Entropy stable discontinuous Galerkin methods using arbitrary bases and quadratures

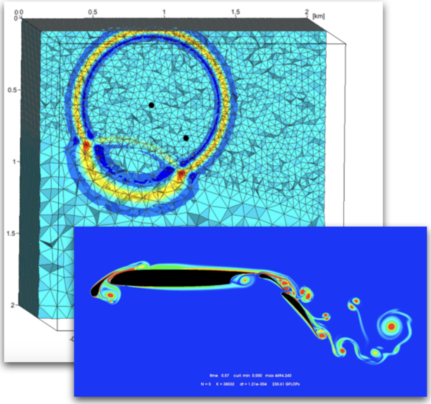
Jesse Chan

[1]Department of Computational and Applied Math

ECCM-CFD 2018
June 9-15, 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).

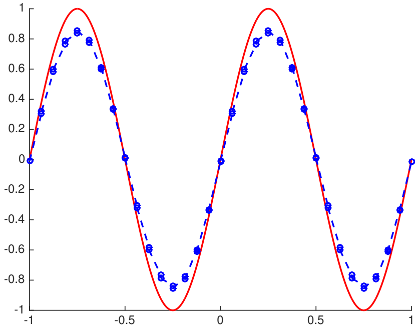


Figures courtesy of T. Warburton, A. Modave.

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).

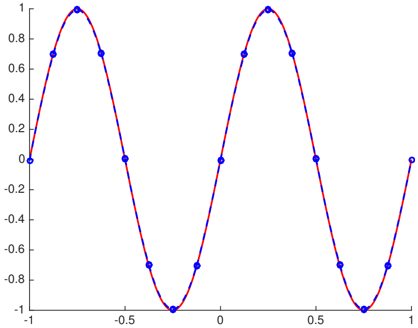


Fine linear approximation.

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).

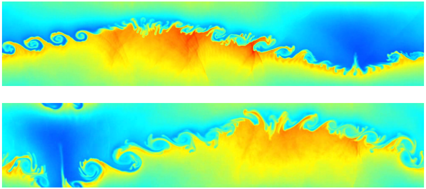


Figure from Per-Olof Persson.

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).

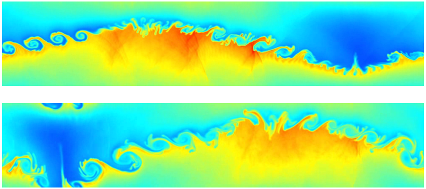
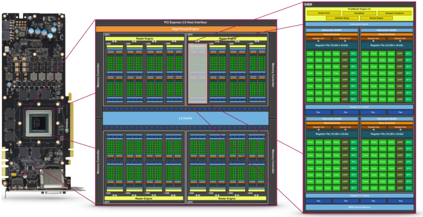


Figure from Per-Olof Persson.

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).

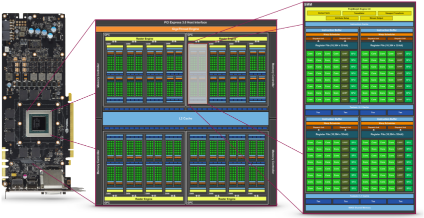


A graphics processing unit (GPU).

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!

Entropy stability for nonlinear conservation laws

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

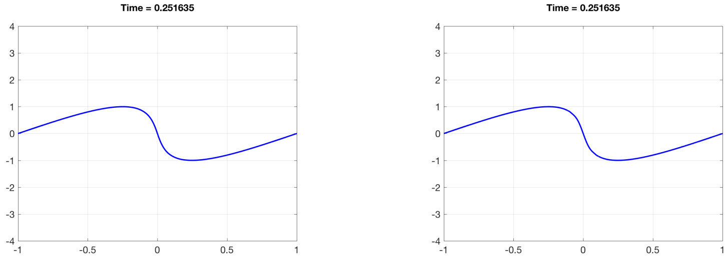
$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = 0.$$

- 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 chain rule, integration by parts.

Why are discretizations of nonlinear PDEs so unstable?



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

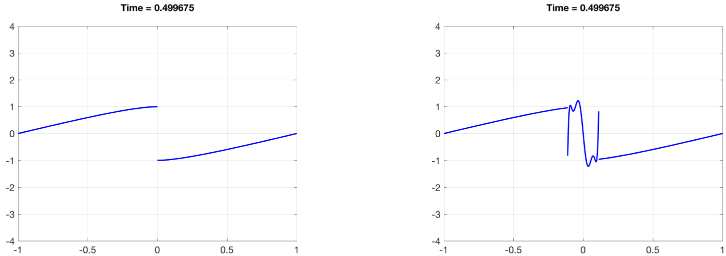
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

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

- 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 discretizations of nonlinear PDEs so unstable?



