

Constructing robust high order entropy stable discontinuous Galerkin methods

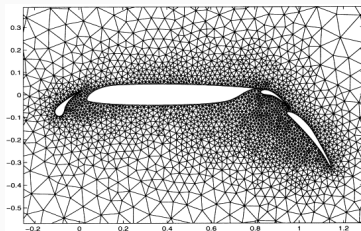
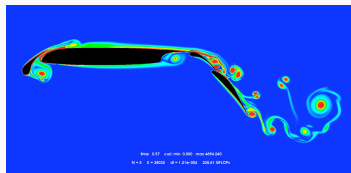
Jesse Chan

Dept. of Computational Applied Mathematics and Operations Research
Rice University

Princeton Plasma Physical Laboratory

High order finite element methods for hyperbolic PDEs

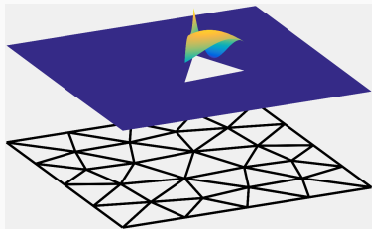
- Aerodynamics applications: acoustics, vorticular flows, turbulence, shocks.
- Goal: **high accuracy** on **unstructured meshes**.
- Discontinuous Galerkin (DG) methods: geometric flexibility + high order.



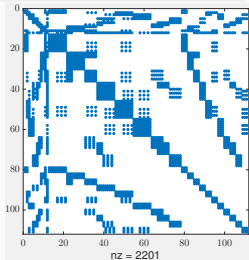
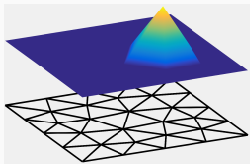
Mesh from Slawig 2001.

High order finite element methods for hyperbolic PDEs

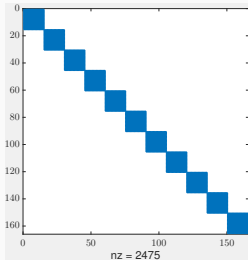
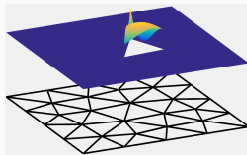
- Aerodynamics applications: acoustics, vorticular flows, turbulence, shocks.
- Goal: **high accuracy** on **unstructured meshes**.
- Discontinuous Galerkin (DG) methods: geometric flexibility + high order.



Why discontinuous Galerkin methods?



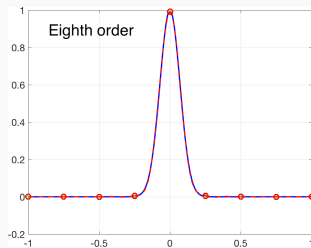
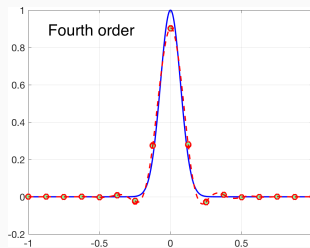
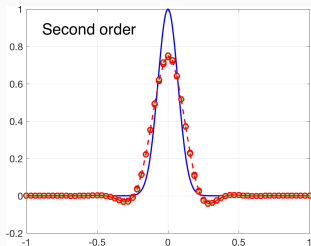
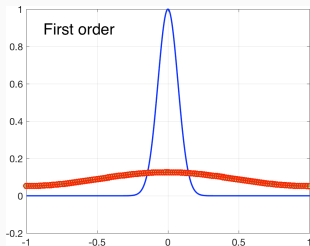
(a) High order FEM



(b) High order DG

High order DG mass matrices: easily invertible for **explicit time-stepping**.

Why high order accuracy?



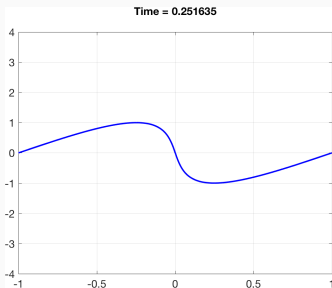
Accurate resolution of propagating vortices and waves.

Why high order accuracy?

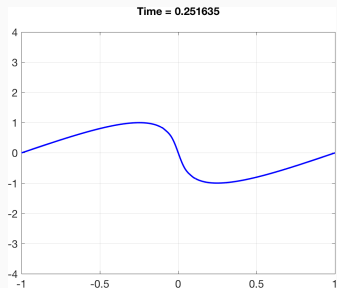


2nd, 4th, and 16th order Taylor-Green vortex. Vorticular structures and acoustic waves are both sensitive to numerical dissipation.

Why *not* high order DG methods?



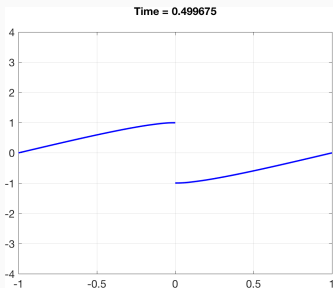
(a) Exact solution



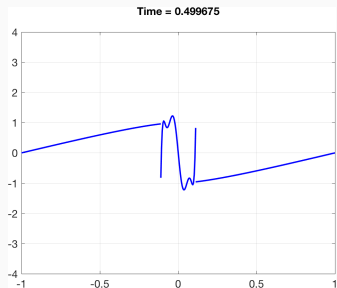
(b) 8th order DG

High order methods blow up for under-resolved solutions of nonlinear conservation laws (e.g., shocks and turbulence).

Why *not* high order DG methods?



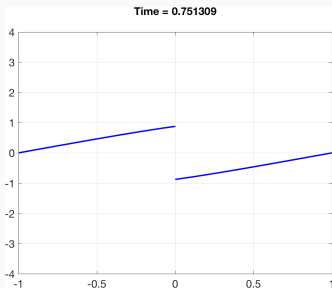
(a) Exact solution



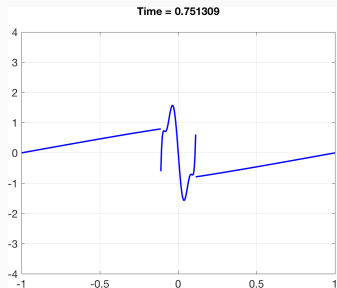
(b) 8th order DG

High order methods blow up for under-resolved solutions of nonlinear conservation laws (e.g., shocks and turbulence).

Why *not* high order DG methods?



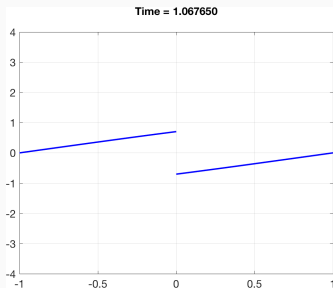
(a) Exact solution



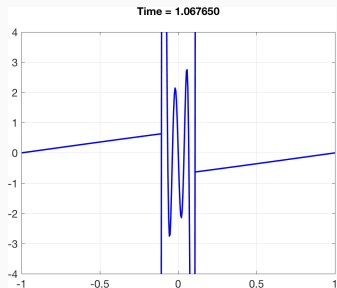
(b) 8th order DG

High order methods blow up for under-resolved solutions of nonlinear conservation laws (e.g., shocks and turbulence).

Why *not* high order DG methods?



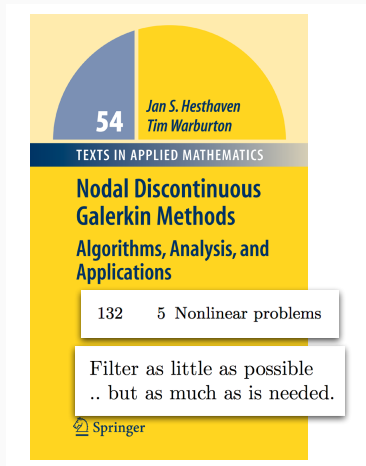
(a) Exact solution



(b) 8th order DG

High order methods blow up for under-resolved solutions of nonlinear conservation laws (e.g., shocks and turbulence).

Why entropy stability for high order schemes?



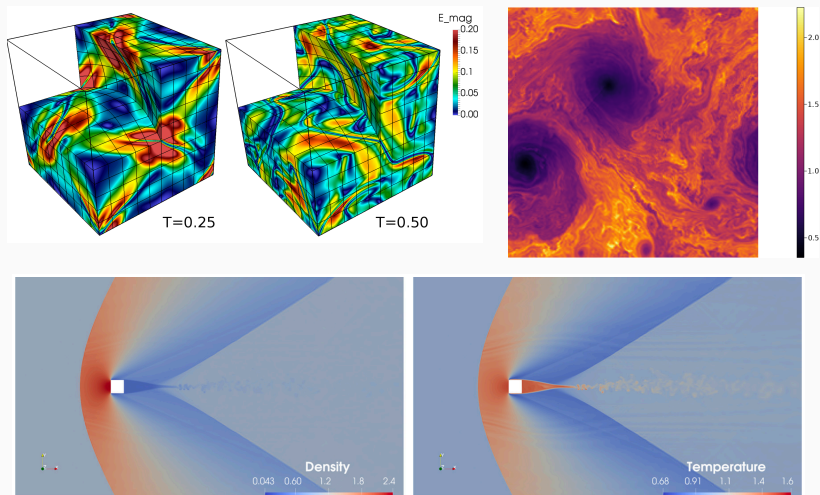
- High order DG needs heuristic stabilization (e.g., artificial viscosity, filtering).
- Entropy stable schemes improve robustness without *no added dissipation*.
- Turns DG into a “good” high order method (though not 100% bulletproof).

