

# Discretely entropy stable discontinuous Galerkin methods for nonlinear conservation laws

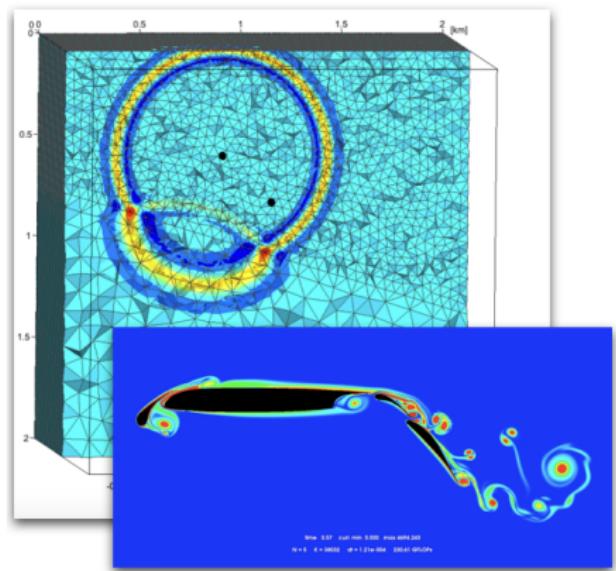
Jesse Chan

<sup>1</sup>Department of Computational and Applied Math

WCCM 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).

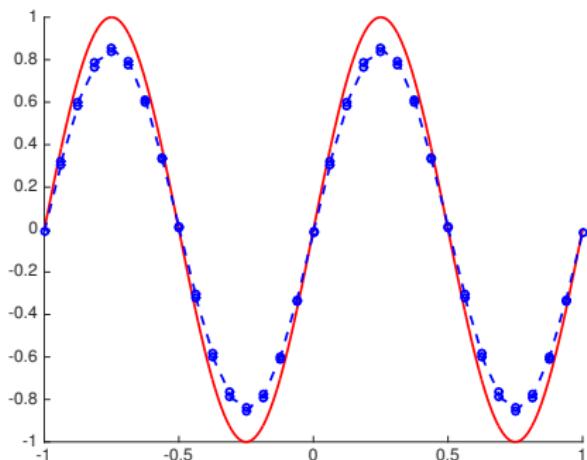


Goal: address instability of high order methods.

Figures courtesy of T. Warburton, A. Modave.

# 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).

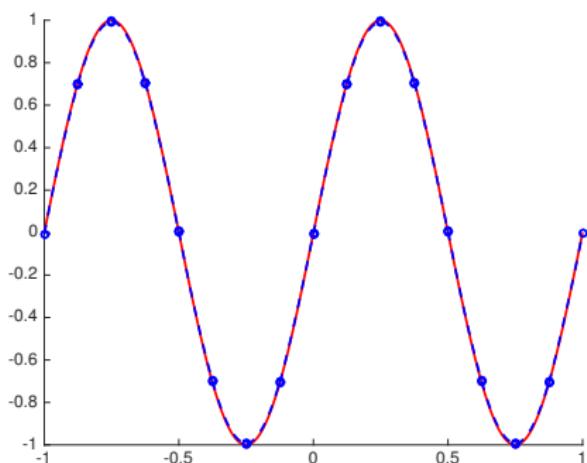


**Fine** linear approximation.

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).

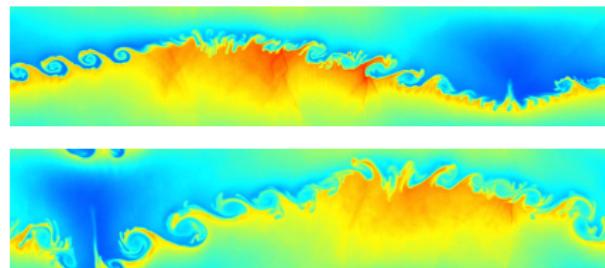


Figure from Per-Olof Persson.

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).

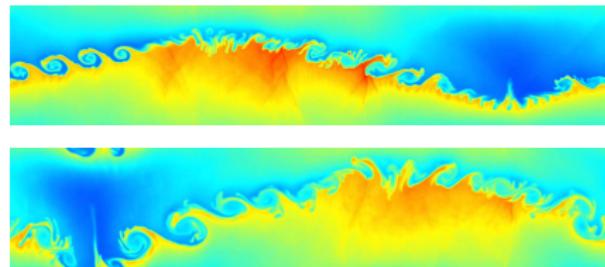
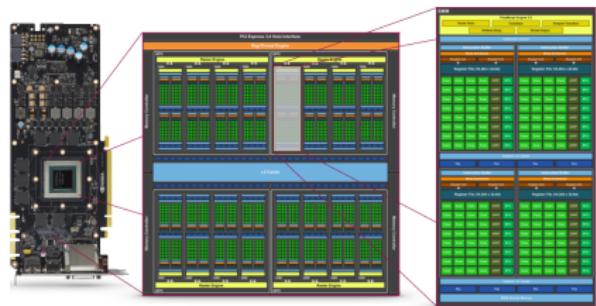


Figure from Per-Olof Persson.

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).

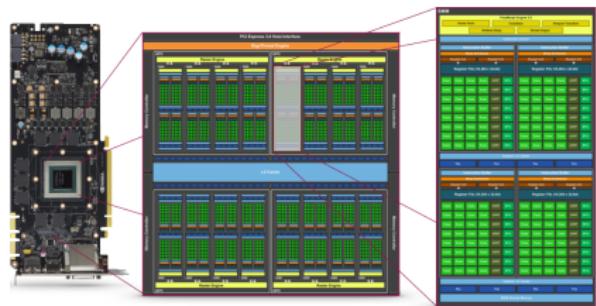


A graphics processing unit (GPU).

Goal: address **instability** of high order methods!

# High order methods for time-dependent hyperbolic PDEs

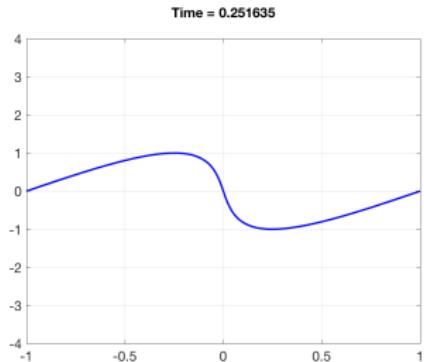
- Accurate resolution of propagating waves and vortices.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Many-core architectures (efficient explicit time-stepping).



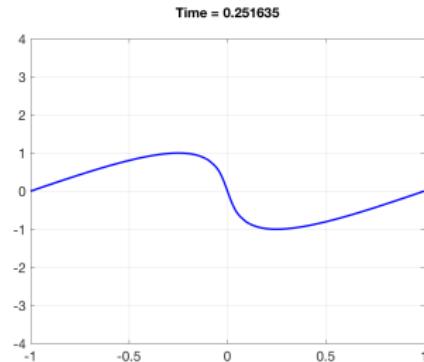
A graphics processing unit (GPU).

Goal: address **instability** of high order methods!

# Why are discretizations of nonlinear PDEs so unstable?



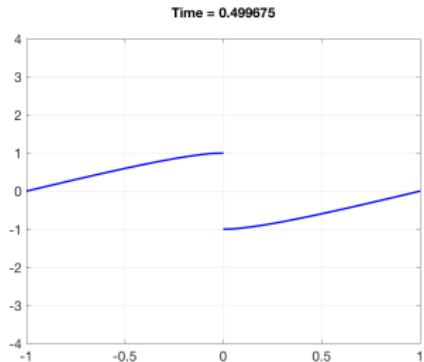
(a)  $N = 7, K = 8$  (aligned mesh)



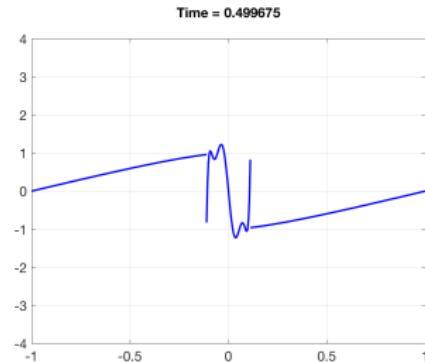
(b)  $N = 7, K = 9$  (non-aligned mesh)

- Burgers' equation:  $f(u) = u^2/2$ . How to compute  $\frac{\partial}{\partial x} f(u)$ ?  
$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0, \quad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$
- Differentiating  $L^2$  projection  $P_N$  + inexact quadrature: **no chain rule**.  
$$\int_{D^k} \left( \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial}{\partial x} P_N u^2 \right) v \, dx = 0, \quad \frac{1}{2} \frac{\partial P_N u^2}{\partial x} \neq P_N \left( u \frac{\partial u}{\partial x} \right)$$

# Why are discretizations of nonlinear PDEs so unstable?



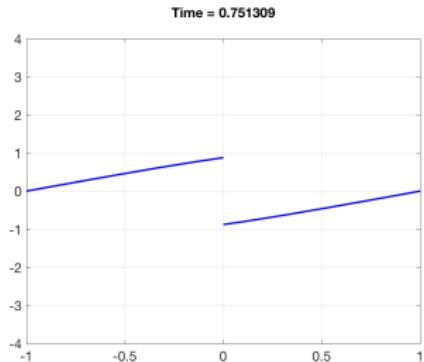
(a)  $N = 7, K = 8$  (aligned mesh)



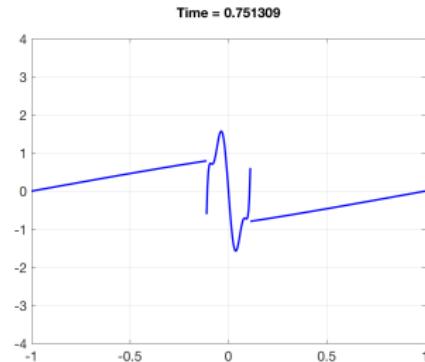
(b)  $N = 7, K = 9$  (non-aligned mesh)

- Burgers' equation:  $f(u) = u^2/2$ . How to compute  $\frac{\partial}{\partial x} f(u)$ ?
$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0, \quad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$
- Differentiating  $L^2$  projection  $P_N$  + inexact quadrature: **no chain rule**.
$$\int_{D^k} \left( \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial}{\partial x} P_N u^2 \right) v \, dx = 0, \quad \frac{1}{2} \frac{\partial P_N u^2}{\partial x} \neq P_N \left( u \frac{\partial u}{\partial x} \right)$$