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

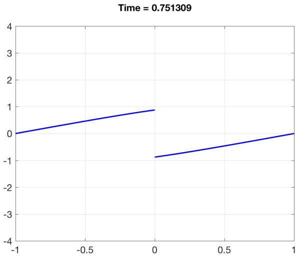
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

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

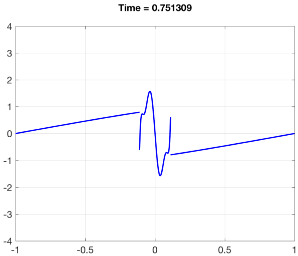
- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

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

Why are discretizations of nonlinear PDEs so unstable?



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



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

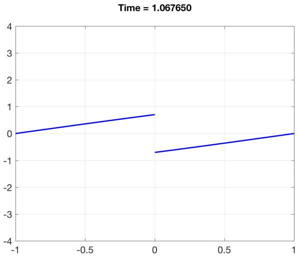
- Burgers' equation: $f(u) = u^2/2$. How to compute $\frac{\partial}{\partial x} f(u)$?

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

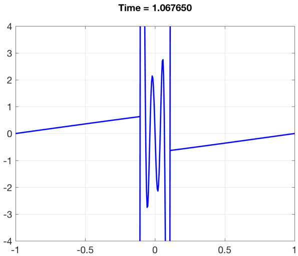
- Differentiating L^2 projection P_N + inexact quadrature: no chain rule.

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

Why are 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 $\frac{\partial}{\partial x} f(u)$?

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

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

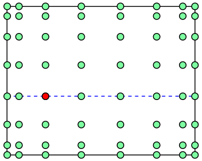
$$\int_{D^k} \left(\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x} P_N u^2 \right) v \, \mathrm{d}x = 0, \qquad \frac{1}{2}\frac{\partial P_N u^2}{\partial x} \neq P_N \left(u \frac{\partial u}{\partial x} \right)$$

Talk outline

1. Summation by parts methods
2. Entropy stable formulations and flux differencing
3. Numerical experiments

Talk outline

Overview of entropy stable high order SBP schemes



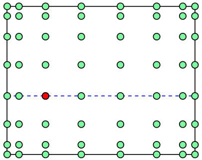
(a) GLL collocation

- 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, wedges, pyramids: over-integration?

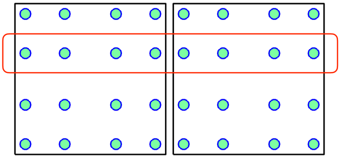
Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher and Carpenter (2013), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), Crean, Hicken, et al. (2018).

Overview of entropy stable high order SBP schemes



(a) GLL collocation



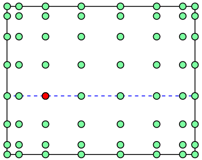
(b) Gauss nodes coupling

- 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, wedges, pyramids: over-integration?

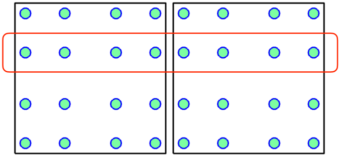
Goals: entropy stability, compact coupling, arbitrary basis/quadrature.

Fisher and Carpenter (2013), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), Crean, Hicken, et al. (2018).

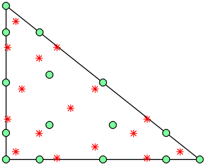
Overview of entropy stable high order SBP schemes



(a) GLL collocation



(b) Gauss nodes coupling



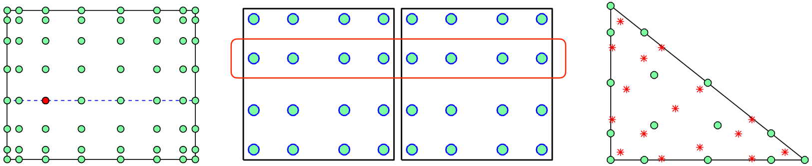
(c) Nodes vs cubature

- 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, wedges, pyramids: over-integration?

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

Fisher and Carpenter (2013), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), Crean, Hicken, et al. (2018).

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, wedges, pyramids: over-integration?

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

Fisher and Carpenter (2013), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), Crean, Hicken, et al. (2018).

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 $2N-1$ (volume) and $2N$ (surface). Define diagonal 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 differentiation matrix $\boldsymbol{D}^i$, interpolation matrices $\boldsymbol{V}_q, \boldsymbol{V}_f$, mass matrix $\boldsymbol{M}$

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

- Useful operators: quadrature-based L^2 projection and lifting matrices

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

- $\boldsymbol{D}_q^i = \boldsymbol{V}_q\boldsymbol{D}^i\boldsymbol{P}_q$: evaluates ith derivative of L^2 projection at $\boldsymbol{x}_i^q$.

