde{\boldsymbol{v}}^T\left(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} = \widetilde{\boldsymbol{v}}^T\left(\left(\begin{bmatrix} 0 & \\ & \boldsymbol{W}_f\boldsymbol{n} \end{bmatrix} + \boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1}.$$

- Only boundary terms appear in final estimate; volume terms become boundary terms using properties of $\left(\boldsymbol{F}_S\right)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right)$

$$\begin{aligned}\widetilde{\boldsymbol{v}}^T\left(\left(\boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1} &= \widetilde{\boldsymbol{v}}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} - \mathbf{1}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\widetilde{\boldsymbol{v}}\\ &= \mathbf{1}^T\boldsymbol{Q}_N\psi - \psi^T\boldsymbol{Q}_N\mathbf{1} = \mathbf{1}^T\boldsymbol{Q}_N\psi\end{aligned}$$

- Applying Tadmor shuffle condition requires $\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}})$; the entropy variables $\widetilde{\boldsymbol{v}}$ must be a function of the conservative variables $\widetilde{\boldsymbol{u}}$.

Flux differencing: avoiding the chain rule

- Test $(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S)\,\mathbf{1}$ with entropy variables $\widetilde{\boldsymbol{v}}$, integrate, use SBP:

$$\widetilde{\boldsymbol{v}}^T\left(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} = \widetilde{\boldsymbol{v}}^T\left(\left(\begin{bmatrix} 0 & \\ & \boldsymbol{W}_f\boldsymbol{n} \end{bmatrix} + \boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1}.$$

- Only boundary terms appear in final estimate; volume terms become boundary terms using properties of $\left(\boldsymbol{F}_S\right)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right)$

$$\widetilde{\boldsymbol{v}}^T\left(\left(\boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1} = \widetilde{\boldsymbol{v}}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} - \mathbf{1}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\widetilde{\boldsymbol{v}}$$
$$= \mathbf{1}^T\left(\boldsymbol{B}_N - \boldsymbol{Q}_N^T\right)\psi = \mathbf{1}^T\boldsymbol{B}_N\psi.$$

- Applying Tadmor shuffle condition requires $\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}})$; the entropy variables $\widetilde{\boldsymbol{v}}$ must be a function of the conservative variables $\widetilde{\boldsymbol{u}}$.

Flux differencing: avoiding the chain rule

- Test $\left(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1}$ with entropy variables $\widetilde{\boldsymbol{v}}$, integrate, use SBP:

$$\widetilde{\boldsymbol{v}}^T\left(2\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} = \widetilde{\boldsymbol{v}}^T\left(\left(\begin{bmatrix} 0 & \\ & \boldsymbol{W}_f\boldsymbol{n} \end{bmatrix} + \boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1}.$$

- Only boundary terms appear in final estimate; volume terms become boundary terms using properties of $\left(\boldsymbol{F}_S\right)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right)$

$$\begin{aligned}\widetilde{\boldsymbol{v}}^T\left(\left(\boldsymbol{Q}_N - \boldsymbol{Q}_N^T\right)\circ \boldsymbol{F}_S\right)\mathbf{1} &= \widetilde{\boldsymbol{v}}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\mathbf{1} - \mathbf{1}^T\left(\boldsymbol{Q}_N \circ \boldsymbol{F}_S\right)\widetilde{\boldsymbol{v}} \\ &= \mathbf{1}^T\left(\boldsymbol{B}_N - \boldsymbol{Q}_N^T\right)\psi = \mathbf{1}^T\boldsymbol{B}_N\psi.\end{aligned}$$

- Applying Tadmor shuffle condition requires $\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}})$; the entropy variables $\widetilde{\boldsymbol{v}}$ must be a function of the conservative variables $\widetilde{\boldsymbol{u}}$.

Modifying the conservative variables

- Conservative variables $\boldsymbol{u}_h$ and test functions are polynomial, but the entropy variables $\boldsymbol{v}(\boldsymbol{u}_h) \notin P^N$!
- Evaluate flux $\boldsymbol{f}_S$ using modified conservative variables $\widetilde{\boldsymbol{u}}$

$$\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(P_N \boldsymbol{v}(\boldsymbol{u}_h)\right).$$

- If $\boldsymbol{v}(\boldsymbol{u})$ is an invertible mapping, this choice of $\widetilde{\boldsymbol{u}}$ ensures that

$$\widetilde{\boldsymbol{v}} = \boldsymbol{v}(\widetilde{\boldsymbol{u}}) = P_N \boldsymbol{v}(\boldsymbol{u}_h) \in P^N.$$

- Local conservation w.r.t. a generalized Lax-Wendroff theorem.