# Why are discretizations of nonlinear PDEs so unstable?



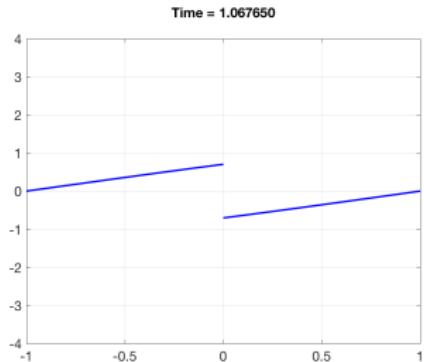
(a)  $N = 7, K = 8$  (aligned mesh)



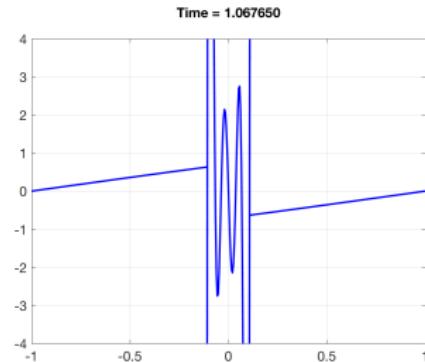
(b)  $N = 7, K = 9$  (non-aligned mesh)

- Burgers' equation:  $f(u) = u^2/2$ . How to compute  $\frac{\partial}{\partial x} f(u)$ ?  
$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0, \quad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$
- Differentiating  $L^2$  projection  $P_N$  + inexact quadrature: **no chain rule**.  
$$\int_{D^k} \left( \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial}{\partial x} P_N u^2 \right) v \, dx = 0, \quad \frac{1}{2} \frac{\partial P_N u^2}{\partial x} \neq P_N \left( u \frac{\partial u}{\partial x} \right)$$

# Why are discretizations of nonlinear PDEs so 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  $\frac{\partial}{\partial x} f(u)$ ?
$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0, \quad u \in P^N(D^k), \quad u^2 \notin P^N(D^k).$$
- Differentiating  $L^2$  projection  $P_N$  + inexact quadrature: **no chain rule**.
$$\int_{D^k} \left( \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial}{\partial x} P_N u^2 \right) v \, dx = 0, \quad \frac{1}{2} \frac{\partial P_N u^2}{\partial x} \neq P_N \left( u \frac{\partial u}{\partial x} \right)$$

# Entropy stability for nonlinear conservation laws

- Analogue of energy for nonlinear systems of conservation laws

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

- Weak solutions satisfy a continuous entropy inequality: convex **entropy** function  $S(\mathbf{u})$  and “entropy potential”  $\psi(\mathbf{u})$ .

$$\begin{aligned} & \int_{\Omega} \mathbf{v}^T \left( \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} \right) = 0, \quad \mathbf{v} = \frac{\partial S}{\partial \mathbf{u}} \\ & \Rightarrow \int_{\Omega} \frac{\partial S(\mathbf{u})}{\partial t} + \left( \mathbf{v}^T \mathbf{f}(\mathbf{u}) - \psi(\mathbf{u}) \right) \Big|_{-1}^1 \leq 0. \end{aligned}$$

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

## Examples of entropy functions

- Burgers': square entropy  $S(\mathbf{u}) = u^2/2$ , entropy variables  $\mathbf{v}(\mathbf{u}) = u$ .
- Shallow water: entropy is total energy, convex if  $h > 0$ .

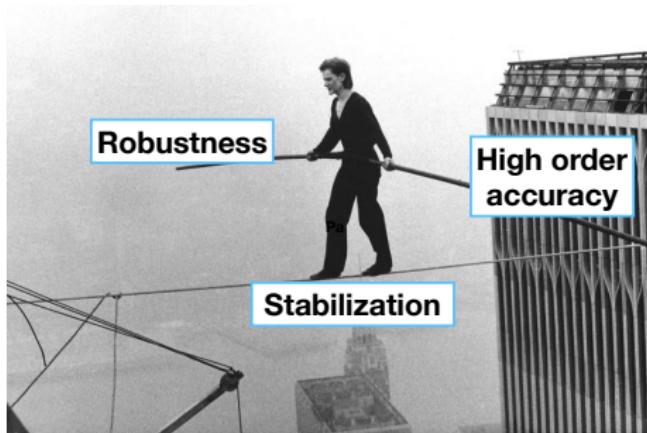
$$S(\mathbf{u}) = \frac{1}{2}hu^2 + \frac{1}{2}gh^2 + ghb, \quad \mathbf{v}(\mathbf{u}) = \left[ \begin{matrix} g(h+b) - \frac{u^2}{2} \\ u \end{matrix} \right].$$

- Compressible Euler equations: physical entropy  $s(\mathbf{u})$  always increases; mathematical entropy  $S(\mathbf{u})$  always decreases.

$$s(\mathbf{u}) = \log \left( \frac{(\gamma - 1)\rho e}{\rho^\gamma} \right), \quad S(\mathbf{u}) = -\rho s(\mathbf{u})$$

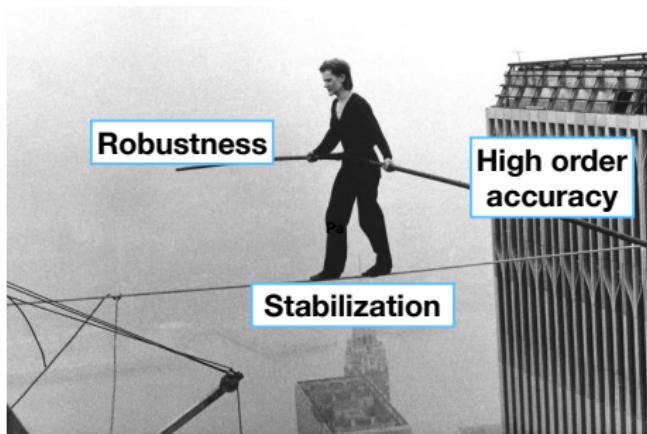
$$\mathbf{v}(\mathbf{u}) = \frac{\partial S}{\partial \mathbf{u}} = \frac{1}{\rho e} \begin{pmatrix} \rho e(\gamma + 1 - s(\mathbf{u})) - E \\ m \\ -\rho \end{pmatrix}$$

# 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 DG as summation by parts finite differences
- 2 Entropy stable formulations through flux differencing
- 3 Numerical experiments: triangular and tetrahedral meshes

# Talk outline

- 1 DG as summation by parts finite differences
- 2 Entropy stable formulations through flux differencing
- 3 Numerical experiments: triangular and tetrahedral meshes

# Entropy stable schemes in a nutshell



$$\mathbf{D}_{ij} = \left. \frac{\partial \ell_j}{\partial x} \right|_{x=x_i}$$

- Collocation scheme using Gauss-Legendre-Lobatto quadrature points

$$\frac{d\mathbf{u}}{dt} + \mathbf{D}\mathbf{f}(\mathbf{u}) = 0 \quad \Rightarrow \quad \frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} \mathbf{f}(\mathbf{u}_i) = 0.$$

- Flux differencing: recover above form if  $f_S(u_i, u_j) = \frac{1}{2}(u_i + u_j)$ .

$$\frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} 2f_S(u_i, u_j) = 0 \quad \Rightarrow \quad \frac{d\mathbf{u}}{dt} + 2(\mathbf{D} \circ \mathbf{F}_S)\mathbf{1} = 0.$$

- Semi-discrete entropy equality (add dissipation for inequality)

$$\mathbf{M} \frac{dS(\mathbf{u})}{dt} + \mathbf{1}^T \mathbf{B} \left( \mathbf{v}^T \mathbf{f}(\mathbf{u}) - \psi(\mathbf{u}) \right) = 0.$$

# Entropy stable schemes in a nutshell



$$\mathbf{D}_{ij} = \left. \frac{\partial \ell_j}{\partial x} \right|_{x=x_i}$$

- Collocation scheme using Gauss-Legendre-Lobatto quadrature points

$$\frac{d\mathbf{u}}{dt} + \mathbf{D}\mathbf{f}(\mathbf{u}) = 0 \implies \frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} \mathbf{f}(\mathbf{u}_i) = 0.$$

- Flux differencing: recover above form if  $\mathbf{f}_S(\mathbf{u}_i, \mathbf{u}_j) = \frac{1}{2}(\mathbf{u}_i + \mathbf{u}_j)$ .

$$\frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} 2\mathbf{f}_S(\mathbf{u}_i, \mathbf{u}_j) = 0 \implies \frac{d\mathbf{u}}{dt} + 2(\mathbf{D} \circ \mathbf{F}_S)\mathbf{1} = 0.$$

- Semi-discrete entropy equality (add dissipation for inequality)

$$\mathbf{M} \frac{dS(\mathbf{u})}{dt} + \mathbf{1}^T \mathbf{B} \left( \mathbf{v}^T \mathbf{f}(\mathbf{u}) - \psi(\mathbf{u}) \right) = 0.$$

# Entropy stable schemes in a nutshell



$$\mathbf{D}_{ij} = \left. \frac{\partial \ell_j}{\partial x} \right|_{x=x_i}$$

- Collocation scheme using Gauss-Legendre-Lobatto quadrature points

$$\frac{d\mathbf{u}}{dt} + \mathbf{D}\mathbf{f}(\mathbf{u}) = 0 \implies \frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} \mathbf{f}(\mathbf{u}_i) = 0.$$

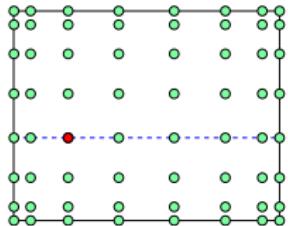
- Flux differencing: recover above form if  $\mathbf{f}_S(\mathbf{u}_i, \mathbf{u}_j) = \frac{1}{2} (\mathbf{u}_i + \mathbf{u}_j)$ .