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

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 $2N-1$ (volume) and $2N$ (surface). Define diagonal weight matrices

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

- Assume some polynomial basis $\phi_1, \ldots, \phi_{N_p}$. Define differentiation matrix $\boldsymbol{D}^i$, interpolation matrices $\boldsymbol{V}_q, \boldsymbol{V}_f$, mass matrix $\boldsymbol{M}$

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

- Useful operators: quadrature-based L^2 projection and lifting matrices

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

- $\boldsymbol{D}_q^i = \boldsymbol{V}_q \boldsymbol{D}^i \boldsymbol{P}_q$: evaluates ith derivative of L^2 projection at $\boldsymbol{x}_i^q$.

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

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 $2N-1$ (volume) and $2N$ (surface). Define diagonal 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 differentiation matrix $\boldsymbol{D}^i$, interpolation matrices $\boldsymbol{V}_q, \boldsymbol{V}_f$, mass matrix $\boldsymbol{M}$

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

- Useful operators: quadrature-based L^2 projection and lifting matrices

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

- $\boldsymbol{D}_q^i = \boldsymbol{V}_q \boldsymbol{D}^i \boldsymbol{P}_q$: evaluates ith derivative of L^2 projection at $\boldsymbol{x}_i^q$.

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

A "decoupled" block SBP operator

- Decoupled SBP: improve approx. by incorporating boundary points:

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

- $\boldsymbol{D}_N^i$ produces a high order approximation of $f\frac{\partial g}{\partial x}$ at $\boldsymbol{x} = [\boldsymbol{x}^q, \boldsymbol{x}^f]$.

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

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

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

- $\boldsymbol{D}_N^i$ also satisfies a summation-by-parts (SBP) property

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{W} & \\ & \boldsymbol{W}_f \end{bmatrix} \boldsymbol{D}_N^i, \qquad \boldsymbol{Q}_N^i + \left(\boldsymbol{Q}_N^i\right)^T = \begin{bmatrix} 0 & \\ & \boldsymbol{W}_f \boldsymbol{n}_i \end{bmatrix}$$

A "decoupled" block SBP operator

- Decoupled SBP: improve approx. by incorporating boundary points:

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

- $\boldsymbol{D}_N^i$ produces a high order approximation of $f\frac{\partial g}{\partial x}$ at $\boldsymbol{x} = [\boldsymbol{x}^q, \boldsymbol{x}^f]$.

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

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

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

- $\boldsymbol{D}_N^i$ also satisfies a summation-by-parts (SBP) property

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{W} & \\ & \boldsymbol{W}_f \end{bmatrix} \boldsymbol{D}_N^i, \qquad \boldsymbol{Q}_N^i + \left(\boldsymbol{Q}_N^i\right)^T = \begin{bmatrix} 0 & \\ & \boldsymbol{W}_f \boldsymbol{n}_i \end{bmatrix}$$

A "decoupled" block SBP operator

- Decoupled SBP: improve approx. by incorporating boundary points:

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

- $\boldsymbol{D}_N^i$ produces a high order approximation of $f\frac{\partial g}{\partial x}$ at $\boldsymbol{x} = [\boldsymbol{x}^q, \boldsymbol{x}^f]$.

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

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

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

- $\boldsymbol{D}_N^i$ also satisfies a summation-by-parts (SBP) property

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{W} & \\ & \boldsymbol{W}_f \end{bmatrix} \boldsymbol{D}_N^i, \qquad \boldsymbol{Q}_N^i + \left(\boldsymbol{Q}_N^i\right)^T = \begin{bmatrix} 0 & \\ & \boldsymbol{W}_f \boldsymbol{n}_i \end{bmatrix}$$

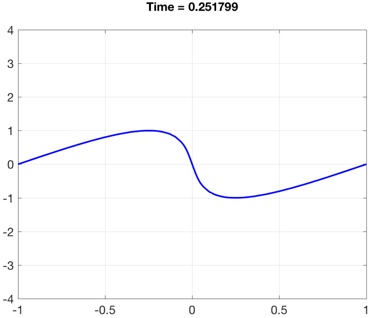
Talk outline

1 Summation by parts methods

2 Entropy stable formulations and flux differencing

3 Numerical experiments

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

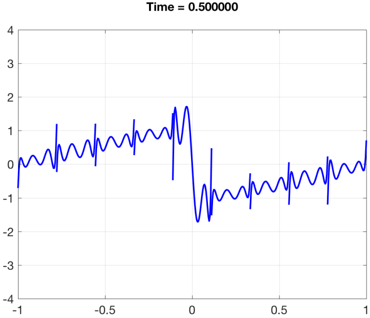


(a) Energy conservative (b) Energy stable

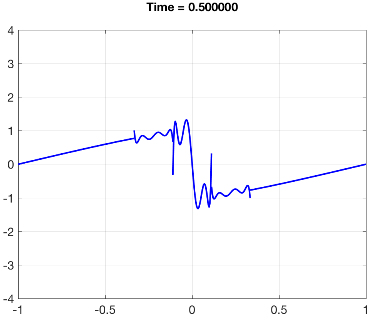
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