Parsani et al. (2016). *ES staggered grid disc. spectral collocation methods of any order for the comp. NS eqns.*

Hughes, Franca, and Mallet (1986). *A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics.*

Shi and Shu (2017). *On local conservation of numerical methods for conservation laws.*

A discretely entropy conservative DG method

Theorem (Chan 2018)

Let $\boldsymbol{u}_h(\boldsymbol{x},t) = \sum_j \widehat{\boldsymbol{u}}_j(t)\phi_j(\boldsymbol{x})$ *and* $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(\begin{bmatrix}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{bmatrix}\boldsymbol{P}_q\boldsymbol{v}\right)$. *Let* $\widehat{\boldsymbol{u}}$ *locally solve*

$$\boldsymbol{M}\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \sum_{i=1}^{d}\left[\begin{array}{c}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{array}\right]^T\left(2\boldsymbol{Q}_N^i \circ \boldsymbol{F}_S^i\right)\boldsymbol{1} + \boldsymbol{V}_f^T\boldsymbol{W}_f\left(\boldsymbol{f}_S^i(\widetilde{\boldsymbol{u}}^+,\widetilde{\boldsymbol{u}}) - \boldsymbol{f}^i(\widetilde{\boldsymbol{u}})\right)\boldsymbol{n}_i = 0.$$

Assuming continuity in time, $\boldsymbol{u}_h(\boldsymbol{x},t)$ *satisfies the quadrature form of*

$$\int_\Omega \frac{\partial S(\boldsymbol{u}_h)}{\partial t} + \sum_{i=1}^{d}\int_{\partial\Omega}\left((P_N\boldsymbol{v})^T\boldsymbol{f}^i(\widetilde{\boldsymbol{u}}) - \psi_i(\widetilde{\boldsymbol{u}})\right)\boldsymbol{n}_i = 0.$$

- Add interface dissipation (e.g. Lax-Friedrichs) for entropy inequality.

Parsani et al. (2016). *ES staggered grid disc. spectral collocation methods of any order for the comp. NS eqns.*

Shi and Shu (2017). *On local conservation of numerical methods for conservation laws.*

A discretely entropy conservative DG method

Theorem (Chan 2018)

Let $\boldsymbol{u}_h(\boldsymbol{x},t) = \sum_j \widehat{\boldsymbol{u}}_j(t)\phi_j(\boldsymbol{x})$ *and* $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(\begin{bmatrix}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{bmatrix}\boldsymbol{P}_q\boldsymbol{v}\right)$. *Let* $\widehat{\boldsymbol{u}}$ *locally solve*

$$\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \sum_{i=1}^{d}\begin{bmatrix}\boldsymbol{P}_q & \boldsymbol{L}_f\end{bmatrix}\left(2\boldsymbol{D}_N^i\circ\boldsymbol{F}_S^i\right)\mathbf{1} + \boldsymbol{L}_f\left(\boldsymbol{f}_S^i(\widetilde{\boldsymbol{u}}^+,\widetilde{\boldsymbol{u}}) - \boldsymbol{f}^i(\widetilde{\boldsymbol{u}})\right)\boldsymbol{n}_i = 0.$$

Assuming continuity in time, $\boldsymbol{u}_h(\boldsymbol{x},t)$ *satisfies the quadrature form of*

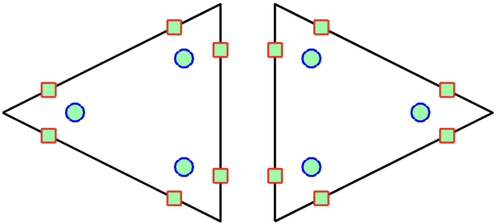
$$\int_\Omega \frac{\partial S(\boldsymbol{u}_h)}{\partial t} + \sum_{i=1}^{d}\int_{\partial\Omega}\left((P_N\boldsymbol{v})^T\boldsymbol{f}^i(\widetilde{\boldsymbol{u}}) - \psi_i(\widetilde{\boldsymbol{u}})\right)\boldsymbol{n}_i = 0.$$

- Add interface dissipation (e.g. Lax-Friedrichs) for entropy inequality.

