Entropy stable high order discontinuous Galerkin methods for nonlinear conservation laws

Jesse Chan

Dept. of Computational and Applied Mathematics
Rice University

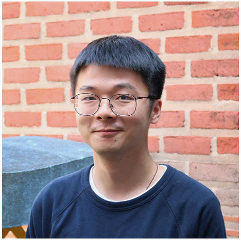
Australian National University
Computational Mathematics Seminar

Collaborators

Tim Warburton (VT)

Mario Bencomo (postdoc, adjoints)

Philip (Xinhui) Wu (GPU + shallow water)

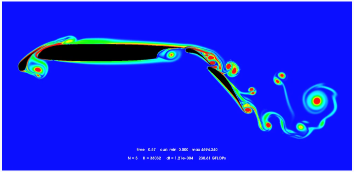


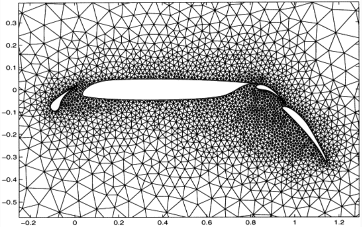
Yimin Lin (GPU + compressible flow)

Christina Taylor (implicit + ROMs?)

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.

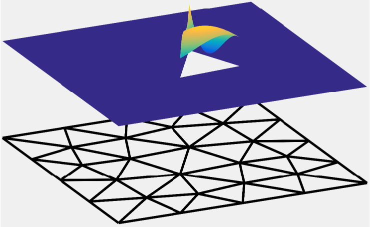




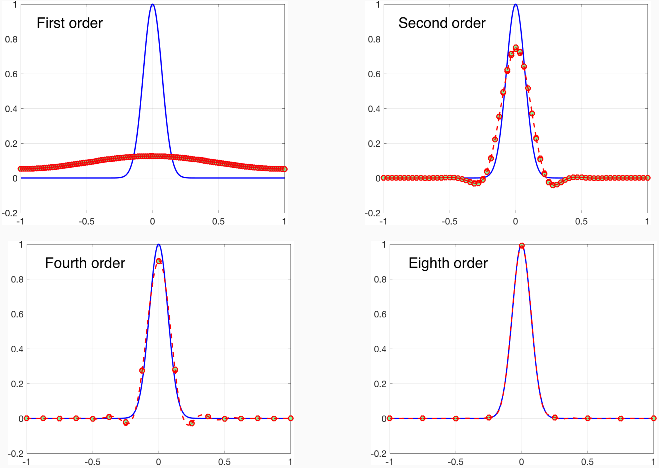
Mesh from Slawig 2001.

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.

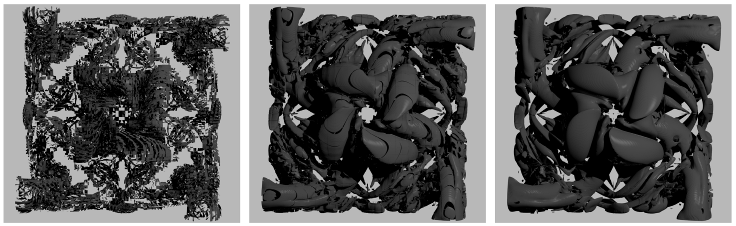


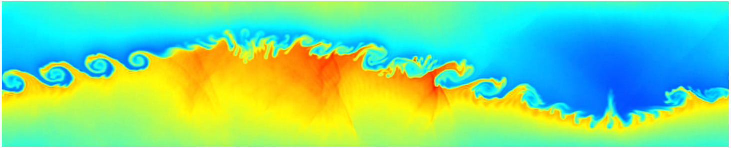
Why high order accuracy?



High order accurate resolution of propagating vortices and waves.

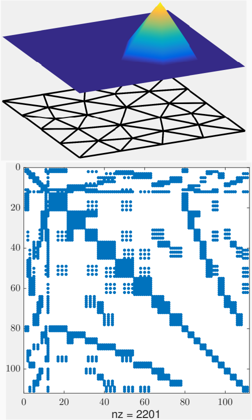
Why high order accuracy?