Finite volume methods: Tadmor, Chandrashekar, Ray, Svard, Fjordholm, Mishra, LeFloch, Rohde, ...

High order tensor product elements: Fisher, Carpenter, Gassner, Winters, Kopriva, Persson, ...

High order general elements: Chen and Shu, Crean, Hicken, Del Rey Fernandez, Zingg, ...

Examples of high order entropy stable simulations



All simulations are ESDG without artificial viscosity, filtering, or slope limiting.

Bohm et al. (2019). *An entropy stable nodal DG method for the resistive MHD equations. Part I.*

Dalcin et al. (2019). *Conservative and ES solid wall BCs for the compressible NS equations.*

1. A brief introduction to entropy stable nodal DG methods
2. Positivity preserving entropy stable nodal DG for compressible Navier-Stokes (with Yimin Lin)
3. “Modal” entropy stable DG formulations

A brief introduction to entropy stable nodal DG methods

Entropy stability for nonlinear problems

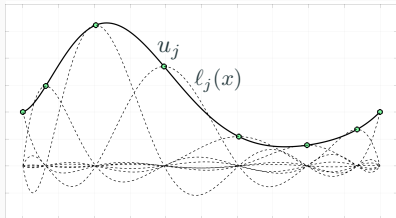
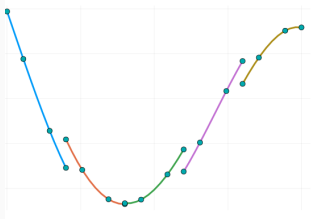
- Energy balance for **nonlinear** conservation laws (Burgers', shallow water, compressible Euler + Navier-Stokes).

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

- Continuous entropy inequality: convex **entropy** function $S(\mathbf{u})$, “entropy potential” $\psi(\mathbf{u})$, entropy variables $\mathbf{v}(\mathbf{u})$

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

A basic intro to nodal discontinuous Galerkin methods



- Multiply by nodal (Lagrange) basis $\ell_i(x)$ and integrate

$$\int_{D^k} \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} \right) \ell_i + \int_{\partial D^k} (\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \mathbf{f}(\mathbf{u}^-)) n \ell_i = 0$$

- The numerical flux $\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) \approx \mathbf{f}(\mathbf{u})$ enforces boundary conditions and weak continuity across interfaces.
- **Nodal** (collocation) DG methods: use Gauss-Lobatto quadrature nodes for both interpolation and integration.

Matrix formulation of nodal DG methods

- Map integrals to the reference interval $\hat{D} = [-1, 1]$

$$\int_{\hat{D}} \left(\frac{h}{2} \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} \right) \ell_i + \int_{\partial \hat{D}} (\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \mathbf{f}(\mathbf{u}^-)) n \ell_i = 0$$

- Matrix formulation: insert $\mathbf{u}(x, t) = \sum_j \mathbf{u}_j(t) \ell_j(x)$

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \mathbf{Q} \mathbf{f}(\mathbf{u}) + \mathbf{E}^T \underbrace{\mathbf{B} (\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \mathbf{f}(\mathbf{u}^-))}_{\text{interface flux}} = \mathbf{0}.$$

where $\mathbf{M} = \frac{h}{2} \text{diag}(w_1, \dots, w_{N+1})$, and $\mathbf{Q}, \mathbf{B}, \mathbf{E}$ are differentiation and boundary matrices

$$\mathbf{Q}_{ij} = \int_{-1}^1 \frac{\partial \ell_j}{\partial x} \ell_i, \quad \mathbf{B} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\mathbf{E} = \begin{bmatrix} \ell_1(-1) & \dots & \ell_{N+1}(-1) \\ \ell_1(1) & \dots & \ell_{N+1}(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}.$$

Reformulating the flux derivative

- Standard DG methods do not yield an entropy inequality (inexact quadrature, no discrete chain rule).
- Solution: reformulate the flux derivative matrix term

$$\int_{-1}^1 \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} \ell_i \approx \mathbf{Q} \mathbf{f}(\mathbf{u}).$$

- Note that $\mathbf{Q} \mathbf{1} = \mathbf{0}$, so $\sum_j \mathbf{Q}_{ij} = 0$. Thus,

$$(\mathbf{Q} \mathbf{f}(\mathbf{u}))_i = \sum_j \mathbf{Q}_{ij} (f(\mathbf{u}_j) + f(\mathbf{u}_i)) = 2 \sum_j \mathbf{Q}_{ij} \underbrace{\frac{f(\mathbf{u}_j) + f(\mathbf{u}_i)}{2}}_{\text{central flux}}$$

- What if we used another numerical flux?

A “flux differencing” formulation

- Let f_{EC} be an **entropy conservative** numerical flux

$$f_{EC}(\mathbf{u}, \mathbf{u}) = f(\mathbf{u}), \quad (\text{consistency})$$

$$f_{EC}(\mathbf{u}, \mathbf{v}) = f_{EC}(\mathbf{v}, \mathbf{u}), \quad (\text{symmetry})$$

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

- Replace the central flux with

$$(\mathbf{Q}f(\mathbf{u}))_i = 2 \sum_j \mathbf{Q}_{ij} \frac{f(\mathbf{u}_j) + f(\mathbf{u}_i)}{2} \approx 2 \sum_j \mathbf{Q}_{ij} f_{EC}(\mathbf{u}_i, \mathbf{u}_j).$$

- Compact notation using the Hadamard product

$$2 \sum_j \mathbf{Q}_{ij} f_{EC}(\mathbf{u}_i, \mathbf{u}_j) = (2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1})_i, \quad \mathbf{F}_{ij} = f_{EC}(\mathbf{u}_i, \mathbf{u}_j).$$

Example of EC fluxes (compressible Euler equations)

- Define average $\{\{u\}\} = \frac{1}{2}(u_L + u_R)$. In one dimension:

$$f_S^1(\mathbf{u}_L, \mathbf{u}_R) = \{\{\rho\}\}^{\log} \{\{u\}\}$$

$$f_S^2(\mathbf{u}_L, \mathbf{u}_R) = \{\{u\}\} f_S^1 + p_{\text{avg}}$$

$$f_S^3(\mathbf{u}_L, \mathbf{u}_R) = (E_{\text{avg}} + p_{\text{avg}}) \{\{u\}\},$$

$$p_{\text{avg}} = \frac{\{\{\rho\}\}}{2 \{\{\beta\}\}}, \quad E_{\text{avg}} = \frac{\{\{\rho\}\}^{\log}}{2 \{\{\beta\}\}^{\log} (\gamma - 1)} + \frac{1}{2} u_L u_R.$$

- Non-standard logarithmic mean, “inverse temperature” β

$$\{\{u\}\}^{\log} = \frac{u_L - u_R}{\log u_L - \log u_R}, \quad \beta = \frac{\rho}{2p}.$$

Extension to multiple elements

- The nodal DG formulation can be rewritten as:

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \mathbf{Q} \mathbf{f}(\mathbf{u}) + \mathbf{E}^T \mathbf{B} \underbrace{(\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \mathbf{f}(\mathbf{u}^-))}_{\text{interface flux}} = \mathbf{0}.$$

- If \mathbf{Q} satisfies the summation-by-parts (SBP) property

$$\mathbf{Q} + \mathbf{Q}^T = \mathbf{E}^T \mathbf{B} \mathbf{E}$$

and if $\mathbf{f}^*(\mathbf{u}^+, \mathbf{u})$ is entropy stable (e.g., local Lax-Friedrichs flux), a quadrature version of the cell entropy inequality holds:

$$\int_{D^k} \frac{\partial S(\mathbf{u})}{\partial t} + \int_{\partial D^k} \left(\mathbf{v}^T \mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \psi(\mathbf{u}) \right) n \leq 0.$$

Extension to multiple elements

- The nodal DG formulation can be rewritten as:

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1} + \mathbf{E}^T \mathbf{B} \underbrace{\left(f^*(\mathbf{u}^+, \mathbf{u}^-) - f(\mathbf{u}^-) \right)}_{\text{interface flux}} = \mathbf{0}.$$

- If \mathbf{Q} satisfies the summation-by-parts (SBP) property

$$\mathbf{Q} + \mathbf{Q}^T = \mathbf{E}^T \mathbf{B} \mathbf{E}$$

and if $f^*(\mathbf{u}^+, \mathbf{u})$ is entropy stable (e.g., local Lax-Friedrichs flux), a quadrature version of the cell entropy inequality holds:

$$\int_{D^k} \frac{\partial S(\mathbf{u})}{\partial t} + \int_{\partial D^k} \left(\mathbf{v}^T f^*(\mathbf{u}^+, \mathbf{u}^-) - \psi(\mathbf{u}) \right) n \leq 0.$$

Extension to multiple elements

- The nodal DG formulation can be rewritten as:

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1} + \mathbf{E}^T \mathbf{B} \underbrace{\left(\mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \mathbf{f}(\mathbf{u}^-) \right)}_{\text{interface flux}} = \mathbf{0}.$$

- If \mathbf{Q} satisfies the **summation-by-parts (SBP)** property

$$\mathbf{Q} + \mathbf{Q}^T = \mathbf{E}^T \mathbf{B} \mathbf{E}$$

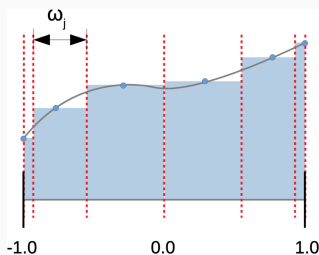
and if $\mathbf{f}^*(\mathbf{u}^+, \mathbf{u})$ is **entropy stable** (e.g., local Lax-Friedrichs flux), a quadrature version of the cell entropy inequality holds:

$$\boxed{\int_{D^k} \frac{\partial S(\mathbf{u})}{\partial t} + \int_{\partial D^k} \left(\mathbf{v}^T \mathbf{f}^*(\mathbf{u}^+, \mathbf{u}^-) - \psi(\mathbf{u}) \right) n \leq 0.}$$

**Positivity preserving entropy stable
nodal DG for compressible
Navier-Stokes (with Yimin Lin)**

Entropy stable schemes require positivity

Entropy stable schemes require positivity of density, pressure (numerical fluxes depend on *logarithm* of density, temperature).