$$\frac{d\mathbf{u}_i}{dt} + \sum_j \mathbf{D}_{ij} 2\mathbf{f}_S(\mathbf{u}_i, \mathbf{u}_j) = 0 \implies \frac{d\mathbf{u}}{dt} + 2(\mathbf{D} \circ \mathbf{F}_S)\mathbf{1} = 0.$$

- Semi-discrete entropy equality (add dissipation for inequality)

$$\mathbf{M} \frac{dS(\mathbf{u})}{dt} + \mathbf{1}^T \mathbf{B} \left( \mathbf{v}^T \mathbf{f}(\mathbf{u}) - \psi(\mathbf{u}) \right) = 0.$$

# Brief overview of entropy stable high order schemes

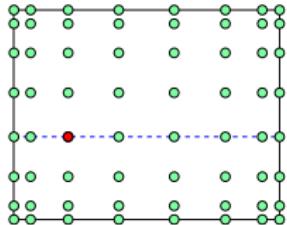


(a) GLL collocation

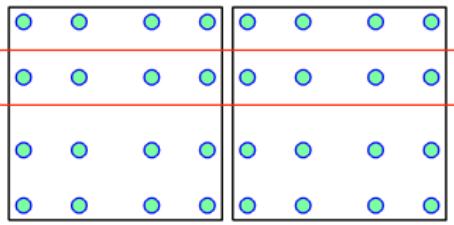
- SBP finite difference schemes: **semi-discrete entropy inequality**.  
Includes DG-SEM (Gauss-Legendre-Lobatto collocation) methods.
- Generalized SBP (e.g. Gauss collocation): higher accuracy, but requires **non-compact coupling conditions** b/w neighboring elements.
- Tetrahedra, wedges, pyramids: how to address over-integration?

My focus: **entropy stability, compact couplings, arbitrary quadratures**.

# Brief overview of entropy stable high order schemes



(a) GLL collocation

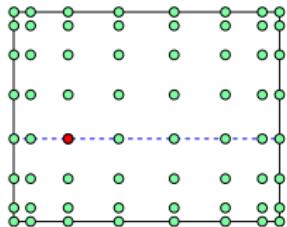


(b) Gauss nodes coupling

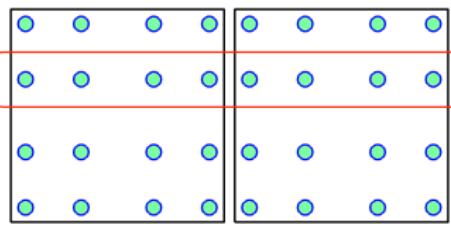
- SBP finite difference schemes: **semi-discrete entropy inequality**.  
Includes DG-SEM (Gauss-Legendre-Lobatto collocation) methods.
- Generalized SBP (e.g. Gauss collocation): higher accuracy, but requires **non-compact coupling conditions** b/w neighboring elements.
- Tetrahedra, wedges, pyramids: how to address over-integration?

My focus: **entropy stability, compact couplings, arbitrary quadratures**.

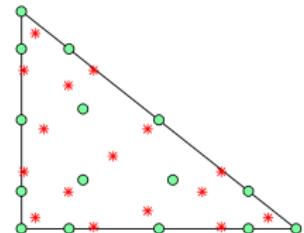
# Brief overview of entropy stable high order schemes



(a) GLL collocation



(b) Gauss nodes coupling

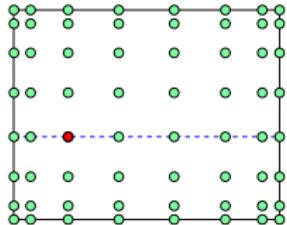


(c) Nodes vs cubature

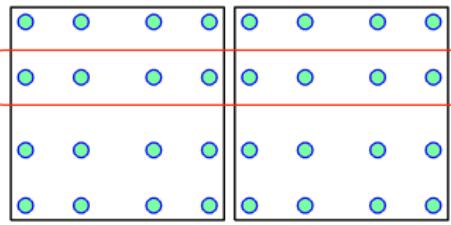
- SBP finite difference schemes: **semi-discrete entropy inequality**.  
Includes DG-SEM (Gauss-Legendre-Lobatto collocation) methods.
- Generalized SBP (e.g. Gauss collocation): higher accuracy, but requires **non-compact coupling conditions** b/w neighboring elements.
- Tetrahedra, wedges, pyramids: how to address over-integration?

My focus: **entropy stability, compact couplings, arbitrary quadratures**.

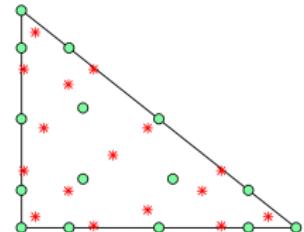
# Brief overview of entropy stable high order schemes



(a) GLL collocation



(b) Gauss nodes coupling



(c) Nodes vs cubature

- SBP finite difference schemes: **semi-discrete entropy inequality**.  
Includes DG-SEM (Gauss-Legendre-Lobatto collocation) methods.
- Generalized SBP (e.g. Gauss collocation): higher accuracy, but requires **non-compact coupling conditions** b/w neighboring elements.
- Tetrahedra, wedges, pyramids: how to address over-integration?

My focus: **entropy stability, compact couplings, arbitrary quadratures**.

# Quadrature-based matrices for polynomial bases

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

$$\mathbf{W} = \text{diag}(\mathbf{w}^q), \quad \mathbf{W}_f = \text{diag}(\mathbf{w}^f).$$

- Assume some polynomial basis  $\phi_1, \dots, \phi_{N_p}$ . Define “modal” differentiation matrix  $\mathbf{D}^i$ , the interpolation matrices  $\mathbf{V}_q, \mathbf{V}_f$

$$(\mathbf{V}_q)_{ij} = \phi_j(\mathbf{x}_i^q), \quad (\mathbf{V}_f)_{ij} = \phi_j(\mathbf{x}_i^f).$$

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

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

- These matrices map to and from **polynomial** and **quadrature** spaces.

# Quadrature-based differentiation matrices

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

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

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

$$\mathbf{W} \mathbf{D}_q^i + (\mathbf{W} \mathbf{D}_q^i)^T = (\mathbf{V}_f \mathbf{P}_q)^T \mathbf{W}_f \text{diag}(\mathbf{n}_i) \mathbf{V}_f \mathbf{P}_q.$$

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

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

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

# Making DG look like FD: “decoupled” SBP operator

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

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

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

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

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

# Differentiation using decoupled SBP operators

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

$$f \frac{\partial g}{\partial x} \approx [ \mathbf{P}_q \quad \mathbf{L}_f ] \operatorname{diag}(\mathbf{f}) \mathbf{D}_N \mathbf{g}, \quad \mathbf{f}_i, \mathbf{g}_i = f(\mathbf{x}_i), g(\mathbf{x}_i).$$

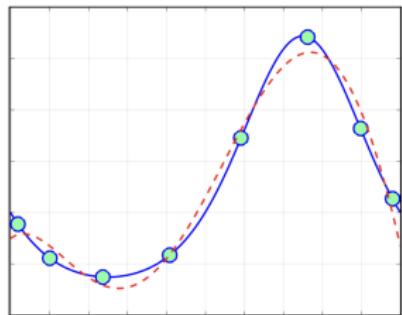
- Equivalent to solving variational problem for  $u(\mathbf{x}) \approx f \frac{\partial g}{\partial x}$

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

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

# Accuracy of $D_N^i$

$f(x)$  and  $L^2$  projection

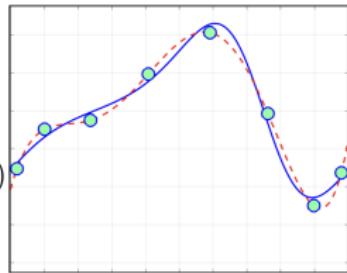
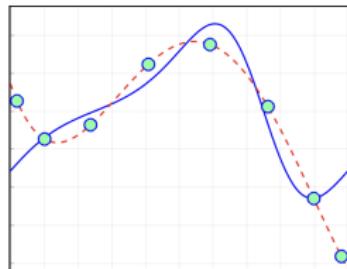


$D_q^i$

$D_N^i$

(boundary correction)

$\frac{\partial f}{\partial x}$  and approx. derivative

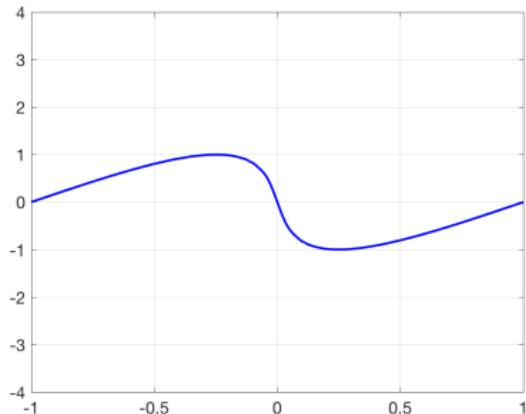


# Talk outline

- 1 DG as summation by parts finite differences
- 2 Entropy stable formulations through flux differencing
- 3 Numerical experiments: triangular and tetrahedral meshes

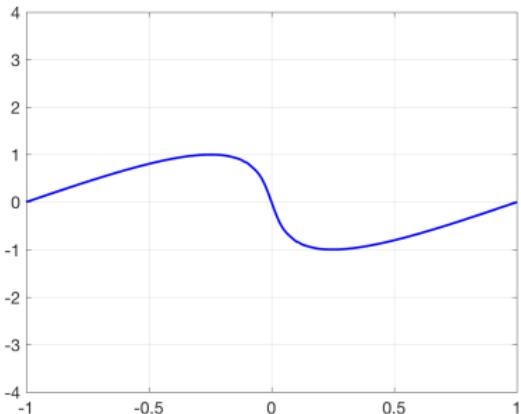
$$\text{Split form of Burgers': } \frac{\partial u}{\partial t} + \frac{1}{3} \left( \frac{\partial u^2}{\partial x} + u \frac{\partial u}{\partial x} \right) = 0$$