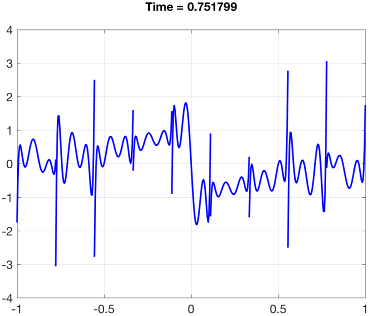


(b) Energy stable

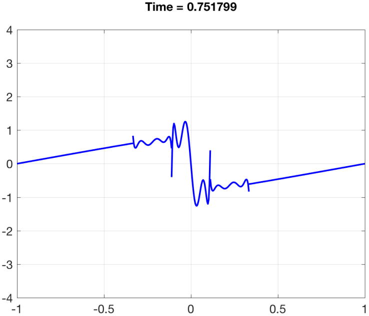
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

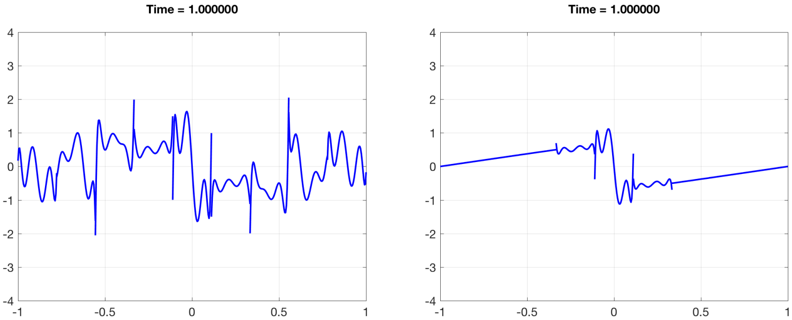


(b) Energy stable

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

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

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

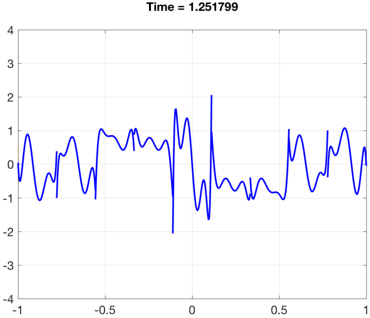


(a) Energy conservative (b) Energy stable

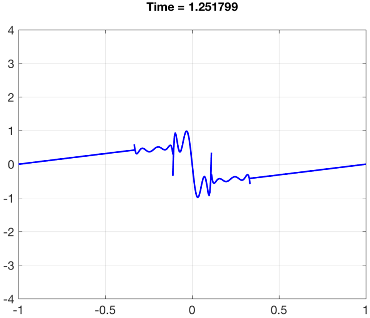
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

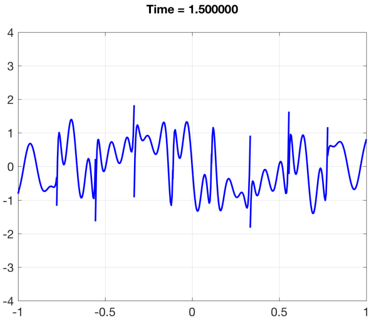


(b) Energy stable

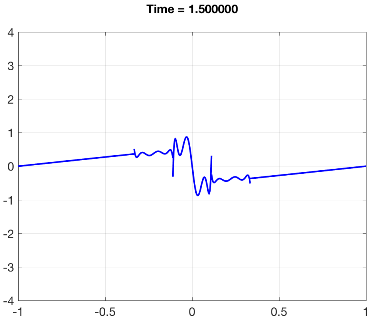
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

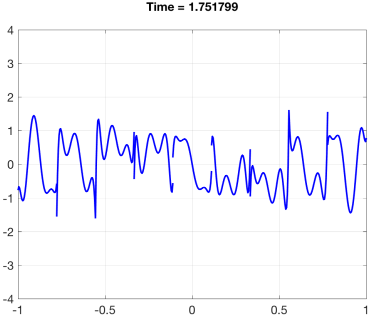


(b) Energy stable

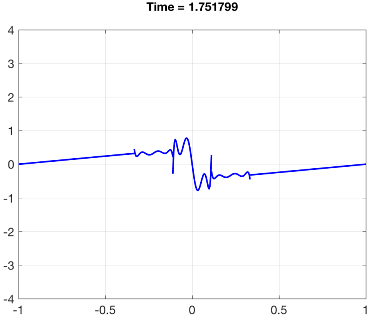
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