Parsani et al. (2016). *ES staggered grid disc. spectral collocation methods of any order for the comp. NS eqns.*

Shi and Shu (2017). *On local conservation of numerical methods for conservation laws.*

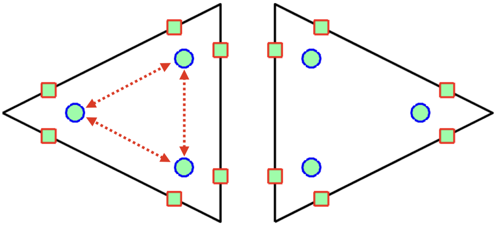
Illustration of main steps of ESDG



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix}\right) \mathbf{1}$$

- Interpolate projected entropy variables $P_N \boldsymbol{v}(\boldsymbol{u})$ to all nodes.
- Perform flux differencing at volume quadrature nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

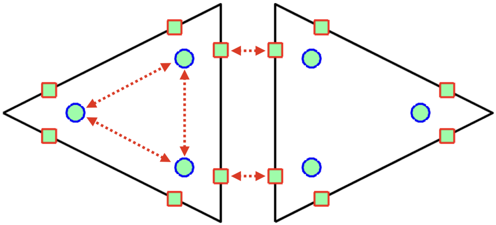
Illustration of main steps of ESDG



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix} \right) \mathbf{1}$$

- Interpolate projected entropy variables $P_N \boldsymbol{v}(\boldsymbol{u})$ to all nodes.
- Perform flux differencing at volume quadrature nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

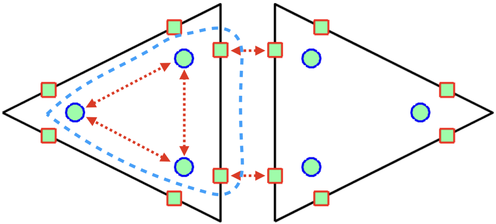
Illustration of main steps of ESDG



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \color{red}{\boldsymbol{C}} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \color{red}{\boldsymbol{F}_S^{ff}} \end{bmatrix}\right)\mathbf{1}$$

- Interpolate projected entropy variables $P_N \boldsymbol{v}(\boldsymbol{u})$ to all nodes.
- Perform flux differencing at volume quadrature nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

Illustration of main steps of ESDG

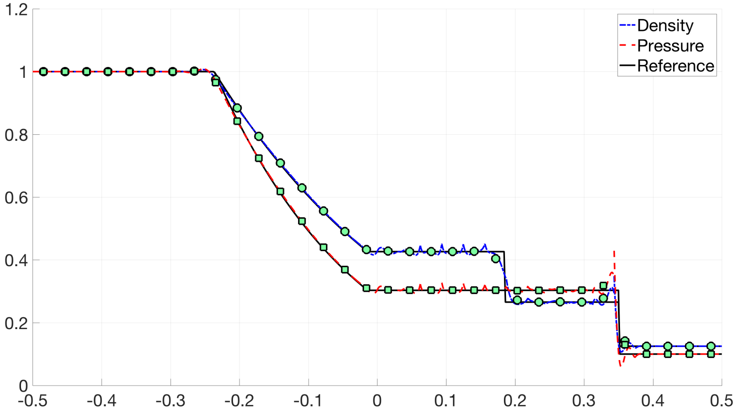


$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix} \right) \mathbf{1}$$

- Interpolate projected entropy variables $P_N \boldsymbol{v}(\boldsymbol{u})$ to all nodes.
- Perform flux differencing at volume quadrature nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

1D Sod shock tube

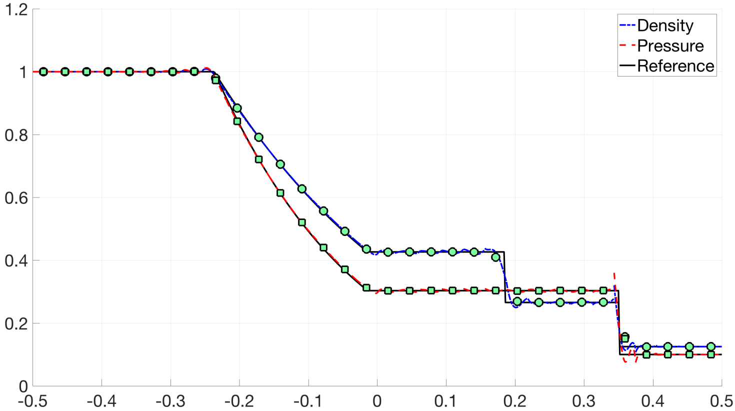
- Circles are cell averages.
- CFL of .125 used for both GLL-$(N+1)$and GQ-$(N+2)$.