Time = 0.251799



(a) Energy conservative

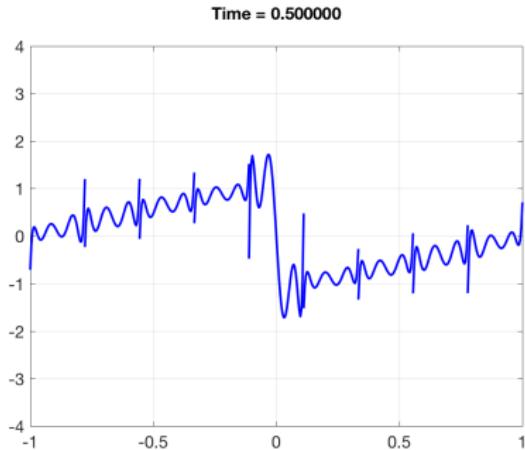
Time = 0.251799



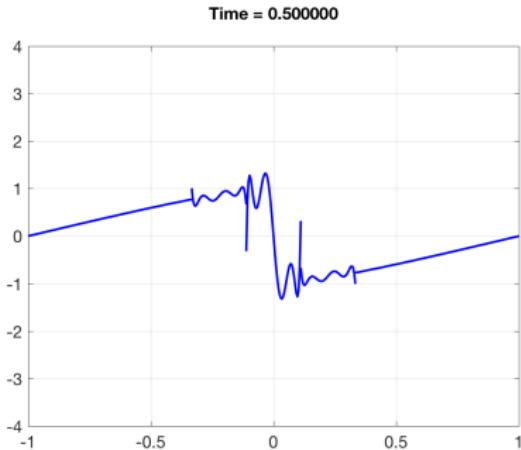
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



(a) Energy conservative

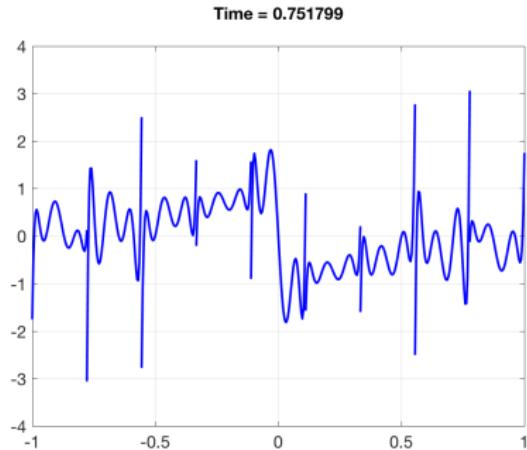


(b) Energy stable

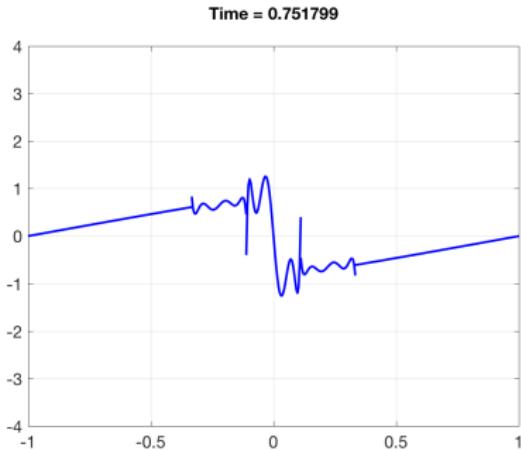
$$\mathbf{u} = \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux}$$

$$\frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) = 0.$$

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



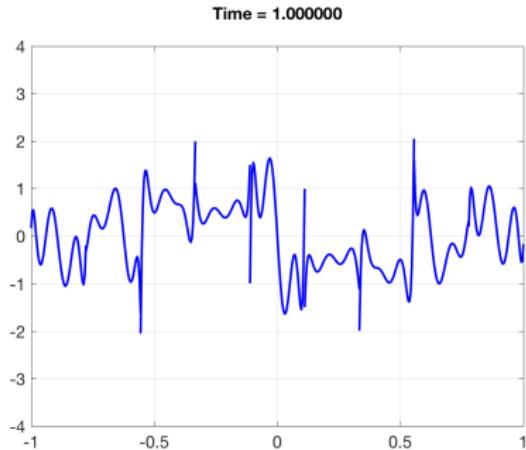
(a) Energy conservative



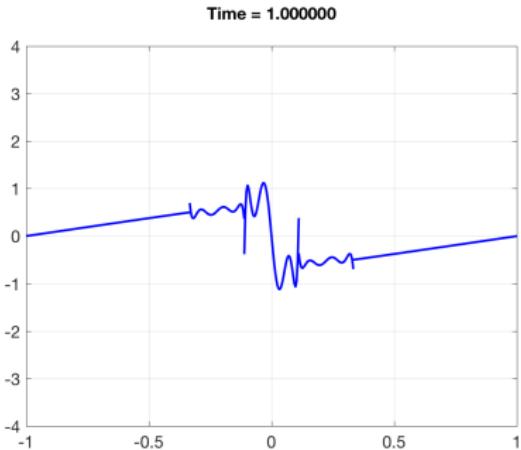
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



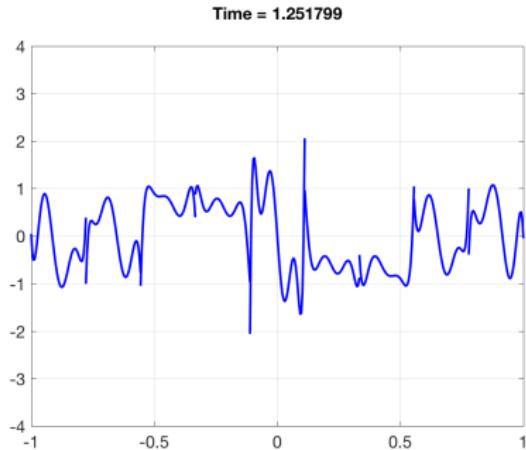
(a) Energy conservative



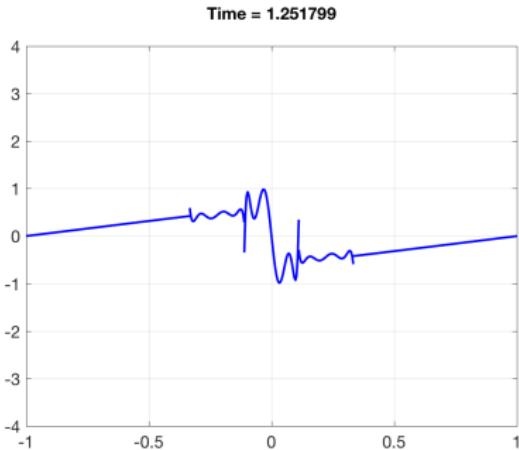
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



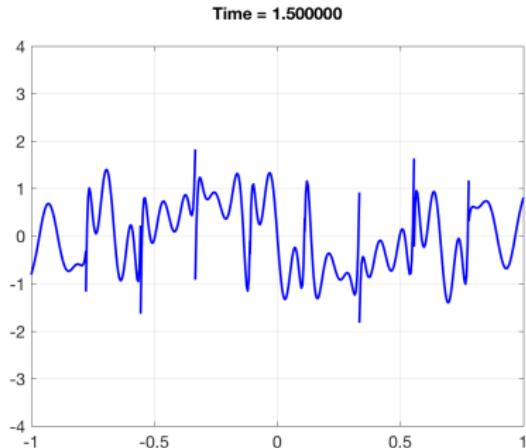
(a) Energy conservative



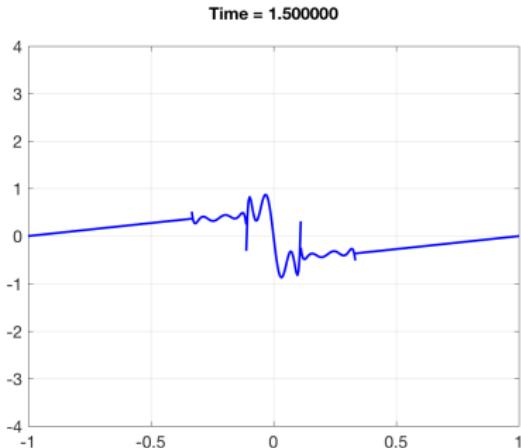
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



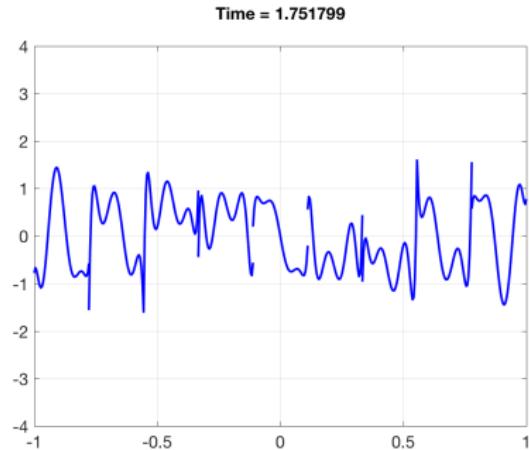
(a) Energy conservative



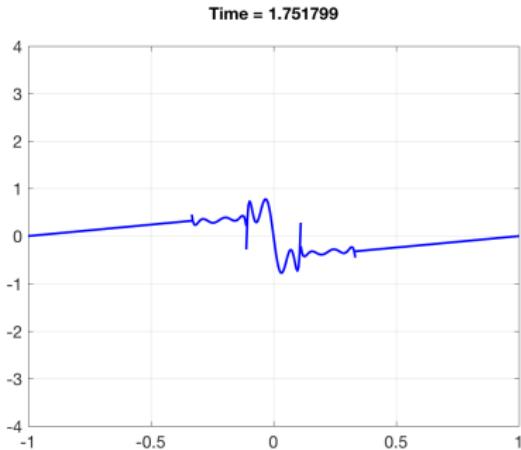
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