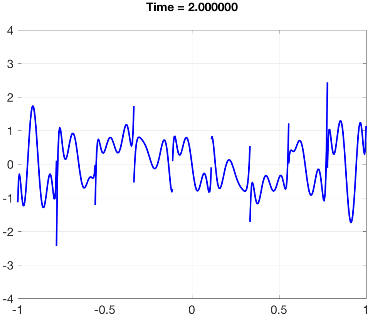


(b) Energy stable

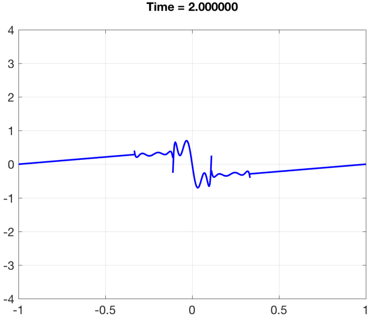
$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widehat{\boldsymbol{u}}, \qquad \boldsymbol{f}^* = \boldsymbol{f}^*(u^+, u) = \text{numerical flux}$$

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

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

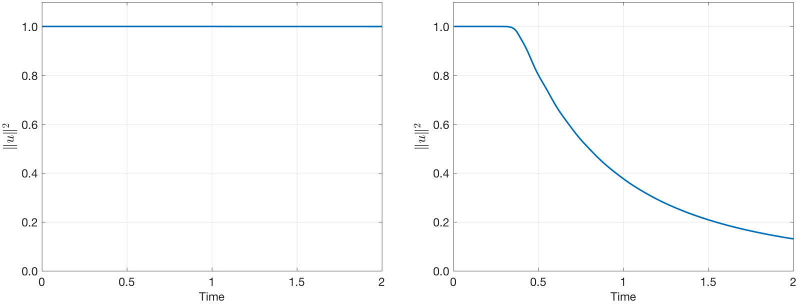


(b) Energy stable

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

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

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

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

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

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.*

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.*

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.*

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 $u_L = u(x), u_R = u(y)$

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

- 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.*

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.*

Flux differencing: implementational details

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

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

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

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

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

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

A discretely entropy conservative DG method

Theorem (Chan 2018)

Let $\boldsymbol{u}_h(\boldsymbol{x}) = \sum_j \widehat{\boldsymbol{u}}_j \phi_j(\boldsymbol{x})$ *and* $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(P_N \boldsymbol{v}\right)$. *Let* $\widehat{\boldsymbol{u}}$ *locally solve*

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

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

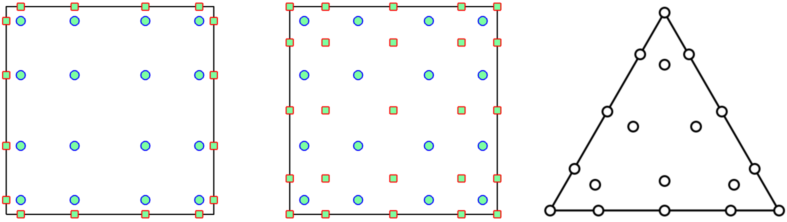
$$\int_\Omega \frac{\partial S(\boldsymbol{u}_h)}{\partial t} + \sum_{i=1}^{d} \int_{\partial\Omega} \left(\left(P_N \boldsymbol{v}\right)^T \boldsymbol{f}^i(\widetilde{\boldsymbol{u}}) - \psi_i(\widetilde{\boldsymbol{u}})\right)\boldsymbol{n}_i = 0.$$

- Need to modify $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(P_N \boldsymbol{v}\right)$ for projected entropy variables $P_N \boldsymbol{v}$!
- 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.*

Unifying (some) finite difference SBP formulations



(a) Generalized SBP (b) Staggered-grid (c) Triangular SBP

- Specific basis and quadrature: recovers several existing entropy stable schemes (DG-SEM, Gauss collocation, staggered grid, etc).
- Setting $\boldsymbol{P}_q = \boldsymbol{I}$ recovers non-basis SBP finite differences on simplices.
- Generates new inter-element coupling terms (SBP-SATs).

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

Crean, Hicken, et al. (2018). *Entropy-stable SBP discretization of the Euler equations on general curved elements.*

Talk outline

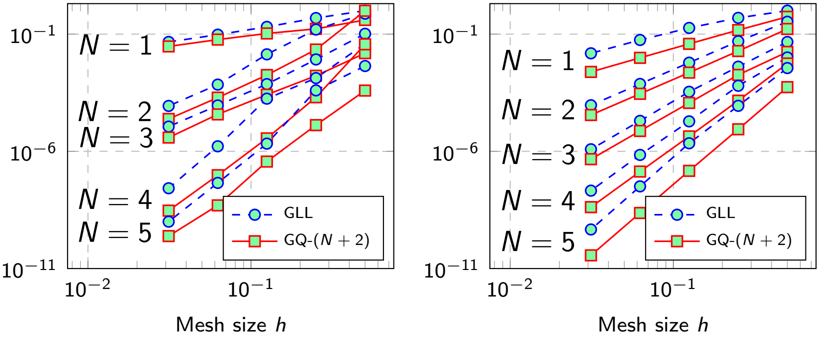
1 Summation by parts methods

2 Entropy stable formulations and flux differencing

3 Numerical experiments

1D compressible Euler equations

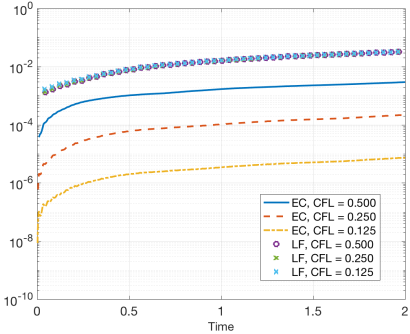
- Inexact Gauss-Legendre-Lobatto (GLL) vs Gauss (GQ) quadratures.
- Entropy conservative (EC) and Lax-Friedrichs (LF) fluxes.
- No additional stabilization, filtering, or limiting.