Interpretation of Lobatto nodes as a sub-cell finite volume grid.

- Hard to enforce both high order accuracy and positivity.
- Strategy: blend high order method with a first order positive method to retain **subcell resolution**.

Enforcing positivity: a first order positive subcell scheme

Global matrix formulation using forward Euler time-stepping (extend to higher order via SSP-RK). Let $\mathbf{Q}_{ij} = -\mathbf{Q}_{ji}$, $\mathbf{f}_j = \mathbf{f}(\mathbf{u}_j)$, and $\mathbf{d}_{ij} = \mathbf{d}_{ji} > 0$ for $i \neq j$

$$\mathbf{m}_i \frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i}{\Delta t} + \sum_{j \in N(i)} \mathbf{Q}_{ij} \mathbf{f}_j - \underbrace{\mathbf{d}_{ij} (\mathbf{u}_j - \mathbf{u}_i)}_{\text{algebraic dissipation}} = \mathbf{0}.$$

Use **conservation**, **SBP** properties to rewrite using intermediate “bar states” $\bar{\mathbf{u}}_{ij} = \frac{1}{2} (\mathbf{u}_i + \mathbf{u}_j) - \frac{\mathbf{Q}_{ij}}{\mathbf{d}_{ij}} (\mathbf{f}_j - \mathbf{f}_i)$.

$$\frac{\mathbf{m}_i}{\Delta t} \mathbf{u}_i^{k+1} = \left(\frac{\mathbf{m}_i}{\Delta t} - \sum_{j \neq i} 2\mathbf{d}_{ij} \right) \mathbf{u}_i + \sum_{j \neq i} \frac{2\Delta t \mathbf{d}_{ij}}{\mathbf{m}_i} \bar{\mathbf{u}}_{ij}.$$

Enforcing positivity: a first order positive subcell scheme

Global matrix formulation using forward Euler time-stepping (extend to higher order via SSP-RK). Let $\mathbf{Q}_{ij} = -\mathbf{Q}_{ji}$, $\mathbf{f}_j = \mathbf{f}(\mathbf{u}_j)$, and $\mathbf{d}_{ij} = \mathbf{d}_{ji} > 0$ for $i \neq j$

$$\mathbf{m}_i \frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i}{\Delta t} + \sum_{j \in N(i)} \mathbf{Q}_{ij} \mathbf{f}_j - \underbrace{\mathbf{d}_{ij} (\mathbf{u}_j - \mathbf{u}_i)}_{\text{algebraic dissipation}} = \mathbf{0}.$$

Use **conservation**, **SBP** properties to rewrite using intermediate “bar states” $\bar{\mathbf{u}}_{ij} = \frac{1}{2} (\mathbf{u}_i + \mathbf{u}_j) - \frac{\mathbf{Q}_{ij}}{\mathbf{d}_{ij}} (\mathbf{f}_j - \mathbf{f}_i)$.

$$\frac{\mathbf{m}_i}{\Delta t} \mathbf{u}_i^{k+1} = \left(\frac{\mathbf{m}_i}{\Delta t} - \sum_{j \neq i} 2\mathbf{d}_{ij} \right) \mathbf{u}_i + \sum_{j \neq i} \frac{2\Delta t \mathbf{d}_{ij}}{\mathbf{m}_i} \bar{\mathbf{u}}_{ij}.$$

Provable positivity under a CFL condition

- Bar states $\bar{\mathbf{u}}_{ij}$ resemble a Lax-Friedrichs finite volume update, and **preserve positivity** if \mathbf{d}_{ij} is sufficiently large

$$\bar{\mathbf{u}}_{ij} = \frac{1}{2} (\mathbf{u}_i + \mathbf{u}_j) - \frac{\mathbf{Q}_{ij}}{\mathbf{d}_{ij}} (\mathbf{f}_j - \mathbf{f}_i), \quad \mathbf{d}_{ij} \geq \lambda_{\max}(\mathbf{u}_i, \mathbf{u}_j, \mathbf{Q}_{ij}).$$

- \mathbf{u}_i^{k+1} is positive (a convex combination of \mathbf{u}_i and $\bar{\mathbf{u}}_{ij}$) if

$$\Delta t \leq \min_i \frac{\mathbf{m}_i}{2 \sum_{i \neq j} \mathbf{d}_{ij}}.$$

Our work: extension to compressible Navier-Stokes

- Entropy stable discretization of viscous terms σ , which include the stress τ + heat conduction q .

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \sum_j \mathbf{Q}_{ij} (\mathbf{f}_j - \sigma_j) - \mathbf{d}_{ij} (\mathbf{u}_j - \mathbf{u}_i) = \mathbf{0}.$$

- Reformulate scheme in terms of viscous bar states:

$$\bar{\mathbf{u}}_{ij} = \frac{1}{2} (\mathbf{u}_i + \mathbf{u}_j) - \frac{\mathbf{Q}_{ij}}{\mathbf{d}_{ij}} ((\mathbf{f}_j - \sigma_j) - (\mathbf{f}_i - \sigma_i))$$

- Positivity of ρ, p under a (viscous) CFL condition with

$$\mathbf{d}_{ij} = \max(\beta(\mathbf{u}_i), \beta(\mathbf{u}_j), \lambda_{\max}(\mathbf{u}_i, \mathbf{u}_j, \mathbf{Q}_{ij}), \lambda_{\max}(\mathbf{u}_j, \mathbf{u}_i, \mathbf{Q}_{ji}))$$
$$\beta(\mathbf{u}) > |\mathbf{v} \cdot \mathbf{n}| + \frac{1}{2\rho^2 e} \left(\sqrt{\rho^2 (\mathbf{q} \cdot \mathbf{n})^2 + 2\rho^2 e \|\tau \cdot \mathbf{n} - p\mathbf{n}\|} \right) + \rho |\mathbf{q} \cdot \mathbf{n}|$$

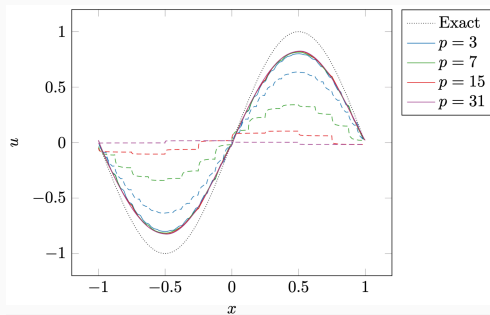
Sparsification of low order matrices

$$\mathbf{Q} = \frac{1}{2} \begin{bmatrix} -1 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 1 \end{bmatrix}$$

$$\mathbf{Q}\mathbf{1} = \mathbf{0}, \quad \underbrace{\mathbf{Q} + \mathbf{Q}^T}_{\text{summation-by-parts property}} = \mathbf{E}^T \mathbf{B} \mathbf{E} \quad .$$

- Note: we use *sparse* SBP operators in the low order method.
- Algebraic artificial dissipation depends on discretization matrices \implies dense operators produce too much diffusion!

Sparsification of low order matrices



Effect of sparsification on solution dissipation; figure taken from Pazner (2021).

- Note: we use *sparse* SBP operators in the low order method.
- **Algebraic** artificial dissipation depends on discretization matrices \implies dense operators produce too much diffusion!

Blending high and low order DG solutions

- Blend high and low order solutions over each element to retain accuracy where possible while ensuring positivity.

$$\mathbf{u}^{k+1} = (1 - \ell)\mathbf{u}^{k+1,\text{low}} + \ell\mathbf{u}^{k+1,\text{high}}$$

- Impose minimal local bounds based on low order solution with relaxation factor α

$$\rho \geq \alpha \rho^{\text{low}}, \quad p \geq \alpha p^{\text{low}}, \quad \alpha \in [0, 1].$$

- Local entropy inequality: preserved for element-wise blending.
- Local conservation: preserved if high and low order schemes use the same interface flux.

Convergence tests: LeBlanc and viscous shock tube

	$N = 2$		$N = 5$	
h	L^1 error	Rate	L^1 error	Rate
0.02	8.681×10^{-2}		5.956×10^{-2}	.
0.01	3.658×10^{-2}	1.25	1.436×10^{-2}	2.05
0.005	1.329×10^{-2}	1.46	3.630×10^{-3}	1.98
0.0025	6.015×10^{-3}	1.14	1.129×10^{-3}	1.69
0.00125	2.910×10^{-3}	1.05	5.889×10^{-4}	0.94

(a) Leblanc shock tube, relaxation factor $\alpha = 0.5$

	$N = 2$		$N = 3$	
h	L^1 error	Rate	L^1 error	Rate
0.025	2.305×10^{-2}		2.071×10^{-2}	
0.0125	9.858×10^{-2}	1.23	6.749×10^{-3}	1.62
0.00625	3.382×10^{-3}	1.54	1.278×10^{-3}	2.40
0.003125	5.765×10^{-4}	2.55	1.163×10^{-4}	3.45
0.0015625	8.836×10^{-5}	2.71	1.269×10^{-5}	3.20

(b) 1D viscous shock, $\text{Re} = 1000$, relaxation factor $\alpha = 0.5$

Viscous shock is run at **Mach 20** to generate positivity violations.