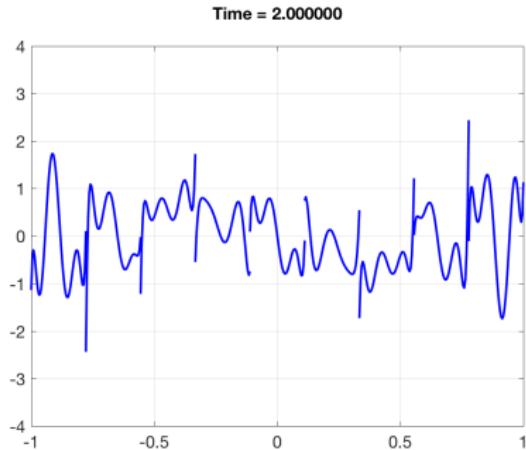
(a) Energy conservative



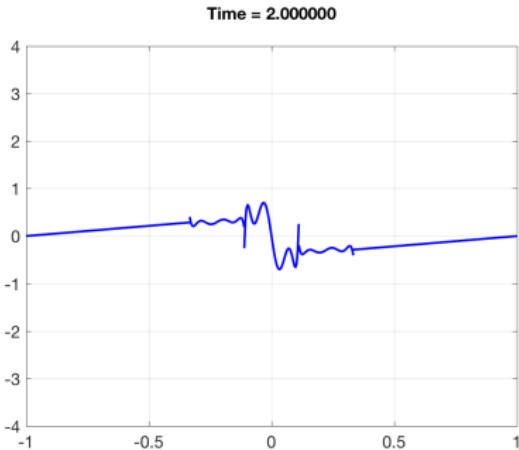
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



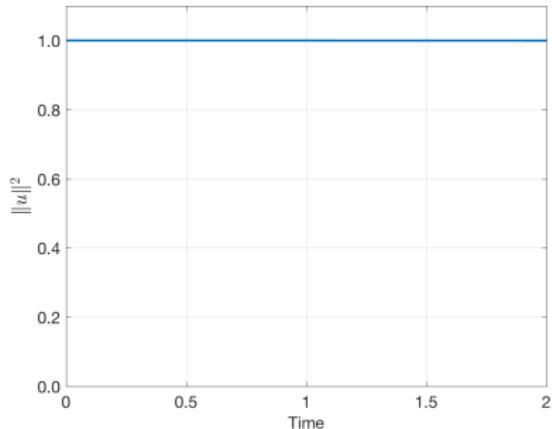
(a) Energy conservative



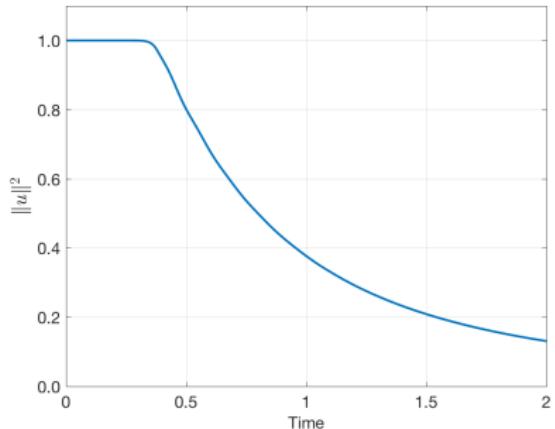
(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(u^+, u) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] (\mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u}) + \mathbf{L}_f (\mathbf{f}^*(u^+, u)) &= 0. \end{aligned}$$

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



(a) Energy conservative



(b) Energy stable

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \hat{\mathbf{u}}, \quad \hat{\mathbf{u}} = \text{coefficients}, \quad \mathbf{f}^*(\mathbf{u}^+, \mathbf{u}) = \text{numerical flux} \\ \frac{d\hat{\mathbf{u}}}{dt} + \frac{1}{3} [ \mathbf{P}_q \quad \mathbf{L}_f ] ( \mathbf{D}_N (\mathbf{u}^2) + \text{diag}(\mathbf{u}) \mathbf{D}_N \mathbf{u} ) + \mathbf{L}_f (\mathbf{f}^*(\mathbf{u}^+, \mathbf{u})) &= 0. \end{aligned}$$

# Flux differencing: entropy conservative finite volume fluxes

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

$$\mathbf{f}_S(\mathbf{u}, \mathbf{u}) = \mathbf{f}(\mathbf{u}), \quad \mathbf{f}_S(\mathbf{u}, \mathbf{v}) = \mathbf{f}_S(\mathbf{v}, \mathbf{u}), \quad (\text{consistency, symmetry})$$

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

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

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

- Extends beyond split formulations: mass flux for compressible Euler

$$f_S^\rho(u_L, u_R) = \{\{\rho\}\}^{\log} \{\{u\}\}, \quad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log \rho_L - \log \rho_R}.$$

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

$$\begin{aligned} \mathbf{f}_S(\mathbf{u}, \mathbf{u}) &= \mathbf{f}(\mathbf{u}), & \mathbf{f}_S(\mathbf{u}, \mathbf{v}) &= \mathbf{f}_S(\mathbf{v}, \mathbf{u}), & (\text{consistency, symmetry}) \\ (\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}(\mathbf{u}_L, \mathbf{u}_R) &= \psi_L - \psi_R, & (\text{conservation}) \end{aligned}$$

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

$$\begin{aligned} f_S(u_L, u_R) &= \frac{1}{6} (u_L^2 + u_L u_R + u_R^2), \\ \frac{\partial f(u)}{\partial x} \implies 2 \frac{\partial f_S(u(x), u(y))}{\partial x} \Big|_{y=x} &= \frac{1}{3} \frac{\partial u^2}{\partial x} + \frac{1}{3} u \frac{\partial u}{\partial x} + \frac{1}{3} u^2 \frac{\partial u}{\partial x} \end{aligned}$$

- Extends beyond split formulations: mass flux for compressible Euler

$$f_S^\rho(u_L, u_R) = \{\{\rho\}\}^{\log} \{\{u\}\}, \quad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log \rho_L - \log \rho_R}.$$

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

$$\begin{aligned} \mathbf{f}_S(\mathbf{u}, \mathbf{u}) &= \mathbf{f}(\mathbf{u}), & \mathbf{f}_S(\mathbf{u}, \mathbf{v}) &= \mathbf{f}_S(\mathbf{v}, \mathbf{u}), & (\text{consistency, symmetry}) \\ (\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}(\mathbf{u}_L, \mathbf{u}_R) &= \psi_L - \psi_R, & (\text{conservation}) \end{aligned}$$

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

$$\begin{aligned} f_S(u(x), u(y)) &= \frac{1}{6} (u(x)^2 + u(x)u(y) + u(y)^2), \\ \frac{\partial f(u)}{\partial x} \implies 2 \frac{\partial f_S(u(x), u(y))}{\partial x} \Big|_{y=x} &= \frac{1}{3} \frac{\partial u^2}{\partial x} + \frac{1}{3} u \frac{\partial u}{\partial x} + \frac{1}{3} u^2 \frac{\partial u}{\partial x} \end{aligned}$$

- Extends beyond split formulations: mass flux for compressible Euler

$$f_S^\rho(u_L, u_R) = \{\{\rho\}\}^{\log} \{\{u\}\}, \quad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log \rho_L - \log \rho_R}.$$

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

$$\begin{aligned} \mathbf{f}_S(\mathbf{u}, \mathbf{u}) &= \mathbf{f}(\mathbf{u}), & \mathbf{f}_S(\mathbf{u}, \mathbf{v}) &= \mathbf{f}_S(\mathbf{v}, \mathbf{u}), & (\text{consistency, symmetry}) \\ (\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}(\mathbf{u}_L, \mathbf{u}_R) &= \psi_L - \psi_R, & (\text{conservation}) \end{aligned}$$

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

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

$$\frac{\partial f(u)}{\partial x} \implies 2 \frac{\partial f_S(u(x), u(y))}{\partial x} \Big|_{y=x} = \frac{1}{3} \frac{\partial u^2}{\partial x} + \frac{1}{3} u \frac{\partial u}{\partial x} + \frac{1}{3} u^2 \frac{\partial u}{\partial x}$$

- Extends beyond split formulations: mass flux for compressible Euler

$$f_S^\rho(u_L, u_R) = \{\{\rho\}\}^{\log} \{\{u\}\}, \quad \{\{\rho\}\}^{\log} = \frac{\rho_L - \rho_R}{\log \rho_L - \log \rho_R}.$$

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

$$\begin{aligned} \mathbf{f}_S(\mathbf{u}, \mathbf{u}) &= \mathbf{f}(\mathbf{u}), & \mathbf{f}_S(\mathbf{u}, \mathbf{v}) &= \mathbf{f}_S(\mathbf{v}, \mathbf{u}), & (\text{consistency, symmetry}) \\ (\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}(\mathbf{u}_L, \mathbf{u}_R) &= \psi_L - \psi_R, & (\text{conservation}) \end{aligned}$$

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

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

$$\frac{\partial f(u)}{\partial x} \implies 2 \frac{\partial f_S(u(x), u(y))}{\partial x} \Big|_{y=x} = \frac{1}{3} \frac{\partial u^2}{\partial x} + \frac{1}{3} u \frac{\partial u}{\partial x} + \frac{1}{3} u^2 \frac{\partial u}{\partial x}$$

- Extends beyond split formulations: mass flux for compressible Euler

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

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  $\mathbf{F}_S$  as evaluation of  $\mathbf{f}_S$  at all combinations of quadrature points

$$(\mathbf{F}_S)_{ij} = \mathbf{f}_S(u(\mathbf{x}_i), u(\mathbf{x}_j)), \quad \mathbf{x} = [\mathbf{x}^q, \mathbf{x}^f]^T.$$

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

$$2 \frac{\partial f_S(u(x), u(y))}{\partial x} \Big|_{y=x} \implies [\mathbf{P}_q \quad \mathbf{L}_f] \operatorname{diag}(2\mathbf{D}_N \mathbf{F}_S).$$

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

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

# A discretely entropy conservative DG method

Theorem (Chan 2018)

Let  $\mathbf{u}_h(\mathbf{x}, t) = \sum_j \hat{\mathbf{u}}_j(t) \phi_j(\mathbf{x})$  and  $\tilde{\mathbf{u}} = \mathbf{u} \begin{pmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{pmatrix} \mathbf{P}_q \mathbf{v}$ . Let  $\hat{\mathbf{u}}$  locally solve

$$\mathbf{M} \frac{d\hat{\mathbf{u}}}{dt} + \sum_{i=1}^d \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (2\mathbf{Q}_N^i \circ \mathbf{F}_S^i) \mathbf{1} + \mathbf{V}_f^T \mathbf{W}_f (\mathbf{f}_S^i(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}}) - \mathbf{f}^i(\tilde{\mathbf{u}})) \mathbf{n}_i = 0.$$

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

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

- Uses  $\tilde{\mathbf{u}} = \mathbf{u}(\mathbf{P}_N \mathbf{v})$  in terms of  **$L^2$  projected** entropy variables  $\mathbf{P}_N \mathbf{v}$ !
- Add interface dissipation (e.g. Lax-Friedrichs) for entropy **inequality**.