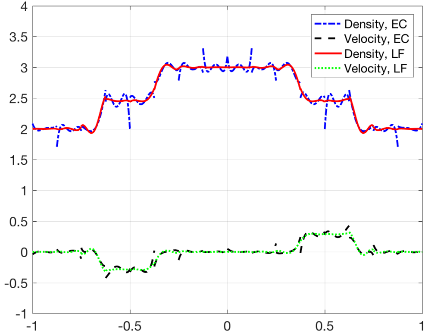
(a) Entropy conservative flux (b) With Lax-Friedrichs penalization

Conservation of entropy: fully discrete schemes

- Entropy conservation: *semi-discrete*, not fully discrete.
- $\Delta S(\boldsymbol{u}) = |S(\boldsymbol{u}(x,t)) - S(\boldsymbol{u}(x,0))| \to 0$ as as $\Delta t \to 0$.



(a) $\Delta S(\boldsymbol{u})$ for various Δt

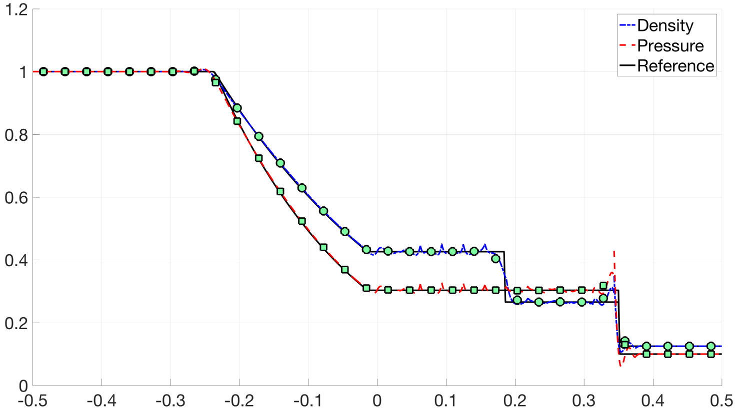


(b) $\rho(x), u(x)$ $(N = 4, K = 16)$

Solution and change in entropy $\Delta S(\boldsymbol{u})$ for entropy conservative (EC) and Lax-Friedrichs (LF) fluxes (using GQ-$(N+2)$ quadrature).

1D Sod shock tube

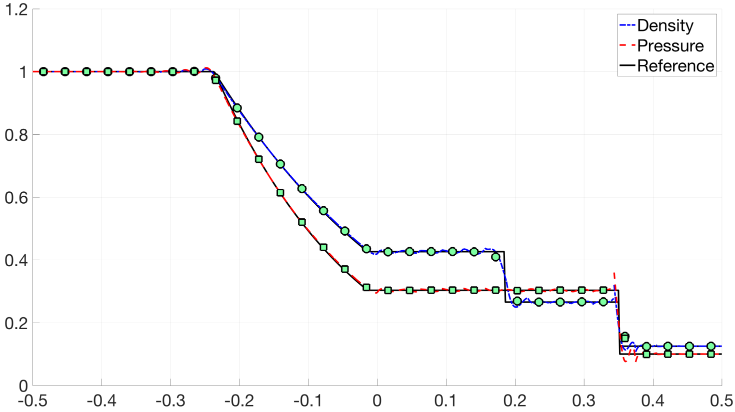
- Circles are cell averages.
- CFL of .125 used for both GLL-$(N+1)$and GQ-$(N+2)$.



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

1D Sod shock tube

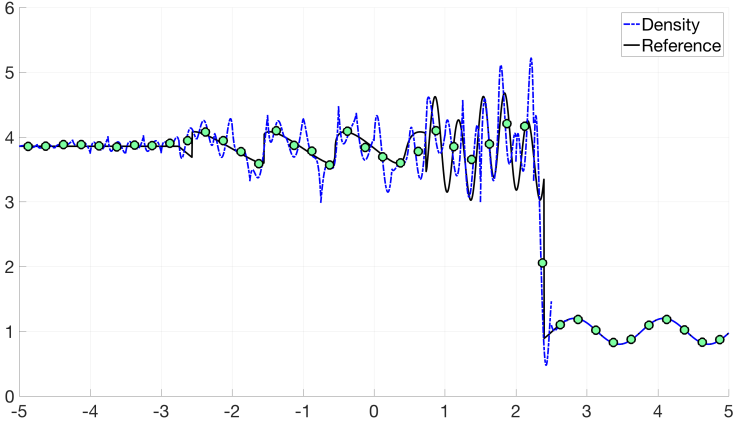
- Circles are cell averages.
- CFL of .125 used for both GLL-$(N+1)$and GQ-$(N+2)$.