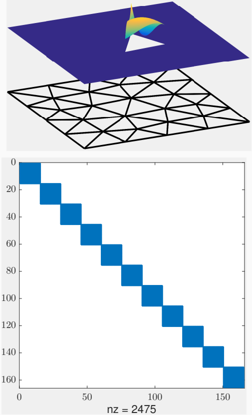


2nd, 4th, and 16th order Taylor-Green (top), 8th order Kelvin-Helmholtz (bottom). Vorticular structures and acoustic waves are both sensitive to numerical dissipation. Results from Beck and Gassner (2013) and Per-Olof Persson's website.

Why discontinuous Galerkin methods?



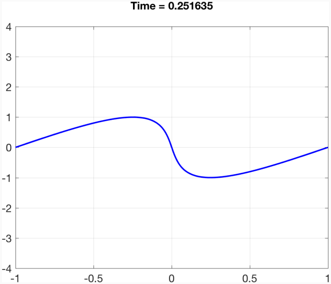
(a) High order FEM



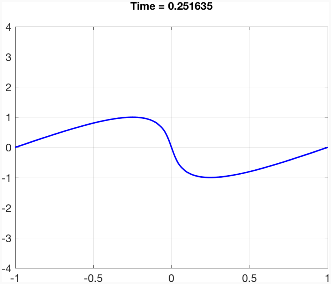
(b) High order DG

The DG mass matrix is easily invertible for explicit time-stepping.

Why *not* high order DG methods?



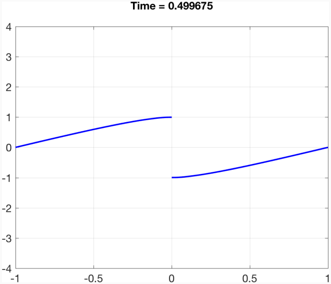
(a) Exact solution



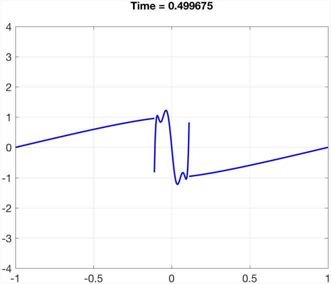
(b) 8th order DG

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

Why *not* high order DG methods?



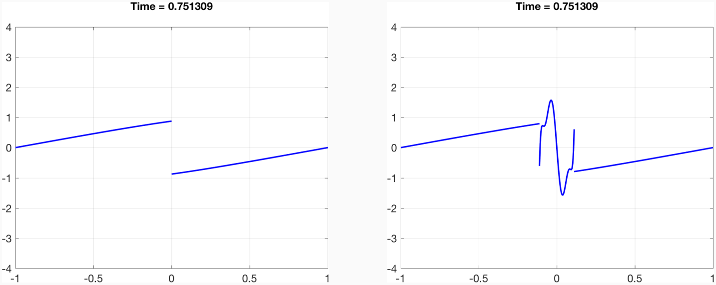
(a) Exact solution



(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?

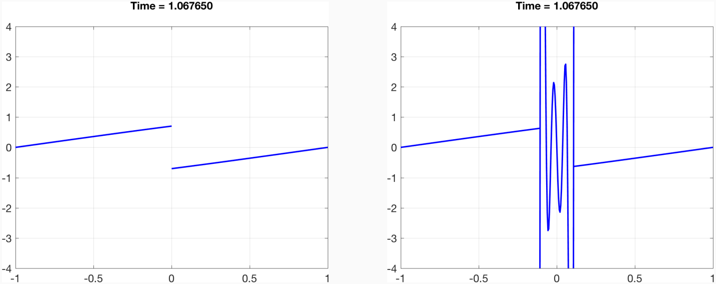


(a) Exact solution

(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?

In practice, high order schemes need solution regularization (e.g., artificial viscosity, filtering, slope limiting).