$N = 4, K = 32$, $(N+1)$ point Gauss-Lobatto-Legendre quadrature.

1D Sod shock tube

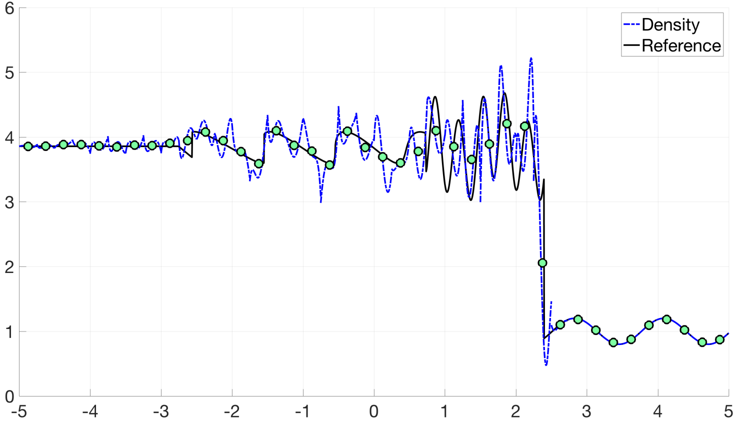
- Circles are cell averages.
- CFL of .125 used for both GLL-$(N+1)$and GQ-$(N+2)$.



$N=4, K=32$, $(N+2)$ point Gauss quadrature.

1D sine-shock interaction

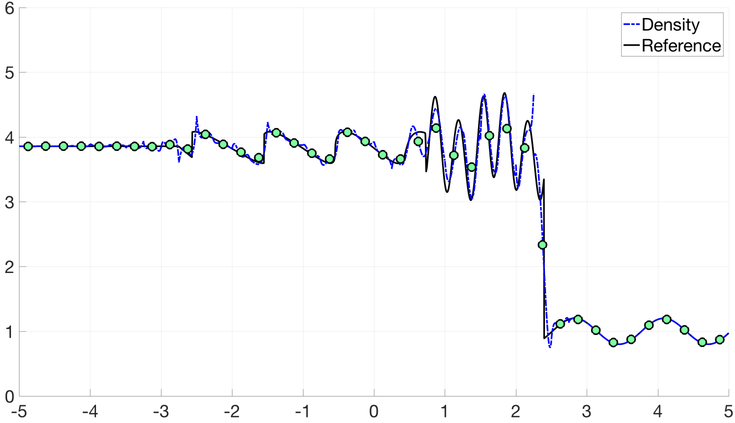
- GQ-$(N+2)$ needs smaller CFL (.05 vs .125) for stability.



$N = 4$, $K = 40$, $CFL = .05$, $(N+1)$ point Gauss-Lobatto-Legendre quadrature.

1D sine-shock interaction

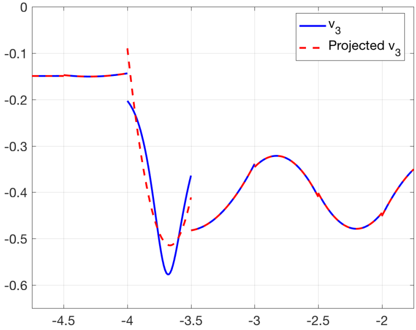
- GQ-$(N+2)$ needs smaller CFL (.05 vs .125) for stability.



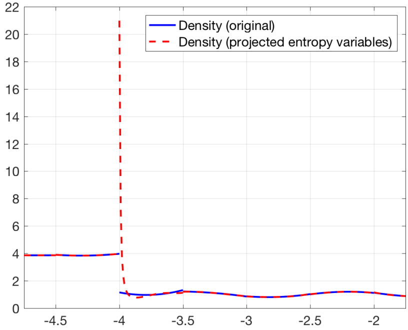
$N = 4, K = 40, CFL = .05$, $(N+2)$ point Gauss quadrature.