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

1D sine-shock interaction

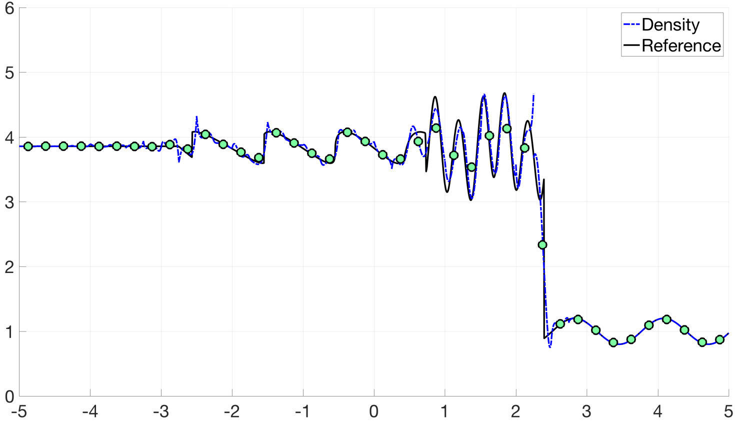
- GQ-$(N+2)$ needs smaller CFL (.05 vs .125) for stability.



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

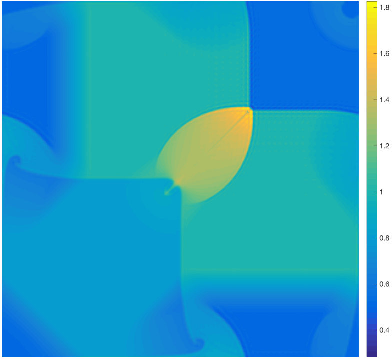
1D sine-shock interaction

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



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

2D Riemann problem

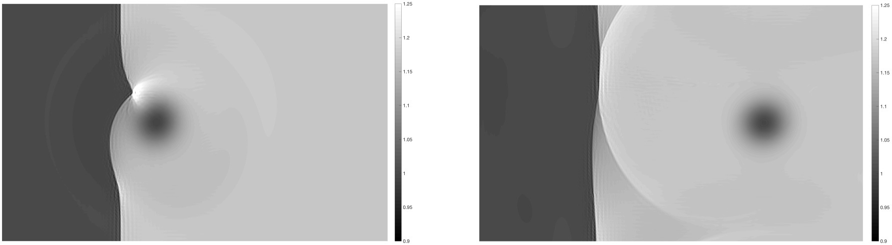


(a) $\Omega = [-1, 1]^2$

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

- Uniform 64×64 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

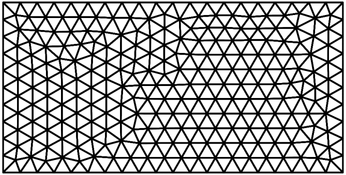


(a) $t = .3$

(b) $t = .7$

- Vortex passing through a shock on a periodic domain.
- Entropy stable wall boundary conditions for GSBP still needed.

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


(a) Affine mesh

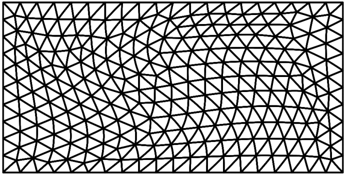
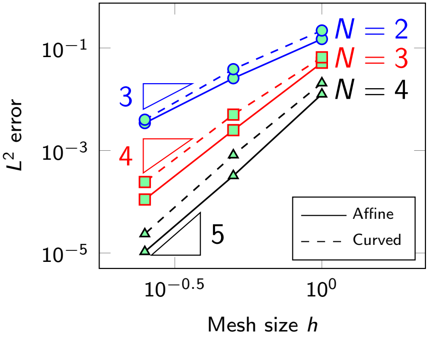

(b) Curved mesh

Figure: Example of an affine and warped 2D mesh (corresponding to $h = 1$).