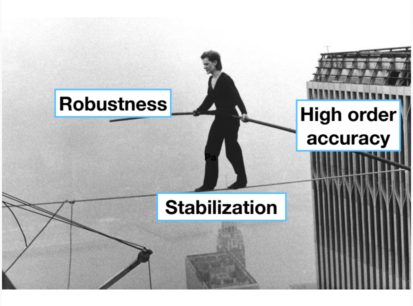
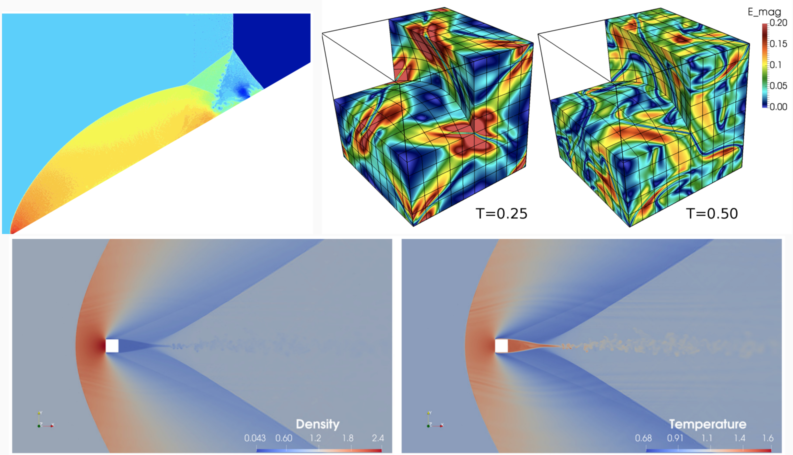


Image adapted from "Man On Wire" (2008)

- Goal: stability independent of solution regularization.
- Entropy stable schemes: improve robustness without reducing accuracy.

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 run without artificial viscosity, filtering, or slope limiting.

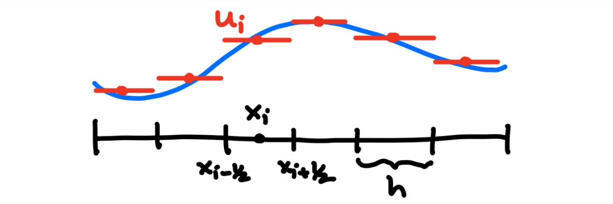
Chen, Shu (2017). *Entropy stable high order DG methods with suitable quadrature rules...*

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.*

Entropy conservative and entropy stable finite volume methods

Basics of finite volume methods



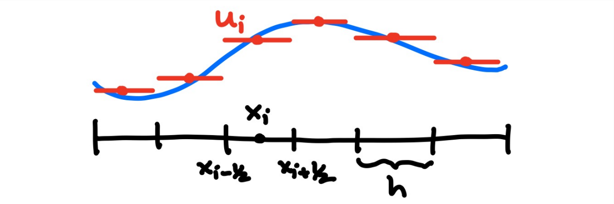
- Solve for $\mathbf{u}_i = \frac{1}{h} \int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{u}(x,t) \, \mathrm{d}x$.

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \mathbf{0}$$

- Replace $\boldsymbol{f}(\boldsymbol{u}(x_{i\pm 1/2}, t))$ with a *numerical flux*

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1}, \mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_i, \mathbf{u}_{i-1})}{h} = \mathbf{0}$$

Basics of finite volume methods



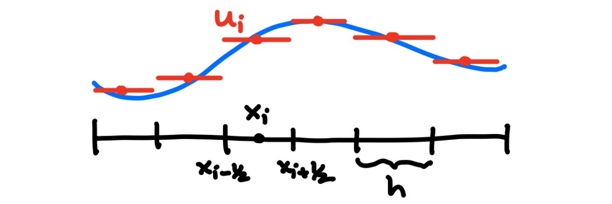
- Solve for $\mathbf{u}_i = \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{u}(x,t)\,\mathrm{d}x$.

$$\frac{\partial}{\partial t}\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}(x_{i+1/2},t)) - \boldsymbol{f}(\boldsymbol{u}(x_{i-1/2},t)) = \mathbf{0}$$

- Replace $\boldsymbol{f}(\boldsymbol{u}(x_{i\pm1/2},t))$ with a *numerical flux*

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1},\mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_i,\mathbf{u}_{i-1})}{h} = \mathbf{0}$$

Basics of finite volume methods



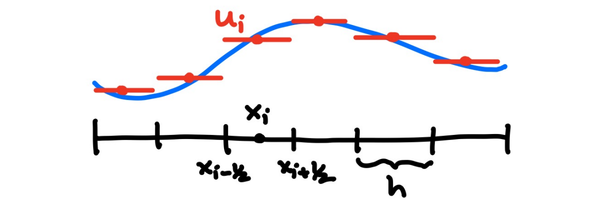
- Solve for $\mathbf{u}_i = \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{u}(x,t)\,\mathrm{d}x$.