Isentropic vortex with small minimum density

	$N = 2$		$N = 3$		$N = 4$	
h	L^2 error	Rate	L^2 error	Rate	L^2 error	Rate
2.5	1.148×10^0		5.958×10^{-1}	1.28	4.073×10^{-1}	
1.25	4.865×10^{-1}	1.24	1.905×10^{-1}	1.64	8.987×10^{-2}	2.18
0.625	1.223×10^{-1}	1.99	2.308×10^{-2}	3.05	1.511×10^{-2}	2.57
0.3125	1.706×10^{-2}	2.84	2.393×10^{-3}	3.27	1.915×10^{-4}	6.30

(c) Quadrilateral meshes, relaxation factor $\alpha = 0.5$

	$N = 2$		$N = 3$		$N = 4$	
h	L^2 error	Rate	L^2 error	Rate	L^2 error	Rate
2.5	7.887×10^{-1}		5.034×10^{-1}		4.059×10^{-1}	
1.25	3.834×10^{-1}	1.04	1.881×10^{-1}	1.42	9.890×10^{-2}	2.04
0.625	8.993×10^{-2}	2.09	2.944×10^{-2}	2.68	1.578×10^{-2}	2.65
0.3125	1.298×10^{-2}	2.79	2.606×10^{-3}	3.50	4.258×10^{-4}	5.21

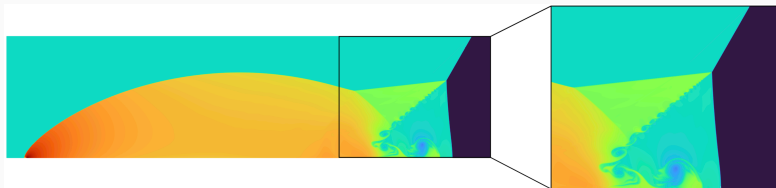
(d) Triangular meshes, relaxation factor $\alpha = 0.5$

Challenging vortex parameters: $\rho_{\min} = 2.145 \times 10^{-3}$!

Compressible Euler: double Mach reflection



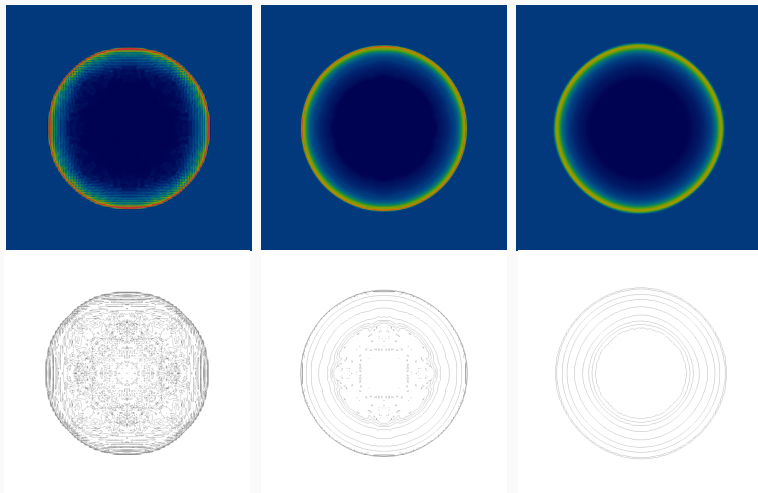
(a) Subcell positivity-preserving entropy stable nodal DG, $\alpha = 0.5$, $T = .2$



(b) Subcell invariant domain preserving nodal DG (Pazner 2021), $T = .275$

Density for $N = 3$ entropy stable DG (250×875 elements) and a reference solution (600×2400 elements). Note: positivity is sensitive to the wall boundary treatment!

Compressible Euler: Sedov blast wave



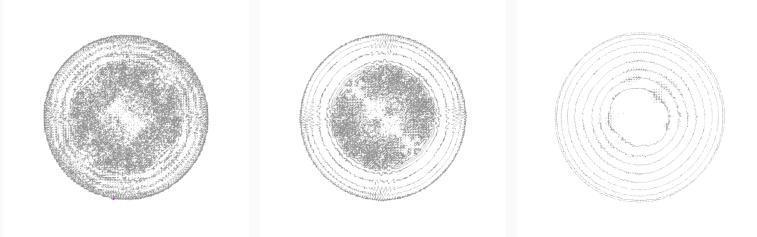
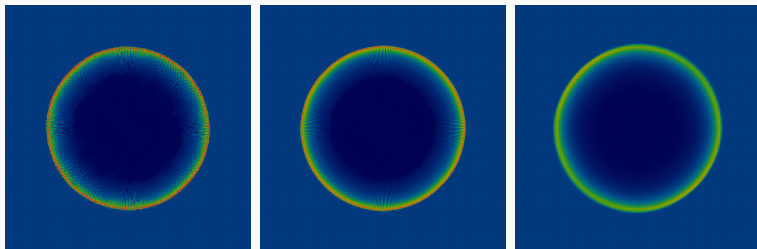
(a) $\alpha = 0.1$

(b) $\alpha = 0.5$

(c) $\alpha = 0.1 + \text{shock capturing}$

Quadrilateral meshes with 100^2 degree $N = 3$ elements.

Compressible Euler: Sedov blast wave



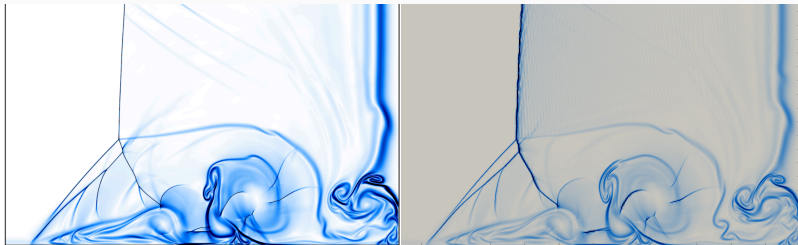
(a) $\alpha = 0.1$

(b) $\alpha = 0.5$

(c) $\alpha = 0.1$ + shock capturing

Triangular meshes with 100^2 degree $N = 3$ elements.

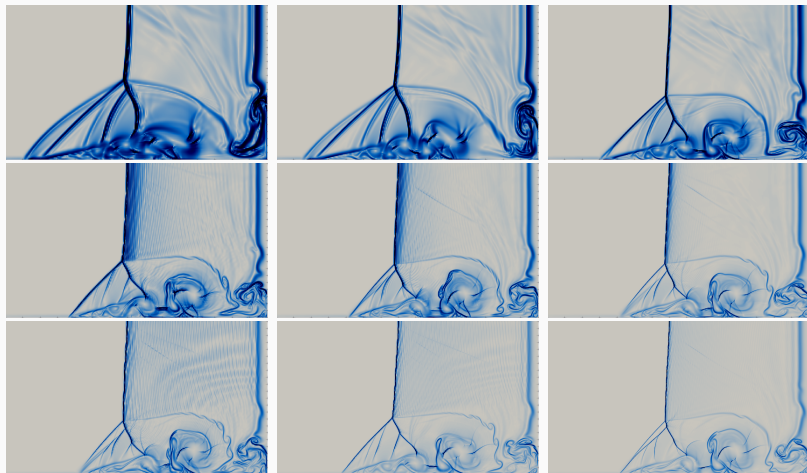
Compressible Navier-Stokes: Daru-Tenaud shock tube



(a) Reference solution (512M nodes) (b) Degree $N = 3$, 600×300 grid

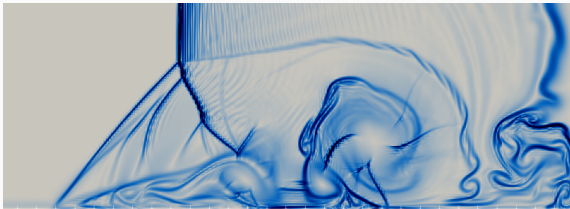
Comparison with a “grid-converged” reference solution from Guermond et al. (2022).

Sensitivity to polynomial degree and mesh size

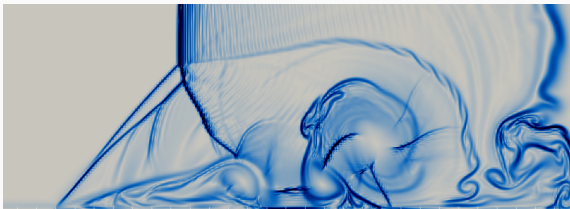


Polynomial degrees $N = 1, 2, 3$ (rows) and $300 \times 150, 400 \times 200, 600 \times 300$ grid resolutions (columns). The limiting relaxation factor is $\alpha = 0.1$.

Sensitivity to limiting parameters ($Re = 1000$)

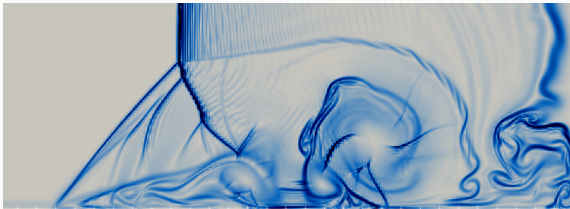


Degree $N = 2$, 200×100 mesh with positivity parameter $\alpha = 0.1$.