On CFL restrictions

- For GLL-$(N+1)$ quadrature, $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(P_N \boldsymbol{v}\right) = \boldsymbol{u}$ at GLL points.
- For GQ-$(N+2)$, discrepancy between L^2 projection and interpolation.
- Still need positivity of thermodynamic quantities for stability!



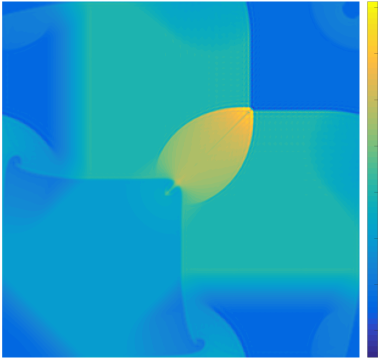
(a) $v_3(x), \left(P_N v_3\right)(x)$



(b) $\rho(x), \rho\left(\left(P_N \boldsymbol{v}\right)(x)\right)$

Talk outline

2D Riemann problem

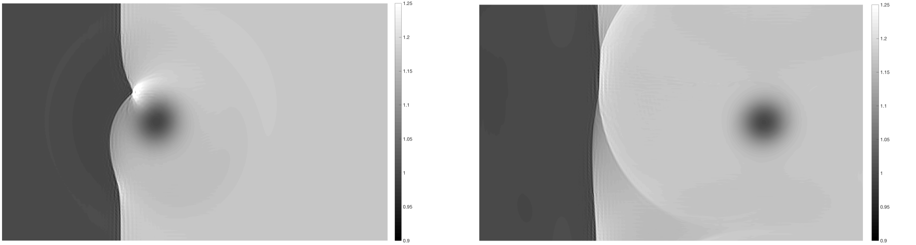


(a) $\Omega = [-1, 1]^2$

(b) $\Omega = [-.5, .5]^2$, 32×32 elements

- Degree N polynomials, degree $2N$ volume and surface quadratures.
- Uniform 64×64 triangle mesh: $N = 3$, CFL .125, Lax-Friedrichs flux.
- Periodic on larger domain ("natural" boundary conditions unstable).

2D shock-vortex interaction

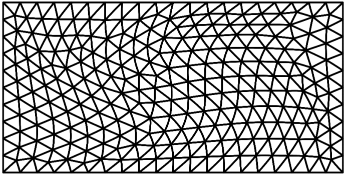


(a) $t = .3$ (b) $t = .7$

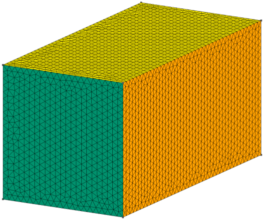
- Vortex passing through a shock on a periodic domain (matrix dissipation, degree $N = 3$ approximation, mesh size $h = 1/128$).
- Can also impose existing entropy stable wall boundary conditions for compressible Euler with decoupled SBP.

Winters, Derigs, Gassner, Walch (2017). *A uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations.*

Smooth isentropic vortex and curved meshes in 2D/3D



(a) 2D triangular mesh



(b) 3D tetrahedral mesh

Figure: Example of 2D and 3D meshes used for convergence experiments.

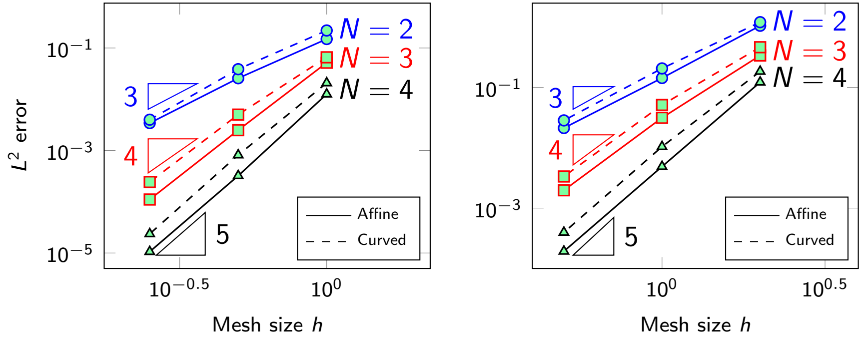
- Entropy stability: needs discrete geometric conservation law (GCL).
- Generalized mass lumping for curved: weight-adjusted mass matrices.
- Modify $\widetilde{\boldsymbol{u}} = \boldsymbol{u}(\widetilde{\boldsymbol{v}})$, $\widetilde{\boldsymbol{v}} = \widetilde{P}_N^k \boldsymbol{v}(\boldsymbol{u}_h)$ using weight-adjusted projection $\widetilde{P}_N^k$.