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{dt}} + \frac{\boldsymbol{f}(\boldsymbol{u}(x_{i+1/2},t)) - \boldsymbol{f}(\boldsymbol{u}(x_{i-1/2},t))}{h} = \mathbf{0}$$

- Replace $\boldsymbol{f}(\boldsymbol{u}(x_{i\pm 1/2},t))$ with a *numerical flux*

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{dt}} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1},\mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_i,\mathbf{u}_{i-1})}{h} = \mathbf{0}$$

Basics of finite volume methods



- Solve for $\mathbf{u}_i = \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{u}(x,t)\,\mathrm{d}x$.

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \frac{\boldsymbol{f}(\boldsymbol{u}(x_{i+1/2},t)) - \boldsymbol{f}(\boldsymbol{u}(x_{i-1/2},t))}{h} = \mathbf{0}$$

- Replace $\boldsymbol{f}(\boldsymbol{u}(x_{i\pm1/2},t))$ with a *numerical flux*

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1},\mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_i,\mathbf{u}_{i-1})}{h} = \mathbf{0}$$

Entropy stability for nonlinear problems

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

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

- Continuous entropy inequality: convex entropy function $S(\boldsymbol{u})$, "entropy potential" $\psi(\boldsymbol{u})$, entropy variables $\boldsymbol{v}(\boldsymbol{u})$

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

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

Entropy conservative finite volume methods

- Finite volume scheme:

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{dt}} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1}, \mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_{i+1}, \mathbf{u}_i)}{h} = \mathbf{0}.$$

- Take $\boldsymbol{f}_S$ to be an entropy conservative numerical flux

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

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

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

- Can show numerical scheme conserves entropy

$$\int_\Omega \frac{\partial S(\boldsymbol{u})}{\partial t} \approx \sum_i h \frac{\mathrm{d}S(\mathbf{u}_i)}{\mathrm{dt}} = 0.$$

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

Entropy stable finite volume methods

- Finite volume scheme with dissipation $\mathbf{d}(\mathbf{u})$:

$$\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{dt}} + \frac{\boldsymbol{f}_S(\mathbf{u}_{i+1}, \mathbf{u}_i) - \boldsymbol{f}_S(\mathbf{u}_{i+1}, \mathbf{u}_i)}{h} = \mathbf{d}(\mathbf{u}).$$

- Take $\boldsymbol{f}_S$ to be an entropy conservative numerical flux

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

- Can show numerical scheme dissipates entropy

$$\int_\Omega \frac{\partial S(\boldsymbol{u})}{\partial t} \approx \sum_i h \frac{\mathrm{d}S(\mathbf{u}_i)}{\mathrm{dt}} = \mathbf{v}^T \mathbf{d}(\mathbf{u}) \overset{?}{\leq} 0.$$

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

Example of EC fluxes (compressible Euler equations)

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

$$f_S^1(\boldsymbol{u}_L, \boldsymbol{u}_R) = \{\{\rho\}\}^{\log} \{\{u\}\}$$
$$f_S^2(\boldsymbol{u}_L, \boldsymbol{u}_R) = \{\{u\}\} f_S^1 + p_{\text{avg}}$$
$$f_S^3(\boldsymbol{u}_L, \boldsymbol{u}_R) = (E_{\text{avg}} + p_{\text{avg}}) \{\{u\}\},$$

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

- Non-standard logarithmic mean, "inverse temperature" β

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

Chandreshekar (2013), *Kinetic energy preserving and entropy stable finite volume schemes for the compressible Euler and Navier-Stokes equations.*

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} + \frac{1}{h} \begin{bmatrix} \boldsymbol{f}_S(\mathbf{u}_1, \mathbf{u}_2) - \boldsymbol{f}_S(\mathbf{u}_N, \mathbf{u}_1) \\ \boldsymbol{f}_S(\mathbf{u}_2, \mathbf{u}_3) - \boldsymbol{f}_S(\mathbf{u}_1, \mathbf{u}_2) \\ \vdots \\ \boldsymbol{f}_S(\mathbf{u}_N, \mathbf{u}_1) - \boldsymbol{f}_S(\mathbf{u}_{N-1}, \mathbf{u}_N) \end{bmatrix} = \mathbf{0}.$$