# A discretely entropy conservative DG method

Theorem (Chan 2018)

Let  $\mathbf{u}_h(\mathbf{x}, t) = \sum_j \hat{\mathbf{u}}_j(t) \phi_j(\mathbf{x})$  and  $\tilde{\mathbf{u}} = \mathbf{u} \begin{pmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{pmatrix}$ . Let  $\hat{\mathbf{u}}$  locally solve

$$\frac{d\hat{\mathbf{u}}}{dt} + \sum_{i=1}^d [\mathbf{P}_q \quad \mathbf{L}_f] (2\mathbf{D}_N^i \circ \mathbf{F}_S^i) \mathbf{1} + \mathbf{L}_f (\mathbf{f}_S^i(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}}) - \mathbf{f}^i(\tilde{\mathbf{u}})) \mathbf{n}_i = 0.$$

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

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

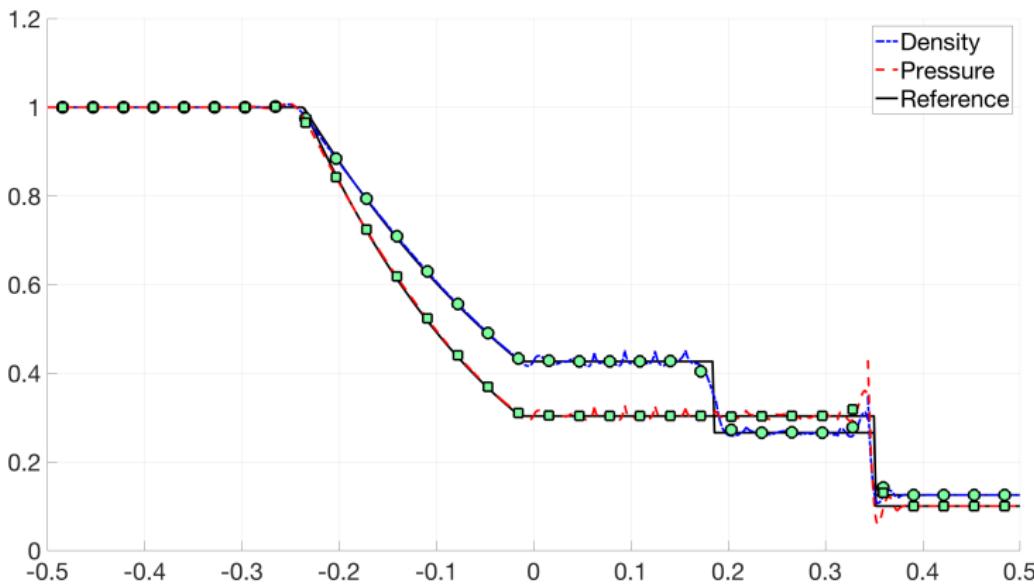
- Uses  $\tilde{\mathbf{u}} = \mathbf{u}(\mathbf{P}_N \mathbf{v})$  in terms of  **$L^2$  projected** entropy variables  $\mathbf{P}_N \mathbf{v}$ !
- Add interface dissipation (e.g. Lax-Friedrichs) for entropy **inequality**.

# Talk outline

- 1 DG as summation by parts finite differences
- 2 Entropy stable formulations through flux differencing
- 3 Numerical experiments: triangular and tetrahedral meshes

# 1D Sod shock tube

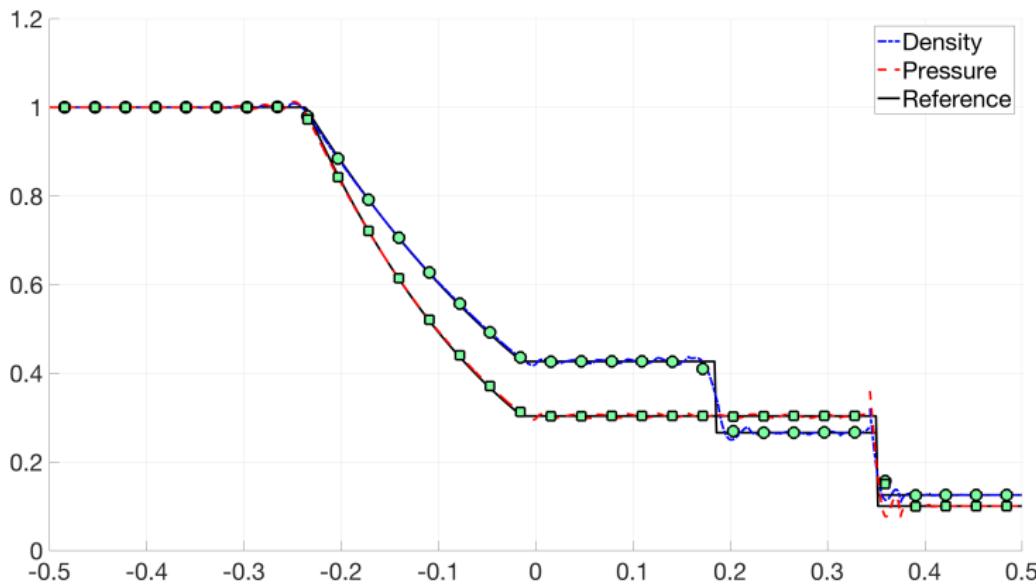
- Circles are cell averages. No limiting or artificial viscosity applied.
- CFL of .125 used for both GLL- $(N + 1)$  and GQ- $(N + 2)$ .



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

# 1D Sod shock tube

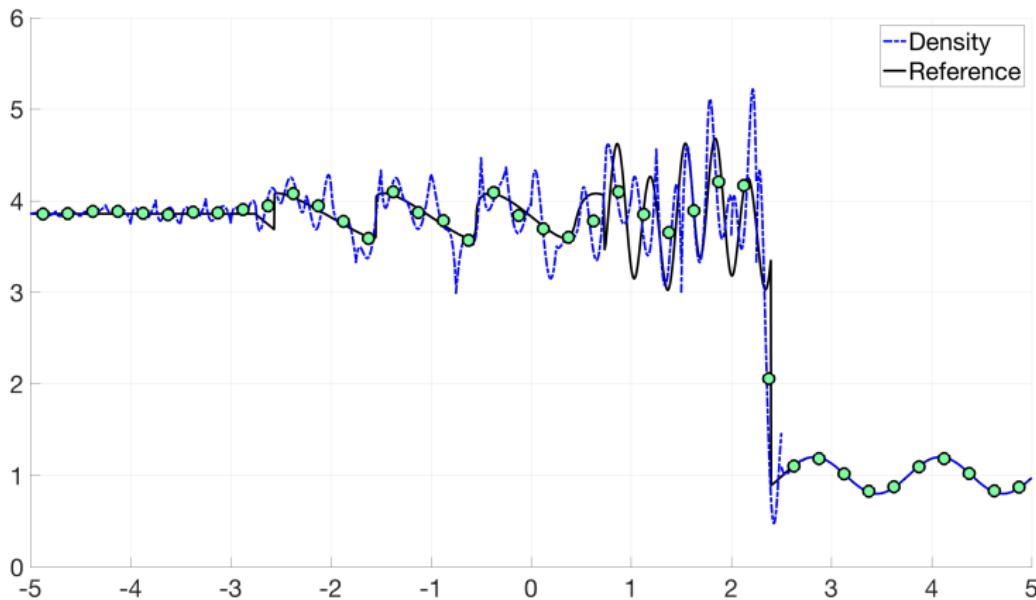
- Circles are cell averages. No limiting or artificial viscosity applied.
- CFL of .125 used for both GLL- $(N + 1)$  and GQ- $(N + 2)$ .



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

# 1D sine-shock interaction

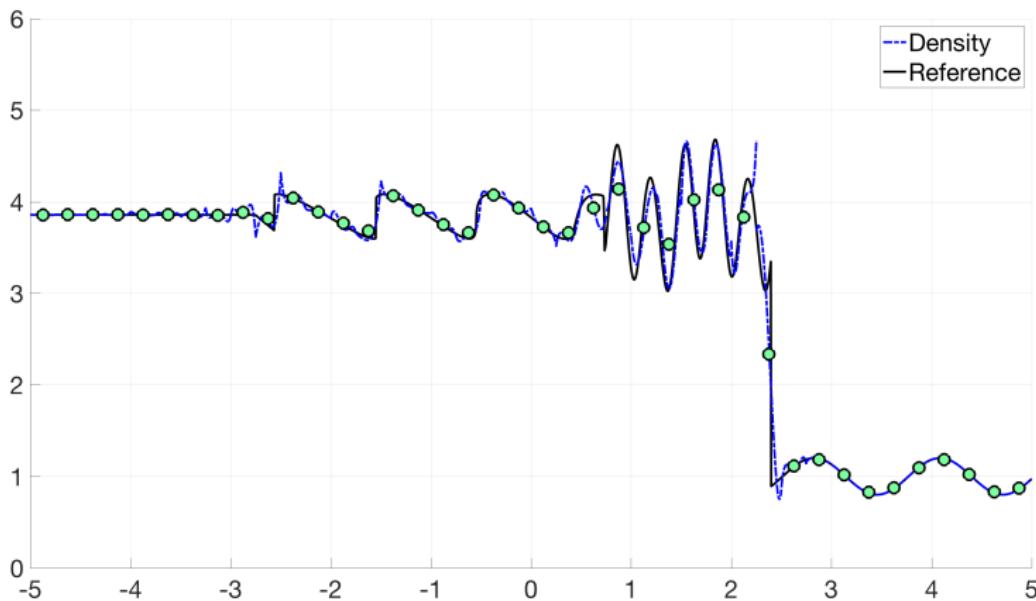
- GQ- $(N + 2)$  needs smaller CFL (.05 vs .125) for stability.