Visbal and Gaitonde (2002). On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes.

Kopriva (2006). Metric identities and the discontinuous spectral element method on curvilinear meshes.

Chan, Hewett, and Warburton (2016). *Weight-adjusted discontinuous Galerkin methods: curvilinear meshes.*

Smooth isentropic vortex and curved meshes in 2D/3D

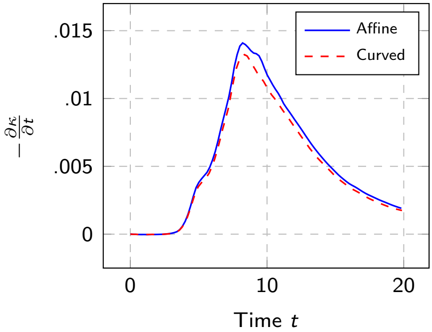


(a) 2D results (b) 3D results

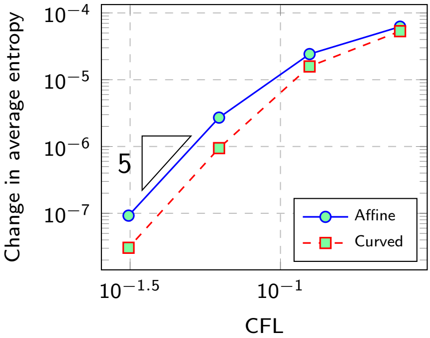
L^2 errors for 2D/3D isentropic vortex at $T = 5$ on affine, curved meshes.

Visbal and Gaitonde (2002). On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes.
Kopriva (2006). Metric identities and the discontinuous spectral element method on curvilinear meshes.
Chan, Hewett, and Warburton (2016). *Weight-adjusted discontinuous Galerkin methods: curvilinear meshes.*

3D inviscid Taylor-Green vortex: KE dissipation rate



(a) KE dissipation rate ($N = 3$, $h = \pi/8$)

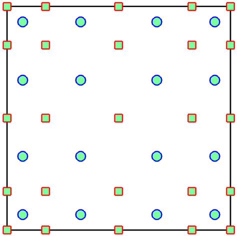


(b) Change in $\int_\Omega U(\boldsymbol{u})$ (EC scheme)

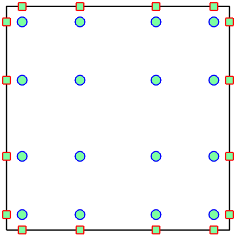
- Kinetic energy dissipation rate: good agreement with literature.
- Change in $\int_\Omega U(\boldsymbol{u}) \to 0$ as $\mathrm{CFL} \to 0$ for entropy conservative scheme.

Talk outline

ES Gauss collocation (w/M. Carpenter, DCDR Fernandez)



(a) Staggered-grid

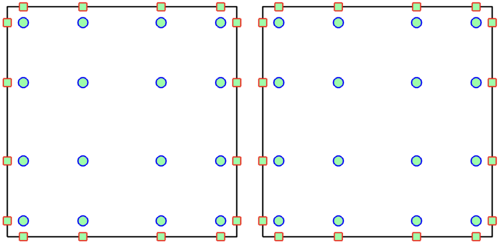


(b) Generalized SBP

- Gauss vs GLL quadrature: exact for degree $(2N+1)$ vs $(2N-1)$.
- Inter-element coupling for Gauss is expensive. Staggered grid collocation is an alternative, but requires degree $(N+1)$ GLL nodes.
- ES Gauss scheme from decoupled SBP (collocation: $\boldsymbol{V}_q = \boldsymbol{P}_q = \boldsymbol{I}$).