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} + \frac{1}{h} \begin{bmatrix} \boldsymbol{f}_S(\mathbf{u}_1, \mathbf{u}_2) - \boldsymbol{f}_S(\mathbf{u}_N, \mathbf{u}_1) \\ \boldsymbol{f}_S(\mathbf{u}_2, \mathbf{u}_3) - \boldsymbol{f}_S(\mathbf{u}_1, \mathbf{u}_2) \\ \vdots \\ \boldsymbol{f}_S(\mathbf{u}_N, \mathbf{u}_1) - \boldsymbol{f}_S(\mathbf{u}_{N-1}, \mathbf{u}_N) \end{bmatrix} = \mathbf{0}.$$

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$h\frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{1,2} - \mathbf{F}_{1,N} \\ \mathbf{F}_{2,3} - \mathbf{F}_{2,1} \\ \vdots \\ \mathbf{F}_{N,1} - \mathbf{F}_{N,N-1} \end{bmatrix} = \mathbf{0}, \qquad \mathbf{F}_{ij} = \boldsymbol{f}_S(\mathbf{u}_i, \mathbf{u}_j).$$

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$\begin{bmatrix} \mathbf{F}_{1,2} - \mathbf{F}_{1,N} \\ \mathbf{F}_{2,3} - \mathbf{F}_{2,1} \\ \vdots \\ \mathbf{F}_{N,1} - \mathbf{F}_{N,N-1} \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 & & -1 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & 1 \\ 1 & & -1 & 0 \end{bmatrix} \circ \begin{bmatrix} \mathbf{F}_{1,1} & \dots & \mathbf{F}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{N,1} & \dots & \mathbf{F}_{N,N} \end{bmatrix} \right) \mathbf{1}.$$

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$\begin{bmatrix} \mathbf{F}_{1,2} - \mathbf{F}_{1,N} \\ \mathbf{F}_{2,3} - \mathbf{F}_{2,1} \\ \vdots \\ \mathbf{F}_{N,1} - \mathbf{F}_{N,N-1} \end{bmatrix} = \left(\underbrace{\begin{bmatrix} 0 & 1 & & -1 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & 1 \\ 1 & & -1 & 0 \end{bmatrix}}_{2\mathbf{Q}} \circ \underbrace{\begin{bmatrix} \mathbf{F}_{1,1} & \dots & \mathbf{F}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{N,1} & \dots & \mathbf{F}_{N,N} \end{bmatrix}}_{\mathbf{F}} \right) \mathbf{1}.$$

Matrix reformulation using Hadamard products

Hadamard product of two matrices $\mathbf{A} \circ \mathbf{B}$

$$\begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \dots & \mathbf{A}_{nn} \end{bmatrix} \circ \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \dots & \mathbf{B}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} & \dots & \mathbf{A}_{1n}\mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}\mathbf{B}_{n1} & \dots & \mathbf{A}_{nn}\mathbf{B}_{nn} \end{bmatrix}.$$

Rewrite an N-point (periodic) finite volume scheme as

$$\begin{bmatrix} \mathbf{F}_{1,2} - \mathbf{F}_{1,N} \\ \mathbf{F}_{2,3} - \mathbf{F}_{2,1} \\ \vdots \\ \mathbf{F}_{N,1} - \mathbf{F}_{N,N-1} \end{bmatrix} = 2(\mathbf{Q} \circ \mathbf{F})\mathbf{1}.$$

Interpretation using finite difference matrices

Let $\mathbf{M} = h\mathbf{I}$. Can reformulate entropy conservative finite volumes as

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + 2\left(\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} = \mathbf{0}, \qquad \mathbf{Q} = \frac{1}{2}\begin{bmatrix} 0 & 1 & & -1 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & 1 \\ 1 & & -1 & 0 \end{bmatrix}$$

Note: $\mathbf{M}^{-1}\mathbf{Q}$ is a 2nd order (periodic) differentiation matrix.

Key observation: generalizable beyond finite volumes

Entropy conservation for any $\underbrace{\mathbf{Q} = -\mathbf{Q}^T}_{\text{skew-symmetry}}$ and $\underbrace{\mathbf{Q1} = \mathbf{0}}_{\text{conservative}}$!

Interpretation using finite difference matrices

Let $\mathbf{M} = h\mathbf{I}$. Can reformulate entropy conservative finite volumes as

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + 2\left(\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} = \mathbf{0}, \qquad \mathbf{Q} = \frac{1}{2}\begin{bmatrix} 0 & 1 & & -1 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & 1 \\ 1 & & -1 & 0 \end{bmatrix}$$

Note: $\mathbf{M}^{-1}\mathbf{Q}$ is a 2nd order (periodic) differentiation matrix.