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

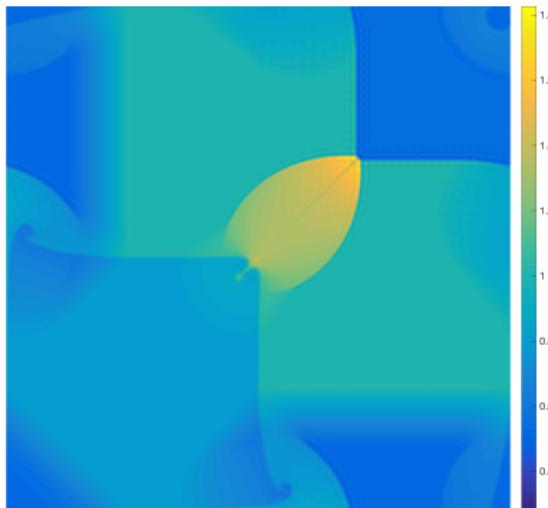
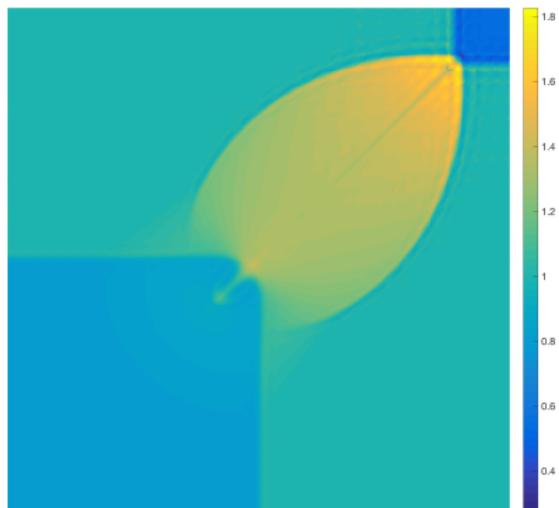
# 1D sine-shock interaction

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



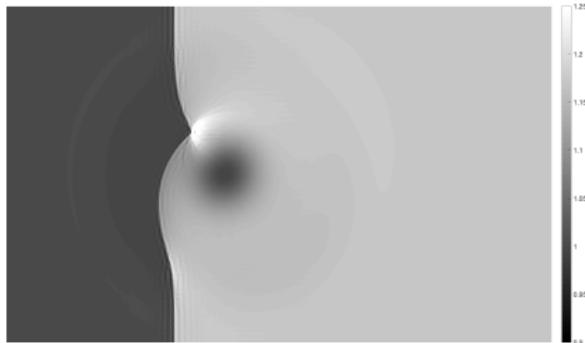
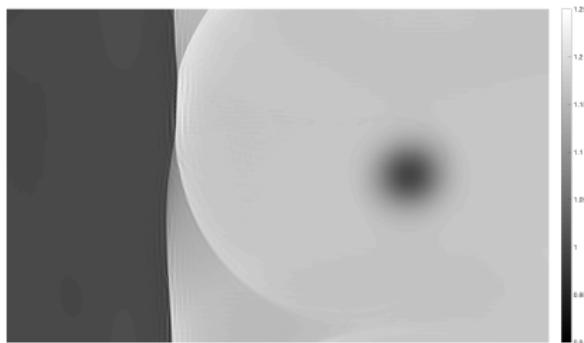
$N = 4, K = 40, \text{CFL} = .05, (N + 2)$  point Gauss quadrature.

# 2D Riemann problem

(a)  $\Omega = [-1, 1]^2$ (b)  $\Omega = [-.5, .5]^2$ ,  $32 \times 32$  elements

- Uniform  $64 \times 64$  mesh:  $N = 3$ , CFL .125, Lax-Friedrichs stabilization.
- No limiting or artificial viscosity required to maintain stability!
- Periodic on larger domain (“natural” boundary conditions unstable).

# 2D shock-vortex interaction

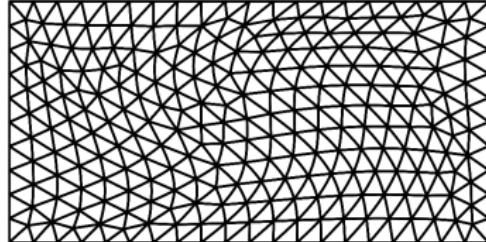
(a)  $t = .3$ (b)  $t = .7$ 

- Vortex passing through a shock on a periodic domain (matrix dissipation, degree  $N = 3$  approximation, mesh size  $h = 1/128$ ).
- Entropy stable wall boundary conditions for GSBP still needed.

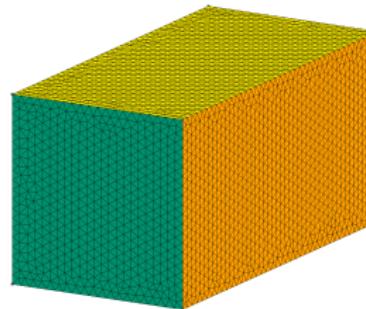
---

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

## 2D and 3D isentropic vortex: affine and curved meshes



(a) 2D triangular mesh



(b) 3D tetrahedral mesh

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

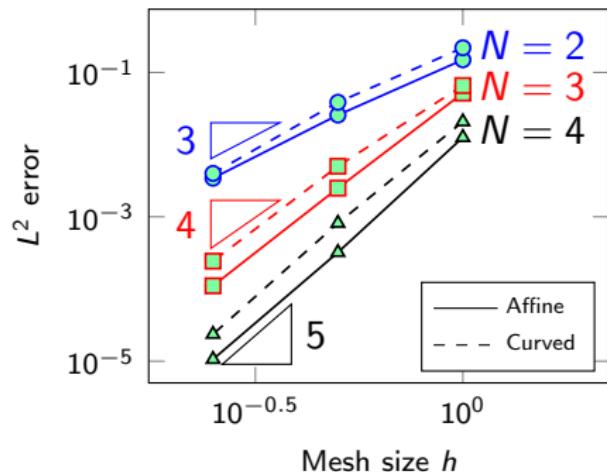
- Entropy stability needs discrete geometric conservation law (GCL).
- Generalized mass lumping: weight-adjusted mass matrices.
- Modify  $\tilde{\mathbf{u}} = \mathbf{u}(\tilde{\mathbf{v}})$ ,  $\tilde{\mathbf{v}} = \tilde{P}_N^k \mathbf{v}(\mathbf{u}_h)$  using weight-adjusted projection  $\tilde{P}_N^k$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

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

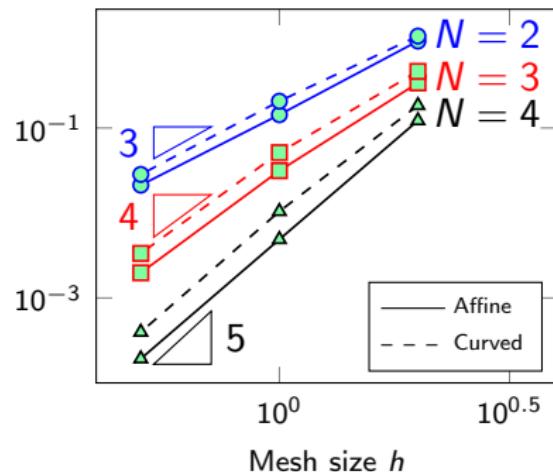
Kopriva (2006). Metric identities and the discontinuous spectral element method on curvilinear meshes.

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

# 2D and 3D isentropic vortex: affine and curved meshes



(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

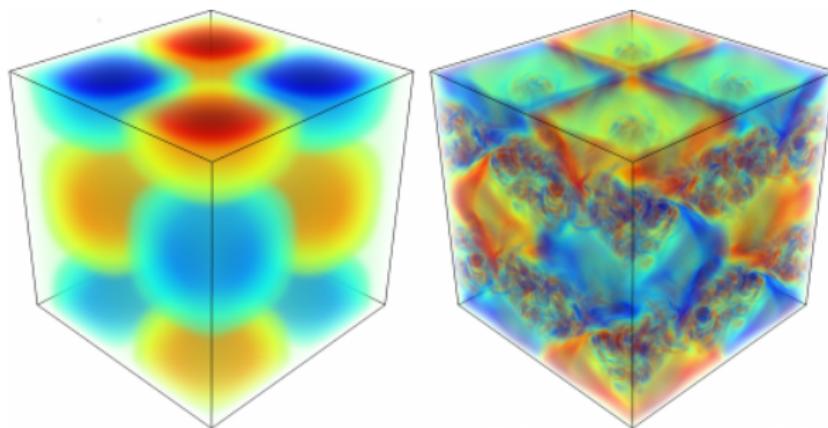
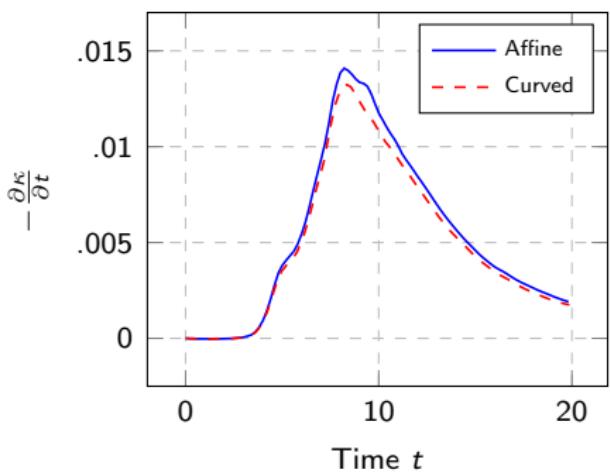
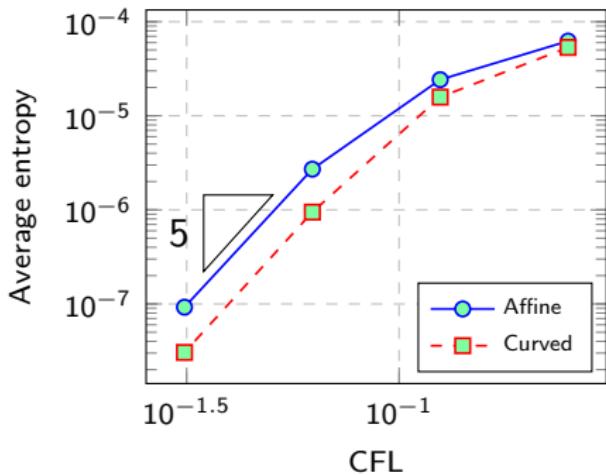


Figure: Isocontours of z-vorticity for Taylor-Green at  $t = 0, 10$  seconds.