Parsani et al. (2016). *ES staggered grid disc. spectral collocation methods of any order for the comp. NS eqns.*

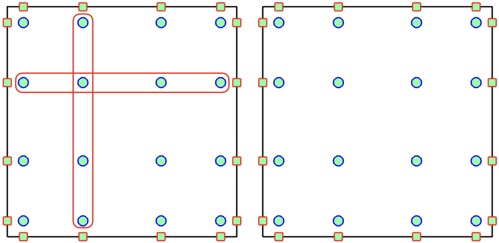
Entropy stable Gauss collocation: main steps



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix} \right) \mathbf{1}$$

- Collocate $\boldsymbol{u}$, interpolate entropy variables $\boldsymbol{v}(\boldsymbol{u})$ to surface nodes.
- Perform flux differencing at Gauss nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

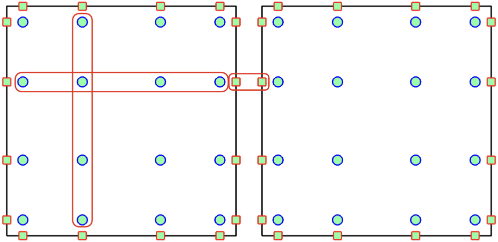
Entropy stable Gauss collocation: main steps



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix} \right) \mathbf{1}$$

- Collocate $\boldsymbol{u}$, interpolate entropy variables $\boldsymbol{v}(\boldsymbol{u})$ to surface nodes.
- Perform flux differencing at Gauss nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

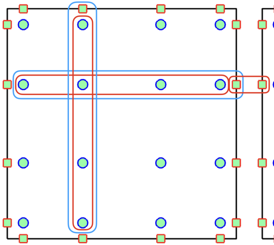
Entropy stable Gauss collocation: main steps



$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff} \end{bmatrix}\right)\mathbf{1}$$

- Collocate $\boldsymbol{u}$, interpolate entropy variables $\boldsymbol{v}(\boldsymbol{u})$ to surface nodes.
- Perform flux differencing at Gauss nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

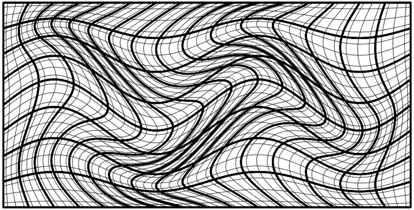
Entropy stable Gauss collocation: main steps



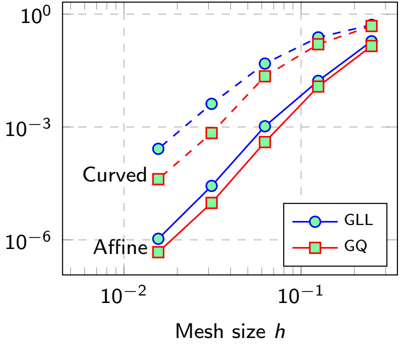
$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^T & \boldsymbol{C}\end{bmatrix}}_{\boldsymbol{Q}_N^i} \circ \begin{bmatrix} \boldsymbol{F}_S^{vv} & \boldsymbol{F}_S^{vf} \\ \boldsymbol{F}_S^{fv} & \boldsymbol{F}_S^{ff}\end{bmatrix}\right)\mathbf{1}$$

- Collocate $\boldsymbol{u}$, interpolate entropy variables $\boldsymbol{v}(\boldsymbol{u})$ to surface nodes.
- Perform flux differencing at Gauss nodes.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ for surface nodes of neighboring elements.
- Compute $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ between volume/surface nodes, apply flux differencing with interp. matrix + transpose for volume/surface nodes.

Numerical results: 2D/3D isentropic vortex



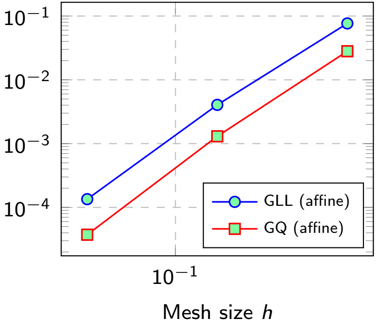
(a) Warped curvilinear mesh



(b) 2D L^2 errors ($N = 4$)

Entropy stability for Gauss collocation on curved meshes: compute geometric terms at GLL points, interpolate to volume and face points.

Numerical results: 2D/3D isentropic vortex



(a) 3D L^2 errors ($N = 4$)

Curvilinear results: in progress!

Summary and future work

- Discrete semi-discrete entropy stability for (almost) arbitrary choices of basis, quadrature. Usual challenges (positivity, Gibbs, BCs) apply.
- DG-SEM: volume/surface cross terms cancel out!
- Current work: Gauss collocation (with DCDR Fernandez, M. Carpenter), adaptivity + hybrid meshes, multi-GPU.
- This work is supported by DMS-1719818 and DMS-1712639.