Key observation: generalizable beyond finite volumes

Entropy conservation for any $\underbrace{\mathbf{Q} = -\mathbf{Q}^T}_{\text{skew-symmetry}}$ and $\underbrace{\mathbf{Q}\mathbf{1} = \mathbf{0}}_{\text{conservative}}$!

Boundary conditions and summation-by-parts (SBP) property

Boundary conditions: choose appropriate "ghost" values $\mathbf{u}_1^+, \mathbf{u}_N^+$

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}} + 2\left(\mathbf{Q}\circ\mathbf{F}\right)\mathbf{1} + \mathbf{B}\begin{bmatrix} \boldsymbol{f}_S(\mathbf{u}_1^+, \mathbf{u}_1) - \boldsymbol{f}(\mathbf{u}_1) \\ \mathbf{0} \\ \boldsymbol{f}_S(\mathbf{u}_N^+, \mathbf{u}_N) - \boldsymbol{f}(\mathbf{u}_N) \end{bmatrix} = \mathbf{0}.$$

Entropy stable if $\mathbf{Q}$ satisfies a summation-by-parts (SBP) property

$$\mathbf{Q} = \frac{1}{2}\begin{bmatrix} -1 & 1 & & 0 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & 1 \\ & & -1 & 1 \end{bmatrix}, \qquad \mathbf{Q}+\mathbf{Q}^T = \mathbf{B} = \begin{bmatrix} -1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

Main innovation: fully algebraic proof of entropy stability

- Discrete analogue of the entropy identity

$$\boxed{\int_{-1}^{1} \boldsymbol{v}^T \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \mathbf{v}^T \boldsymbol{f}(\mathbf{u}) - \psi(\mathbf{u})\Big|_{-1}^{1}} \iff \boxed{\begin{aligned} &\mathbf{v}^T \left(2\mathbf{Q} \circ \mathbf{F}\right) \mathbf{1} \\ &= \mathbf{1}^T \mathbf{B} \left(\mathbf{v}^T \boldsymbol{f}(\mathbf{u}) - \psi(\mathbf{u})\right) \end{aligned}}$$

- Key step in proof: use entropy conservative property of flux on skew-symmetric form of $\mathbf{v}^T \left(2\mathbf{Q} \circ \mathbf{F}\right) \mathbf{1}$

$$\mathbf{v}^T \left(\left(\mathbf{Q} - \mathbf{Q}^T\right) \circ \mathbf{F}\right) \mathbf{1} = \sum_{ij} \mathbf{Q}_{ij} \underbrace{\left(\mathbf{v}_i - \mathbf{v}_j\right)^T \boldsymbol{f}_S\left(\mathbf{u}_i, \mathbf{u}_j\right)}_{\left(\psi(\mathbf{u}_i) - \psi(\mathbf{u}_j)\right)}$$

Tadmor (1987), Carpenter et al. (2014), Gassner, Winters, and Kopriva (2016).

Main innovation: fully algebraic proof of entropy stability

- Discrete analogue of the entropy identity

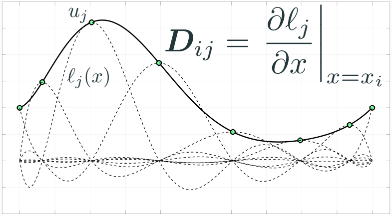
$$\boxed{\int_{-1}^{1} \boldsymbol{v}^T \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \mathbf{v}^T \boldsymbol{f}(\mathbf{u}) - \psi(\mathbf{u})\Big|_{-1}^{1}} \iff \boxed{\begin{aligned}&\mathbf{v}^T \left(2\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} \\ &= \mathbf{1}^T \mathbf{B} \left(\mathbf{v}^T \boldsymbol{f}(\mathbf{u}) - \psi(\mathbf{u})\right)\end{aligned}}$$

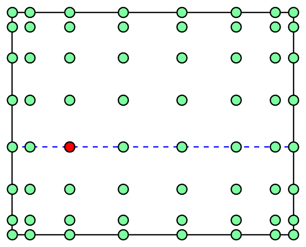
- Key step in proof: use entropy conservative property of flux on skew-symmetric form of $\mathbf{v}^T \left(2\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1}$

$$\mathbf{v}^T \left(\left(\mathbf{Q} - \mathbf{Q}^T\right) \circ \mathbf{F}\right)\mathbf{1} = \sum_{ij} \mathbf{Q}_{ij} \underbrace{\left(\mathbf{v}_i - \mathbf{v}_j\right)^T \boldsymbol{f}_S\left(\mathbf{u}_i, \mathbf{u}_j\right)}_{\left(\psi(\mathbf{u}_i) - \psi(\mathbf{u}_j)\right)}$$

Tadmor (1987), Carpenter et al. (2014), Gassner, Winters, and Kopriva (2016).