- Simple turbulence-like behavior (generation of small scales).
- Inviscid Taylor-Green: tests robustness w.r.t. under-resolved solutions.

## 3D inviscid Taylor-Green vortex: KE dissipation rate

(a) KE dissipation rate ( $N = 3$ ,  $h = \pi/8$ )(b)  $\int_{\Omega} U(\mathbf{u}) \rightarrow 0$  without dissipation

- Kinetic energy dissipation rate: good agreement with literature.
- $\int_{\Omega} U(\mathbf{u}) \rightarrow 0$  as  $\text{CFL} \rightarrow 0$  for entropy conservative scheme.

# Summary and future work

- More general class of high order DG methods which satisfy a semi-discrete entropy inequality.
- Challenges: BCs, positivity, strong shocks, computational cost.
- Current work: SWE, Gauss collocation (with DCDR Fernandez, M. Carpenter), adaptivity + hybrid meshes, multi-GPU (with L. Wilcox).
- 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

# Entropy stable shallow water fluxes

- 1D shallow water equations

$$\begin{aligned}\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} &= 0 \\ \frac{\partial(hu)}{\partial t} + \frac{\partial}{\partial x} \left( hu^2 + \frac{1}{2}gh^2 \right) &= -gh \frac{\partial b}{\partial x}.\end{aligned}$$

- Entropy stable finite volume fluxes (include  $b$  for well-balanced)

$$\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) = \begin{bmatrix} \{\{hu\}\} \\ \{\{hu\}\} \{\{u\}\} + \frac{1}{2}gh_Lh_R \end{bmatrix}$$

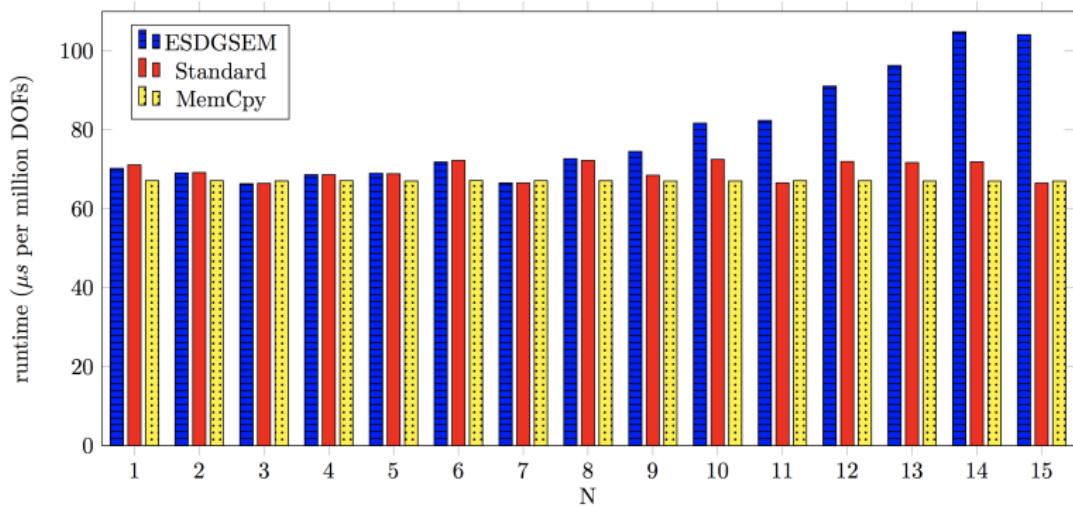
- For collocation: equivalent to **split** form of momentum equation.

$$\frac{\partial hu}{\partial t} + \frac{1}{2} \left( \frac{\partial hu^2}{\partial x} + hu \frac{\partial u}{\partial x} \right) + gh \frac{\partial(h+b)}{\partial x} = 0.$$

---

Wintermeyer, Winters, Gassner, Warburton (2018). *An entropy stable discontinuous Galerkin method for the shallow water equations on curvilinear meshes with wet/dry fronts accelerated by GPUs*.

# GPUs and flux differencing: when FLOPS are free



- High arithmetic intensity: compute while waiting for global memory.
- On GPUs, extra operations don't increase runtime until  $N \geq 9$ !

---

Wintermeyer, Winters, Gassner, Warburton (2018). *An entropy stable discontinuous Galerkin method for the shallow water equations on curvilinear meshes with wet/dry fronts accelerated by GPUs*.

# Over-integration is ineffective without $L^2$ projection

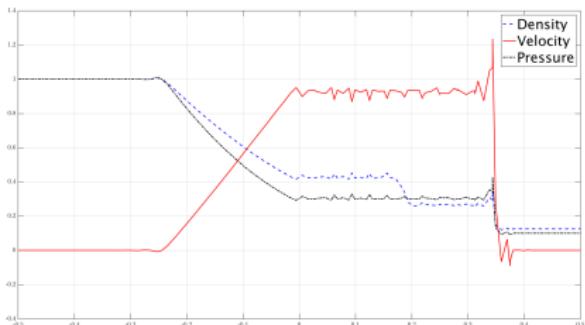
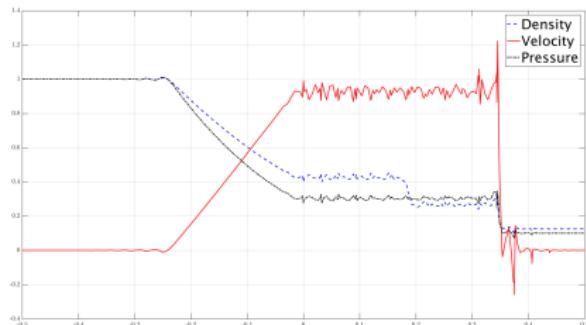
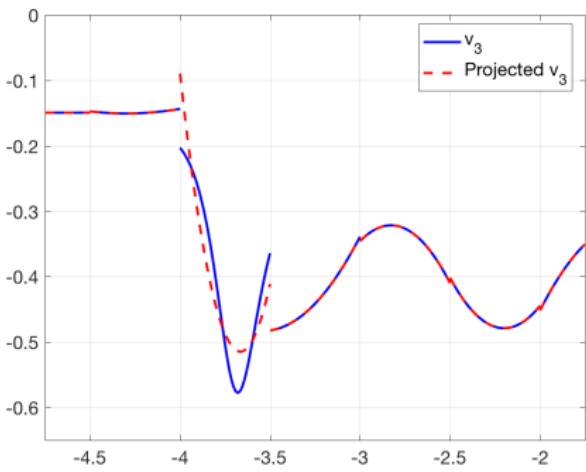
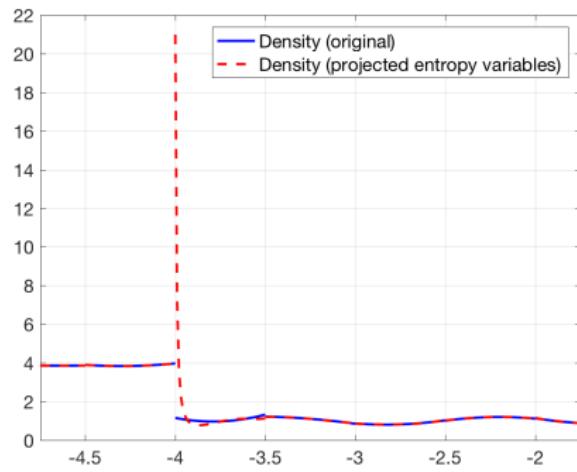
(a) Degree  $N$  GLL,  $(N + 1)$  points(b) Degree  $N$  GLL,  $(N + 4)$  points

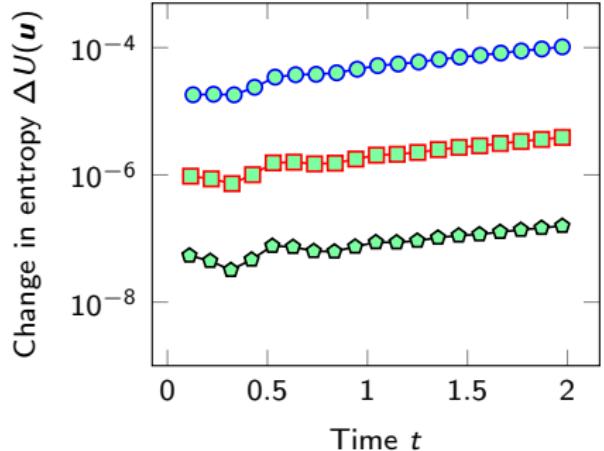
Figure: Sod shock tube for  $N = 4$  and  $K = 32$  elements. Over-integrating by increasing the number of quadrature points does not improve solution quality.

# On CFL restrictions

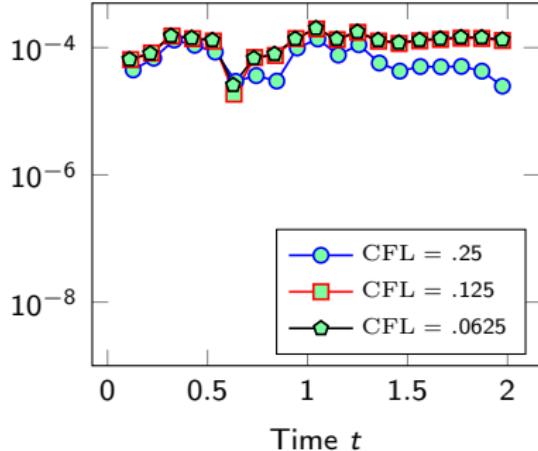
- For GLL- $(N + 1)$  quadrature,  $\tilde{\mathbf{u}} = \mathbf{u}(P_N \mathbf{v}) = \mathbf{u}$  at GLL points.
- For GQ- $(N + 2)$ , discrepancy between  $L^2$  projection and interpolation.
- Still need **positivity** of thermodynamic quantities for stability!

(a)  $v_3(x), (P_N v_3)(x)$ (b)  $\rho(x), \rho((P_N \mathbf{v})(x))$

# 2D curved meshes: conservation of entropy



(c) With weight-adjusted projection



(d) Without weight-adjusted projection

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