Thank you! Questions?

Chan, Wilcox (2018). *On discretely entropy stable weight-adjusted DG methods: curvilinear meshes.*

Chan (2017). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods.*

Additional slides

Sketch of proof of entropy conservation (one element)

- Multiply by mass matrix on both sides, rewrite as

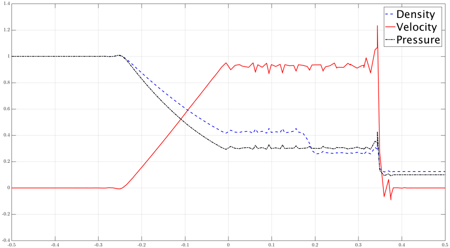
$$\boldsymbol{M}\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} + \begin{bmatrix}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{bmatrix}^T \left(\boldsymbol{Q}_N \circ \boldsymbol{f}_S\left(\begin{bmatrix}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{bmatrix}\boldsymbol{P}_q\boldsymbol{v}_q\right)\right)\mathbf{1} = 0.$$

- Test with L^2 projection of entropy variables $\boldsymbol{P}_q\boldsymbol{v}_q = \boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}\boldsymbol{v}_q$.

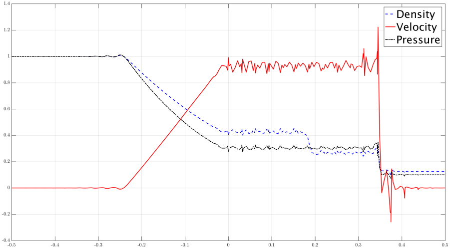
$$\begin{aligned}(\boldsymbol{P}_q\boldsymbol{v}_q)^T\boldsymbol{M}\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t} &= \boldsymbol{v}_q^T\boldsymbol{W}\boldsymbol{V}_q\boldsymbol{M}^{-1}\boldsymbol{M}\boldsymbol{V}_q\frac{\mathrm{d}\widehat{\boldsymbol{u}}}{\mathrm{d}t}\\ &= \boldsymbol{v}_q^T\boldsymbol{W}\frac{\mathrm{d}(\boldsymbol{V}_q\widehat{\boldsymbol{u}})}{\mathrm{d}t} = \mathbf{1}^T\boldsymbol{W}\left(\frac{\mathrm{d}S(\boldsymbol{u}_q)}{\mathrm{d}\boldsymbol{u}}\frac{\mathrm{d}\boldsymbol{u}_q}{\mathrm{d}t}\right) = \frac{\mathrm{d}S(\boldsymbol{u}_q)}{\mathrm{d}t}.\end{aligned}$$

- Spatial term vanishes using SBP, skew-symmetry, and properties of $\boldsymbol{f}_S$.

1D Sod: over-integration ineffective w/out L^2 projection



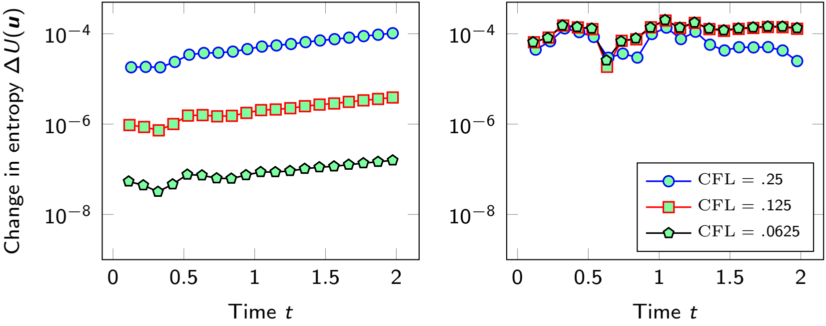
(a) Degree N GLL, $(N+1)$ points



(b) Degree N GLL, $(N+4)$ points

Figure: Sod shock tube for $N = 4$ and $K = 32$ elements. Over-integrating by increasing the number of quadrature points does not improve solution quality.

2D curved meshes: conservation of entropy



(a) With weight-adjusted projection (b) Without weight-adjusted projection

Figure: Change in entropy under an entropy conservative flux with $N = 4$. In both cases, the spatial formulation tested with $\widetilde{\boldsymbol{v}} = P_N \boldsymbol{v}(\boldsymbol{u})$ is $O\left(10^{-14}\right)$.