Extension to high order summation by parts (SBP) schemes





- Nodal differentiation matrix $\mathbf{D}$ has zero row sums

$$\sum_j \mathbf{D}_{ij} = 0 \quad \Longrightarrow \quad \mathbf{D1} = \mathbf{0}.$$

- Lobatto quadrature nodes recover summation-by-parts (SBP) property! Let $\mathbf{M}$ = lumped diagonal mass matrix:

$$\mathbf{Q} = \mathbf{MD}, \qquad \boxed{\mathbf{Q} + \mathbf{Q}^T = \mathbf{B}}.$$

Extension to high order DG methods (e.g., multiple elements)

- If $\mathbf{Q}$ is conservative $(\mathbf{Q1} = \mathbf{0})$ and satisfies summation-by-parts (SBP) property, then DG formulation is entropy *conservative*

$$\boxed{\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + 2\left(\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} + \mathbf{B}\Big(\underbrace{\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right)}_{\text{interface flux}} - \boldsymbol{f}(\mathbf{u})\Big) = \mathbf{0}}$$

- Generalizes to arbitrarily high polynomial degree N.
- Adding interface dissipation (e.g., Lax-Friedrichs) yields an entropy stable DG scheme.

$$\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) \rightarrow \boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) - \frac{\lambda}{2}[\![\mathbf{u}]\!]\boldsymbol{n}, \qquad \lambda > 0.$$

Fisher and Carpenter (2014), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), etc.

Extension to high order DG methods (e.g., multiple elements)

- If $\mathbf{Q}$ is conservative $(\mathbf{Q1} = \mathbf{0})$ and satisfies summation-by-parts (SBP) property, then DG formulation is entropy *conservative*

$$\boxed{\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}} + 2\left(\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} + \mathbf{B}\Big(\underbrace{\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right)}_{\text{interface flux}} - \boldsymbol{f}(\mathbf{u})\Big) = \mathbf{0}}$$

- Generalizes to arbitrarily high polynomial degree N.
- Adding interface dissipation (e.g., Lax-Friedrichs) yields an entropy stable DG scheme.

$$\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) \rightarrow \boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) - \frac{\lambda}{2}[\![\mathbf{u}]\!]\boldsymbol{n}, \qquad \lambda > 0.$$

Fisher and Carpenter (2014), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), etc.

Extension to high order DG methods (e.g., multiple elements)

- If $\mathbf{Q}$ is conservative $(\mathbf{Q1} = \mathbf{0})$ and satisfies summation-by-parts (SBP) property, then DG formulation is entropy *conservative*

$$\boxed{\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}} + 2\left(\mathbf{Q} \circ \mathbf{F}\right)\mathbf{1} + \mathbf{B}\Big(\underbrace{\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right)}_{\text{interface flux}} - \boldsymbol{f}(\mathbf{u})\Big) = \mathbf{0}}$$

- Generalizes to arbitrarily high polynomial degree N.
- Adding interface dissipation (e.g., Lax-Friedrichs) yields an entropy stable DG scheme.

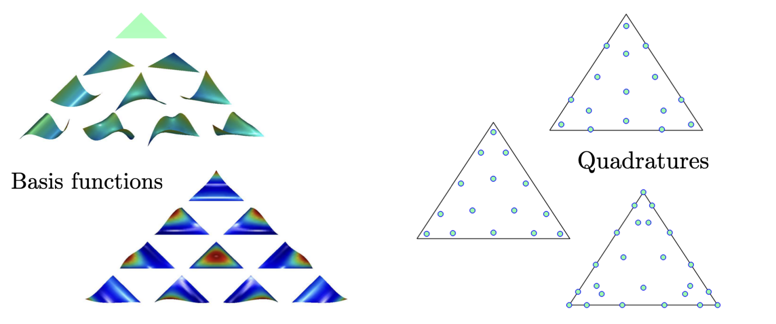
$$\boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) \rightarrow \boldsymbol{f}_S\left(\mathbf{u}^+, \mathbf{u}\right) - \frac{\lambda}{2}[\![\mathbf{u}]\!]\boldsymbol{n}, \qquad \lambda > 0.$$

Fisher and Carpenter (2014), Gassner, Winters, and Kopriva (2016), Chen and Shu (2017), etc.