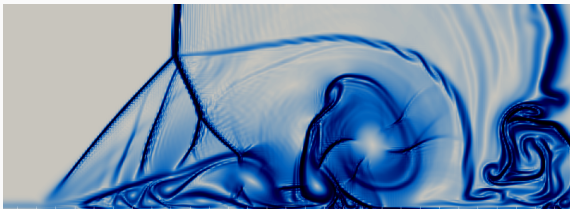


Same setup with positivity parameter $\alpha = 0.5$.

Sensitivity to limiting parameters ($Re = 1000$)

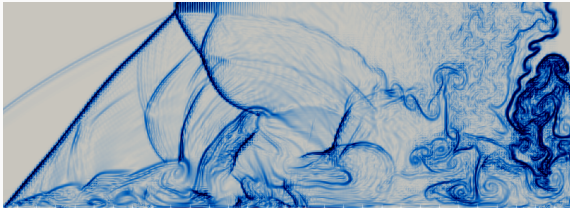


Degree $N = 2$, 200×100 mesh with positivity parameter $\alpha = 0.1$.

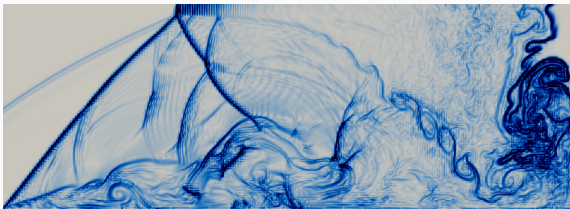


With Hennemann (2021) shock capturing - similar to Dzanic & Witherden!

Sensitivity to limiting parameters ($Re = 10000$)

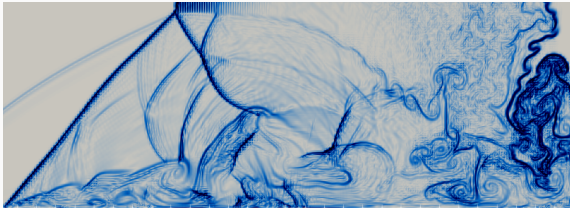


Degree $N = 2$, 200×100 mesh with positivity parameter $\alpha = 0.1$.

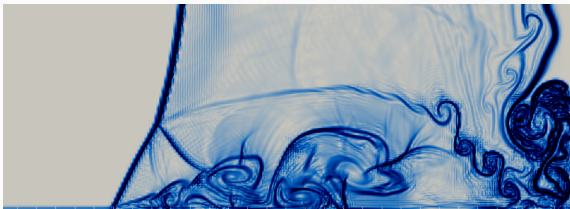


Same setup with positivity parameter $\alpha = 0.5$.

Sensitivity to limiting parameters ($Re = 10000$)



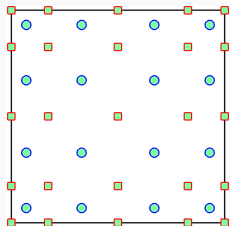
Degree $N = 2$, 200×100 mesh with positivity parameter $\alpha = 0.1$.



Positivity parameter $\alpha = 0.1$ with Hennemann (2021) shock capturing.

“Modal” entropy stable DG formulations

Beyond nodal formulations: entropy projection



For non-located methods (e.g., staggered grid, modal), entropy stability requires interpolating via the “entropy projection”

$$\tilde{u} = u(\Pi_N v(u))$$

$\Pi_N = L^2$ projection onto degree N polynomials.

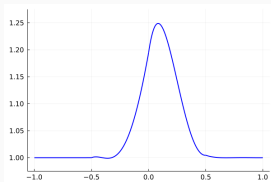
- Entropy projection recovers nodal collocation for appropriate choices of quadrature for the L^2 projection.
- Entropy stable modal formulations also require boundary correction terms for high order accuracy (Chan 2018, 2019).

Parsani, Carpenter, Fisher, Nielsen (2016). *Entropy stable staggered grid discontinuous spectral collocation methods of any order for the compressible Navier-Stokes equations*.

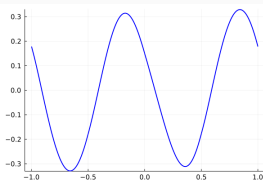
Fernandez, Crean, Carpenter, Hicken (2019). *Staggered-grid entropy-stable multidimensional summation-by-parts discretizations on curvilinear coordinates*.

Pazner, Persson (2019). *Analysis and entropy stability of the line-based discontinuous Galerkin method*.

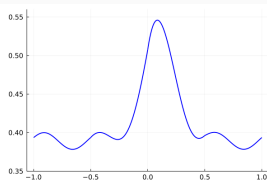
Illustration of the entropy projection



(a) ρ

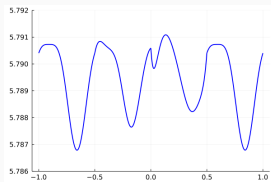


(b) u

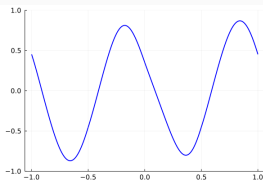


(c) p

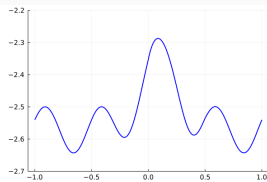
Primitive variables ρ , u , p and their entropy projection.



(d) v_1



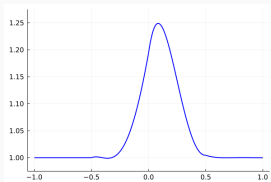
(e) v_2



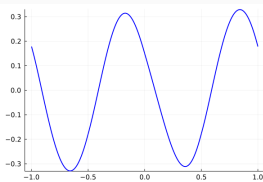
(f) v_3

Entropy variables and their L^2 projection.

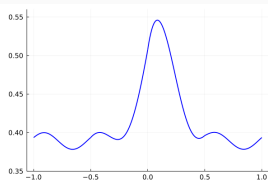
Illustration of the entropy projection



(a) ρ

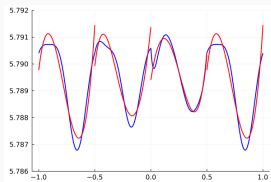


(b) u

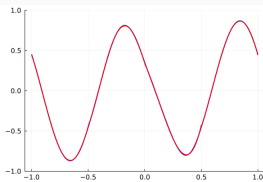


(c) p

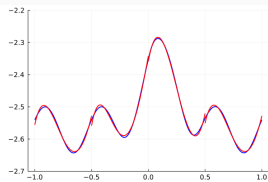
Primitive variables ρ, u, p and their entropy projection.



(d) v_1



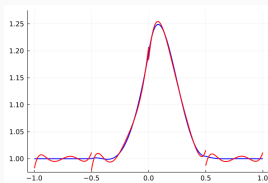
(e) v_2



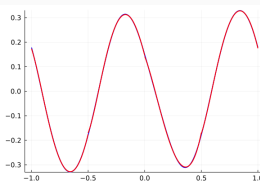
(f) v_3

Entropy variables and their L^2 projection.

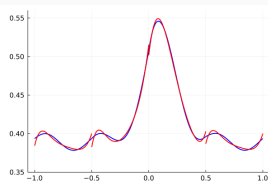
Illustration of the entropy projection



(a) ρ

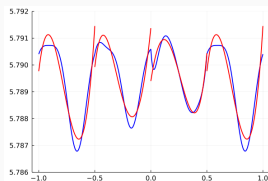


(b) u

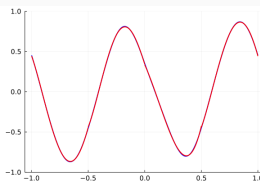


(c) p

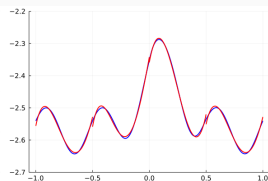
Primitive variables ρ , u , p and their entropy projection.



(d) v_1



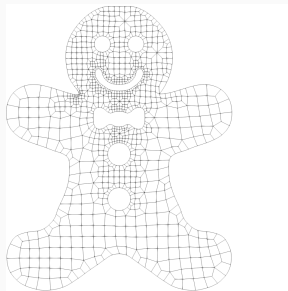
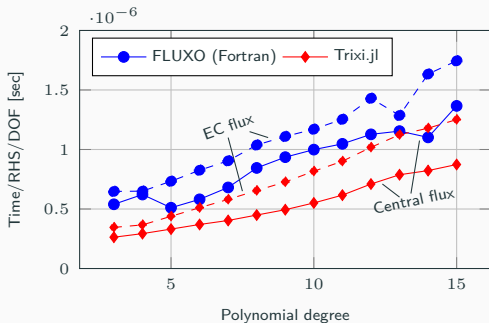
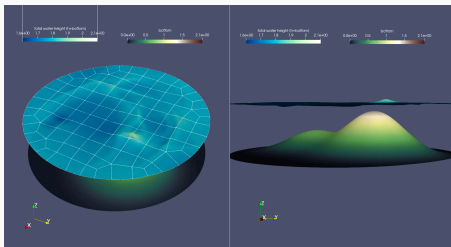
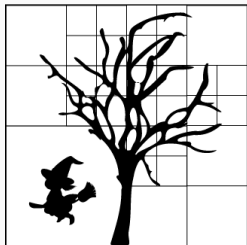
(e) v_2



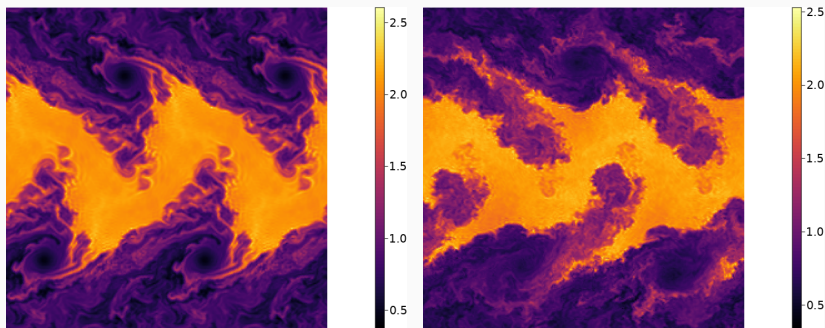
(f) v_3

Entropy variables and their L^2 projection.