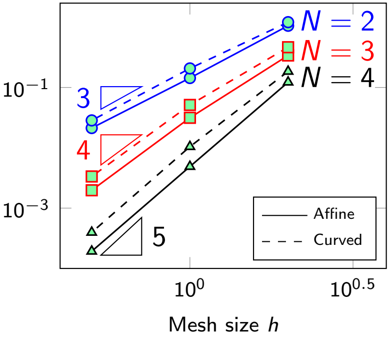
- Entropy stability: needs discrete geometric conservation law (GCL).
- Generalized mass lumping: weight-adjusted mass matrices.
- Modify $\widetilde{\boldsymbol{u}} = \boldsymbol{u}(\widetilde{\boldsymbol{v}})$, $\widetilde{\boldsymbol{v}} = \widetilde{P}_N^k \boldsymbol{v}(\boldsymbol{u}_h)$ using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

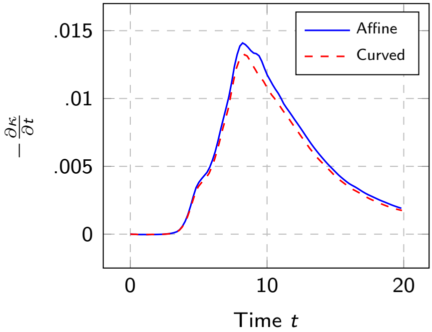


(b) 3D results

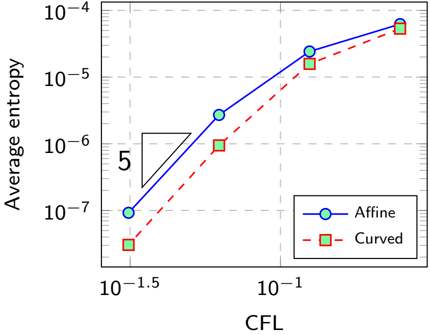
L^2 errors for 2D/3D isentropic vortex at $T = 5$ on affine, curved meshes.

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 ($N = 3$, $h = \frac{\pi}{8}$)



(b) $\int_\Omega U(\boldsymbol{u})$ for entropy cons. scheme

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

Summary and future work

- Discretely stable time-domain high order discontinuous Galerkin methods: provable semi-discrete stability
- Additional work required for strong shocks, positivity preservation.
- Current work: 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.

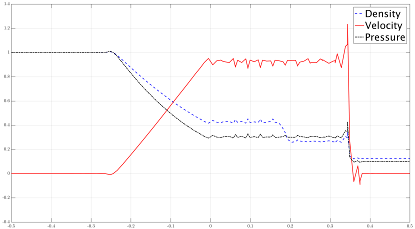
Thank you! Questions?

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

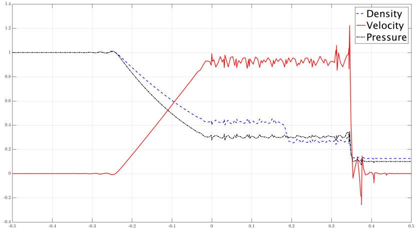
Chan (2017). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods.*

Additional slides

Over-integration is ineffective without L^2 projection



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

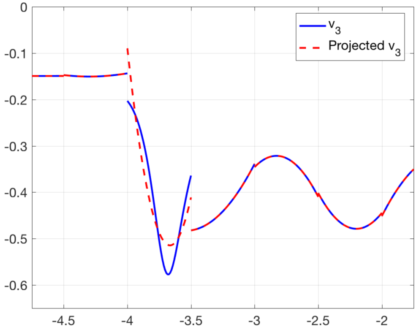


(b) Degree N GLL, $(N+4)$ points

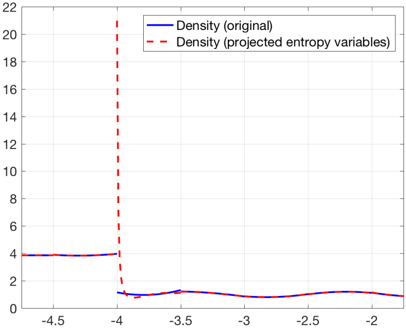
Figure: Sod shock tube for $N = 4$ and $K = 32$ elements. Over-integrating by increasing the number of quadrature points does not improve solution quality.

On CFL restrictions

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

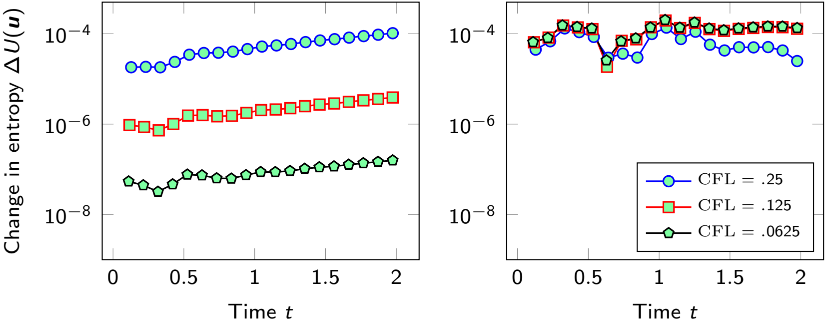


(a) $v_3(x), \left(P_N v_3\right)(x)$



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

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 $\widetilde{\boldsymbol{v}} = P_N \boldsymbol{v}(\boldsymbol{u})$ is $O\left(10^{-14}\right)$.