This section uses the Julia library Trixi.jl, adaptive explicit RK



Differences in ESDG robustness for compressible Euler



(a) Degree $N = 3$ and a 64×64 mesh. **(b)** Degree $N = 7$ and a 32×32 mesh.

Density at time $T = 10$ for the Kelvin-Helmholtz instability using an entropy stable DG method with entropy projection.

Differences in ESDG robustness for compressible Euler

<div>Solver \ Degree</div>	1	2	3	4	5	6	7
Collocation	15	4.81	3.77	4.43	3.74	3.37	3.64
Entropy projection	15	15	15	15	15	15	15

$$N_{\text{cells}} = 16$$

<div>Solver \ Degree</div>	1	2	3	4	5	6	7
Collocation	15	4.12	3.65	4.27	3.54	3.66	3.56
Entropy projection	15	15	15	15	15	15	15

$$N_{\text{cells}} = 32$$

End times for the Kelvin-Helmholtz instability on **quadrilateral** meshes.
Blue indicates stable simulations, while red indicate crashes.

Differences in ESDG robustness for compressible Euler

<div>Solver \ Degree</div>	1	2	3	4	5	6
Collocation	15	3.98	3.44	2.99	2.94	3.13
Entropy projection	15	15	15	15	15	15

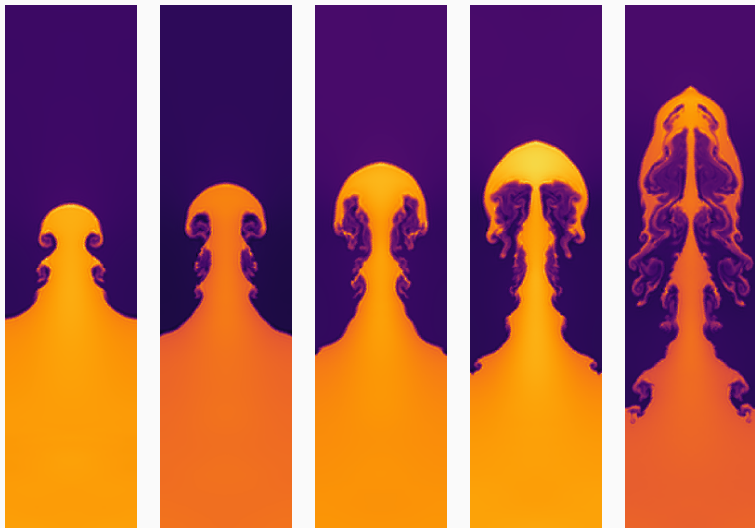
$$N_{\text{cells}} = 16$$

<div>Solver \ Degree</div>	1	2	3	4	5	6
Collocation	3.919	3.45	3.19	2.96	3.06	3.27
Entropy projection	15	15	15	15	15	15

$$N_{\text{cells}} = 32$$

End times for the Kelvin-Helmholtz instability on triangular meshes. Blue indicates stable simulations, while red indicate crashes.

Similar behavior for Rayleigh-Taylor, Richtmeyer-Meshkov



(a) $t = 1.25$

(b) $t = 1.5$

(c) $t = 1.75$

(d) $t = 2$

(e) $t = 2.5$

Rayleigh-Taylor instability: $N = 3$ entropy projection DG, 32×128 elements.

Similar behavior for Rayleigh-Taylor, Richtmeyer-Meshkov

Solver \ Degree	1	2	3	4	5	6	7
Collocation	3.67	3.4	3.33	3.26	3.11	3.03	3.04
Entropy projection	15	15	15	15	15	15	15

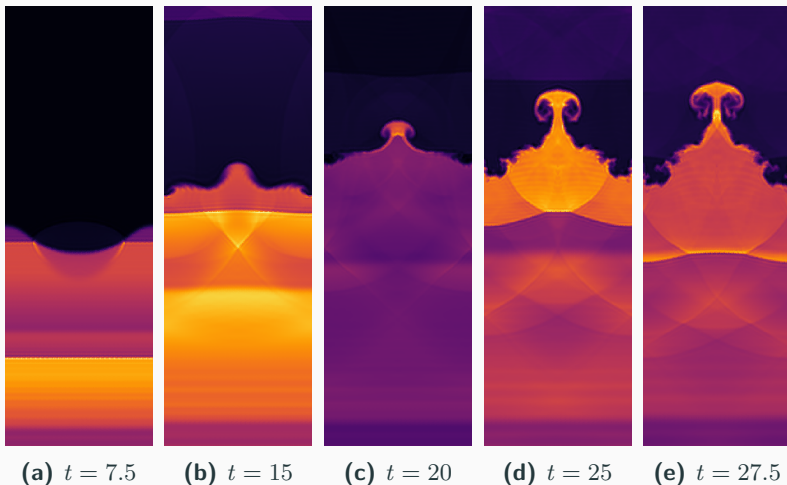
RTI, quadrilateral mesh, $N_{\text{cells}} = 16$

Solver \ Degree	1	2	3	4	5	6	7
Collocation	4.00	3.14	3.44	3.16	3.03	2.97	2.98
Entropy projection	15	15	15	15	15	15	15

RTI, quadrilateral mesh, $N_{\text{cells}} = 32$

End times for the Rayleigh-Taylor instability. Blue indicates stable simulations, while red indicate crashes.

Similar behavior for Rayleigh-Taylor, Richtmeyer-Meshkov



Richtmeyer-Meshkov instability using $N = 3$ entropy projection DG with 32×96 elements. Entropy projection is stable up to $T = 50$; entropy stable collocation crashes at $T \approx 20.1$.

Similar behavior for Rayleigh-Taylor, Richtmeyer-Meshkov

Solver \ Degree	1	2	3	4	5	6	7
Collocation	30	30	27.96	24.94	8.851	8.853	8.85
Entropy projection	30	30	30	30	30	30	30

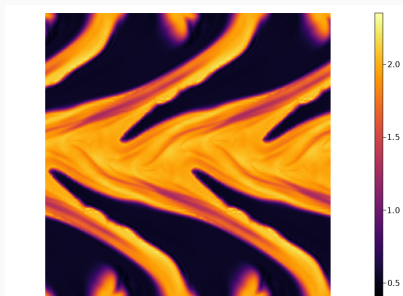
RMI, quadrilateral mesh, $N_{\text{cells}} = 16$

Solver \ Degree	1	2	3	4	5	6	7
Collocation	30	25.52	23.34	8.759	7.808	7.014	7.01
Entropy projection	30	30	30	30	30	30	30

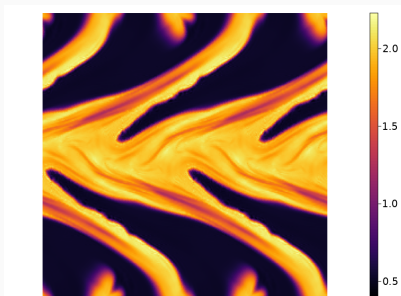
RMI, quadrilateral mesh, $N_{\text{cells}} = 32$

End times for the Richtmeyer-Meshkov instability. Blue indicates stable simulations, while red indicate crashes.

Similar behavior is observed for ideal magento-hydrodynamics



Density for entropy stable Gauss DG at time $T = 10$ (degree $N = 3$ and a 64×64 mesh).



Density for entropy stable Gauss DG at time $T = 10$ (degree $N = 7$ and a 32×32 mesh).

Solution snapshots for a weakly magnetized Kelvin-Helmholtz instability using an entropy stable Gauss DG scheme on uniform quadrilateral meshes.

Similar behavior is observed for ideal magento-hydrodynamics

Solver \ Degree	1	2	3	4	5	6	7
Collocation	15	15	11.50	10.99	10.32	10.23	10.27
Entropy projection	15	15	15	15	15	15	15

MHD KHI, quadrilateral mesh, $N_{\text{cells}} = 16$

Solver \ Degree	1	2	3	4	5	6	7
Collocation	15	11.64	11.05	11.11	11.48	10.17	10.92
Entropy projection	15	15	15	15	15	15	15

MHD KHI, quadrilateral mesh, $N_{\text{cells}} = 32$

End times for the magnetized Kelvin-Helmholtz instability. Blue indicates stable simulations, while red indicate crashes.

Similar behavior is observed for ideal magento-hydrodynamics

<div>Solver \ Degree</div>	1	2	3	4	5	6
Collocation	12.85	13.78	10.63	10.21	10.99	9.97
Entropy projection	15	15	15	15	15	15

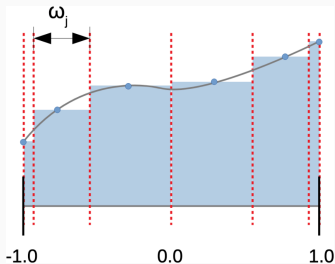
MHD KHI, triangular mesh, $N_{\text{cells}} = 16$

<div>Solver \ Degree</div>	1	2	3	4	5	6
Collocation	14.88	11.12	9.75	10.08	10.31	10.22
Entropy projection	15	15	15	15	15	15

MHD KHI, triangular mesh, $N_{\text{cells}} = 32$

End times for the magnetized Kelvin-Helmholtz instability. Blue indicates stable simulations, while red indicate crashes.

Why not just use shock capturing and positivity limiting?



Interpretation of Lobatto nodes as a sub-cell finite volume grid.

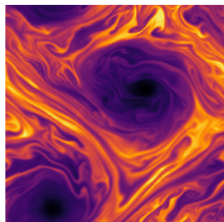
We compare entropy projection DG to two state-of-the-art schemes:

- DGSEM-SC-PP: **very light** entropy stable shock capturing + Zhang-Shu positivity limiting.
- DGSEM-subcell: positivity and shock capturing using **subcell limiting** (not entropy stable).

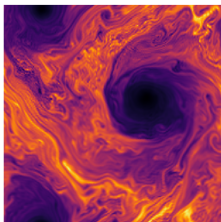
Hennemann, Ruéda-Ramírez, Hindenlang, Gassner (2021). A provably entropy stable subcell shock capturing approach for high order split form DG for the compressible Euler equations.

Ruéda-Ramírez, Pazner, Gassner (2022, preprint). Subcell limiting strategies for DGSEM.

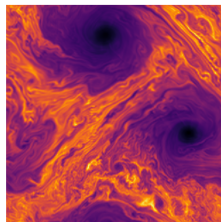
Application: under-resolved “turbulent” flows



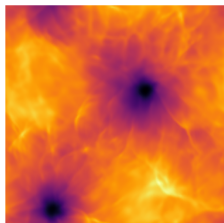
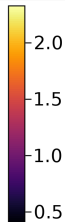
DGSEM-SC-PP density



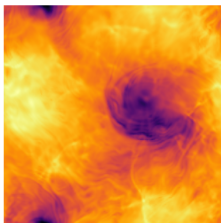
DGSEM-subcell density



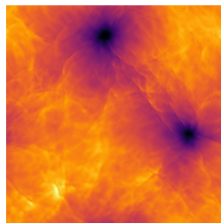
Gauss density



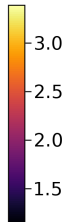
DGSEM-SC-PP pressure



DGSEM-subcell pressure

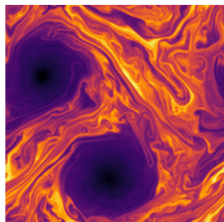


Gauss pressure

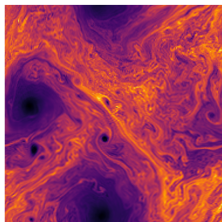


Kelvin-Helmholtz instability at $T_{\text{final}} = 25$ on a $N = 3, 64^2$ mesh.

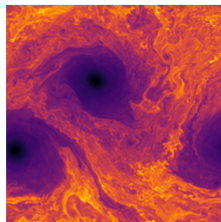
Application: under-resolved “turbulent” flows



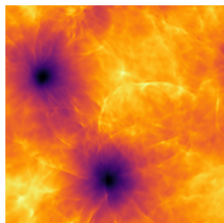
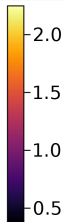
DGSEM-SC-PP density



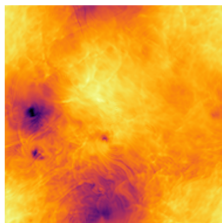
DGSEM-subcell density



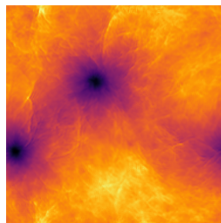
Gauss density



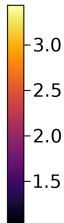
DGSEM-SC-PP pressure



DGSEM-subcell pressure

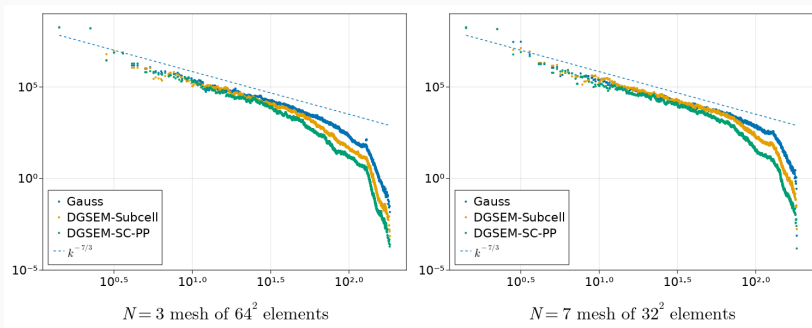


Gauss pressure



Kelvin-Helmholtz instability at $T_{\text{final}} = 25$ on a $N = 7, 32^2$ mesh.

Under-resolved “turbulence” is sensitive to extra dissipation



- Sample with $(N + 1) \times$ number of elements points (uniformly spaced to avoid element interfaces) along each dimension.
- Compute Fourier modes of velocity weighted by $\sqrt{\rho}$, sum energy over “effective wavenumbers” for a 1D power spectra.

Conclusion

- Positivity preserving limiters enable robust entropy stable nodal DG simulations of compressible flow.
- The “entropy projection” appears to improve robustness for under-resolved flows.

This work is supported by DMS-1943186.

Thank you! Questions?

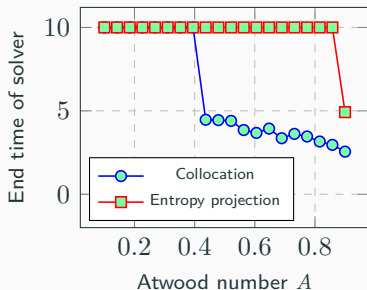


Chan, Ranocha, Rueda-Ramírez, Gassner, Warburton (2022). *On the entropy projection and the robustness of high order entropy stable discontinuous Galerkin schemes for under-resolved flows.*

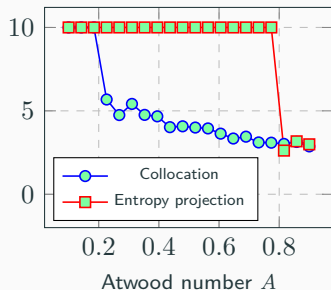
Lin, Chan, Tomas (2022). *A positivity preserving strategy for entropy stable discontinuous Galerkin discretizations of the compressible Euler and Navier-Stokes equations.*

Additional slides

Robustness depends on the Atwood number



(a) $N = 3$, 32×32 quad mesh



(b) $N = 7$, 16×16 quad mesh

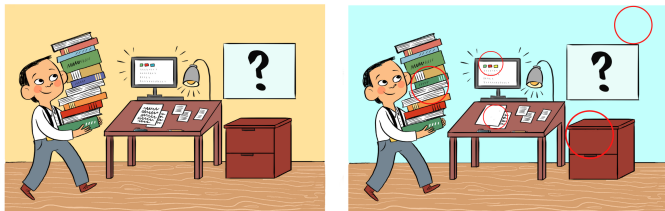
- Entropy stable collocation DG is robust when density is near-constant, but crashes at higher Atwood numbers

$$A = (\rho_2 - \rho_1)/(\rho_1 + \rho_2), \quad A \in [0, 1).$$

- Entropy projection is stable up to $A \approx .8$.

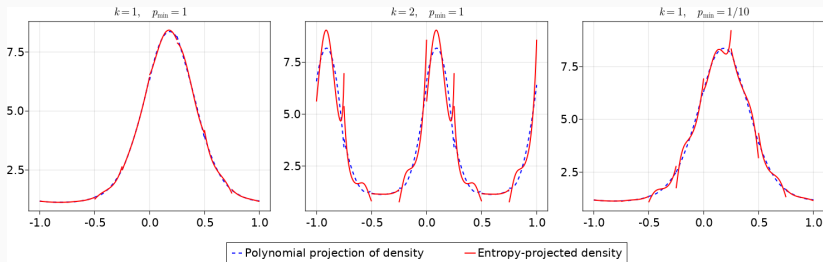
Why the difference in robustness?

CAN YOU SPOT ALL 5 DIFFERENCES BETWEEN
THESE TWO discretizations ?



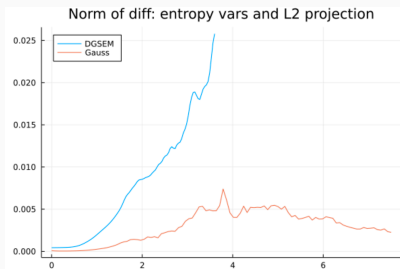
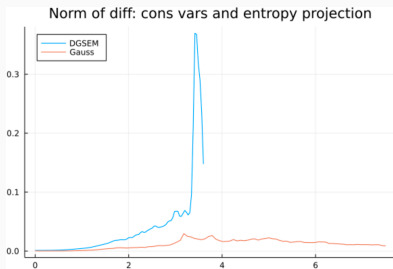
- Both are entropy stable, but Gauss collocation increases quadrature accuracy (reduces aliasing).
- Gauss introduces interface corrections and **entropy projection**.

Why would the entropy projection improve robustness?



Some clues: entropy projection uses L^2 projection of entropy variables, amplifies effects of **under-resolution** and **near-zero density or pressure**.

Evolution of differences between the conservative variables and entropy projected variables



Difference over time between the conservative and entropy projected variables $\|\tilde{\mathbf{u}} - \mathbf{u}\|_{L^2}$ for collocation and entropy projection schemes.

If $\tilde{\mathbf{u}} \approx \mathbf{u}$, the mapping between conservative and entropy variables is well-posed \implies the density and pressure are positive?

“Hybridization” for efficient interface coupling

- Hybridized SBP operators involve *both* volume/face nodes.

$$\mathbf{Q}_h = \frac{1}{2} \begin{bmatrix} \mathbf{Q} - \mathbf{Q}^T & \mathbf{E}^T \mathbf{B} \\ -\mathbf{B} \mathbf{E} & \mathbf{B} \end{bmatrix},$$

- Let $g(x)$ be a function. We can approximate $\frac{\partial g}{\partial x}$ via

$$\frac{\partial g}{\partial x} \approx \mathbf{M}^{-1} \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \mathbf{Q}_h \begin{bmatrix} g(\mathbf{x}_q) \\ g(\mathbf{x}_f) \end{bmatrix},$$

where $\mathbf{x}_q, \mathbf{x}_f$ are volume and face nodes, $\mathbf{V}_q, \mathbf{V}_f$ are **volume and face interpolation matrices**.

- Equivalent to adding error-reducing **correction terms** of the form “ $\mathbf{E}f(\mathbf{u}) - f(\mathbf{E}\mathbf{u})$ ”.

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1} + \mathbf{E}^T \mathbf{B} (\mathbf{f}^* - \mathbf{f}(\mathbf{u})) = 0.$$

- \mathbf{F} is the matrix of flux evaluations using solution values at *both* volume and face nodes + entropy projection:

$$\mathbf{F}_{ij} = \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad \tilde{\mathbf{u}} = \text{evaluate } \mathbf{u}(\Pi_N \mathbf{v}(\mathbf{u})).$$

- Entropy stable if $\mathbf{Q}_h \mathbf{1} = \mathbf{0}$ (true under weak conditions on quadrature accuracy).

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2 \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (\mathbf{Q}_h \circ \mathbf{F}) \mathbf{1} + \mathbf{V}_f^T \mathbf{B} (\mathbf{f}^* - \mathbf{f}(\mathbf{u})) = 0.$$

- \mathbf{F} is the matrix of flux evaluations using solution values at *both* volume and face nodes + entropy projection:

$$\mathbf{F}_{ij} = \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad \tilde{\mathbf{u}} = \text{evaluate } \mathbf{u}(\Pi_N \mathbf{v}(\mathbf{u})).$$

- Entropy stable if $\mathbf{Q}_h \mathbf{1} = \mathbf{0}$ (true under weak conditions on quadrature accuracy).

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2 \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (\mathbf{Q}_h \circ \mathbf{F}) \mathbf{1} + \mathbf{V}_f^T \mathbf{B} (\mathbf{f}^* - \mathbf{f}(\mathbf{u})) = 0.$$

- \mathbf{F} is the matrix of flux evaluations using solution values at *both* volume and face nodes + **entropy projection**:

$$\mathbf{F}_{ij} = \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad \tilde{\mathbf{u}} = \text{evaluate } \mathbf{u}(\Pi_N \mathbf{v}(\mathbf{u})).$$

- Entropy stable if $\mathbf{Q}_h \mathbf{1} = \mathbf{0}$ (true under weak conditions on quadrature accuracy).

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + 2 \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (\mathbf{Q}_h \circ \mathbf{F}) \mathbf{1} + \mathbf{V}_f^T \mathbf{B} (\mathbf{f}^* - \mathbf{f}(\mathbf{u})) = 0.$$

- \mathbf{F} is the matrix of flux evaluations using solution values at *both* volume and face nodes + **entropy projection**:

$$\mathbf{F}_{ij} = \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad \tilde{\mathbf{u}} = \text{evaluate } \mathbf{u}(\Pi_N \mathbf{v}(\mathbf{u})).$$

- Entropy stable if $\mathbf{Q}_h \mathbf{1} = \mathbf{0}$ (true under weak conditions on quadrature accuracy).

Estimated cost for DGSEM and Gauss

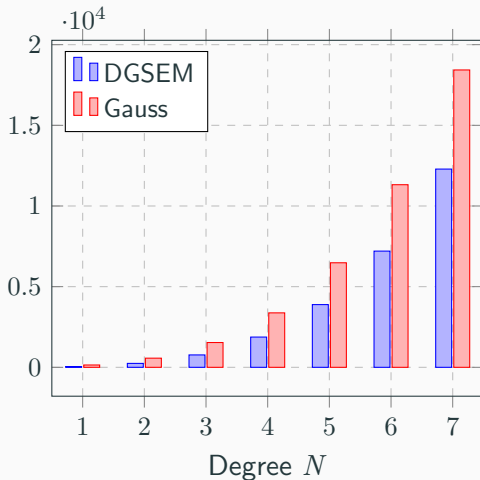
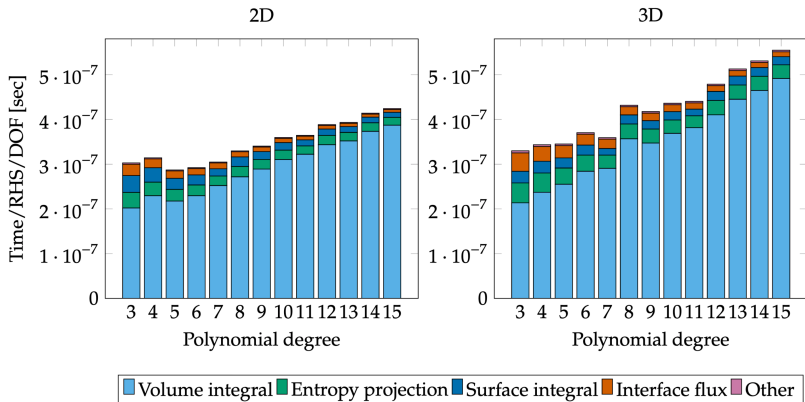


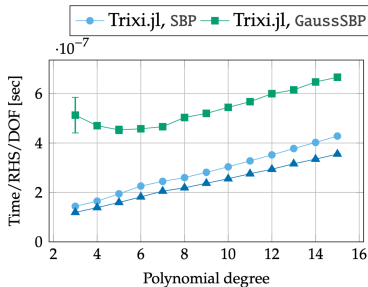
Figure 1: Comparison of 3D entropy stable DGSEM and entropy stable Gauss collocation in terms of two-point numerical flux evaluations.

Actual cost comparison for DGSEM and Gauss

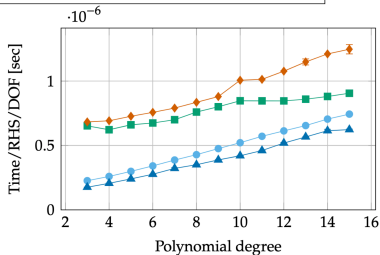


Performance index (PID) for entropy stable Gauss collocation.

Cost comparison of different implementations



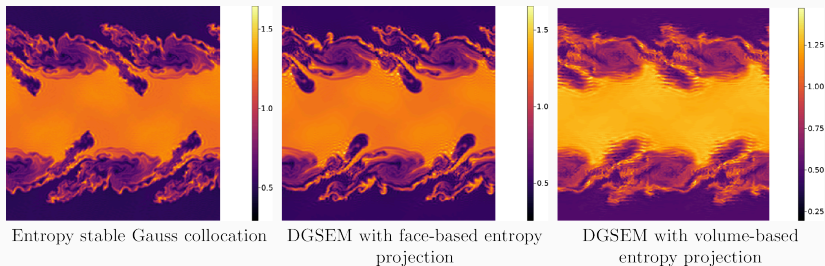
(a) 2D.



(b) 3D.

Runtime per RHS evaluation for different implementations of entropy stable DGSEM and Gauss collocation.

Does the entropy projection also help “bad” DG schemes?



Degree $N = 3$ and 64×64 grid Kelvin-Helmholtz simulations at $T = 5$. All methods run until $T = 25$, while DGSEM crashes at $T \approx 3.5$.

“Variant” schemes introduce entropy projection, but have similar or lower quadrature accuracy compared with DGSEM.

Improved robustness is not due to interface dissipation

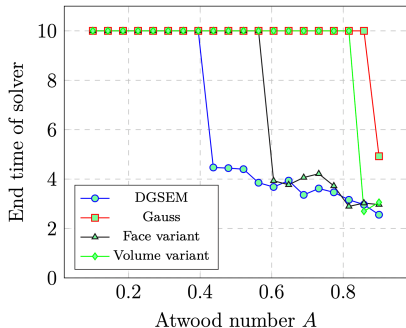
<div>Solver \ Degree</div>	1	2	3	4	5	6	7
Collocation	20	20	20	20	6.035	5.29	5.02
Entropy projection	20	20	20	20	20	20	20

$$N_{\text{cells}} = 8^3$$

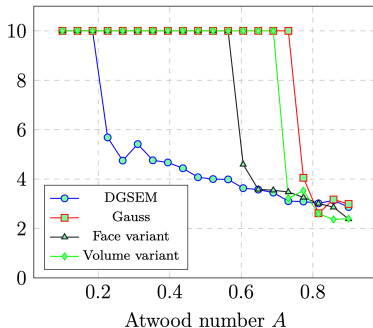
End times for entropy *conservative* simulations of the Taylor-Green vortex on hex meshes. Blue indicates stable simulations, while red indicate crashes.

We observe differences in robustness even for *entropy conservative* schemes (no entropy dissipation).

Improved robustness is not (only) due to quadrature accuracy



$N = 3, 32 \times 32$ quadrilateral mesh



$N = 7, 16 \times 16$ quadrilateral mesh

Entropy projection is not the only factor: “bad” entropy projection variant schemes improve robustness, but not as much.