Entropy stable modal discontinuous Galerkin formulations

Why "modal" formulations?

Nodal formulations: tied to specific nodes and basis.
"Modal" formulations: arbitrary basis functions and quadrature.

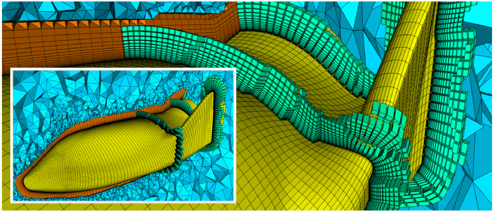


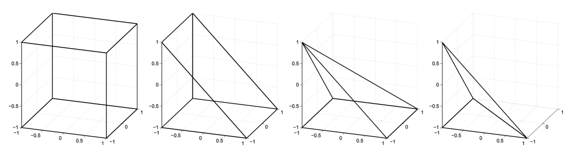
Enables use of standard tools in finite elements.

Figures from http://www2.compute.dtu.dk/ apek/DGFEMCourse2009/Lecture05.pdf.

Why “modal” formulations?

Nodal formulations: tied to specific nodes and basis.
“Modal” formulations: arbitrary basis functions and quadrature.





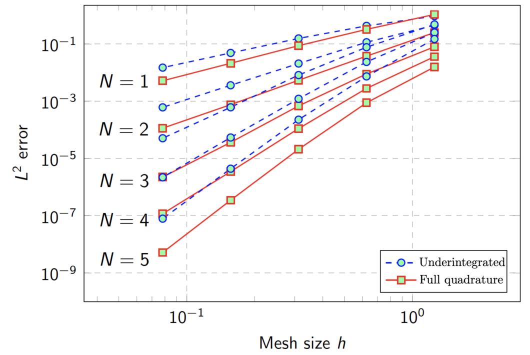
Applicable to multiple reference elements and hybrid meshes.

https://www.pointwise.com/news/2014/Hybrid-Hex-Tet-Meshing-Latest-Pointwise-Release.html.

Why "modal" formulations?

Nodal formulations: tied to specific nodes and basis.

"Modal" formulations: arbitrary basis functions and quadrature.

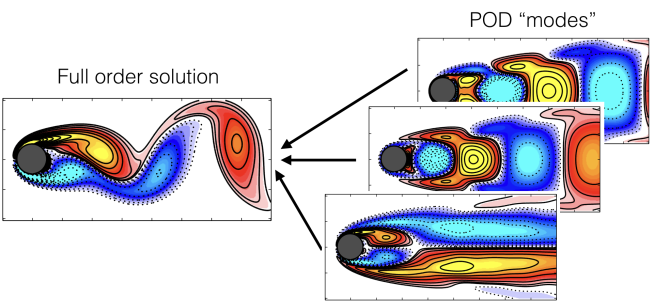


Can avoid *underintegration errors* for nonlinear terms + curved elements.

Wu, Chan, Kubatko (2020). *High-order ESDG methods for the shallow water equations . . .*

Why “modal” formulations?

Nodal formulations: tied to specific nodes and basis.
“Modal” formulations: arbitrary basis functions and quadrature.



Projection-based reduced order models: learn basis functions from data.

Figure adapted from Brunton, Proctor, Kutz (2016), *Discovering governing equations from data* ....

Challenge 1 for modal formulations: entropy projection

- Test functions must be polynomial. Entropy variables are not.

- If $\boldsymbol{u}_N$ is polynomial, testing with L^2 projection of entropy variables $\Pi_N \boldsymbol{v}(\boldsymbol{u}_N)$ recovers rate of change of entropy

$$\int_{D^k} \Pi_N \boldsymbol{v}(\boldsymbol{u}_N)^T \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \underbrace{\boldsymbol{v}(\boldsymbol{u}_N)^T}_{\frac{\partial S(\boldsymbol{u})}{\partial \boldsymbol{u}}} \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \frac{\partial S(\boldsymbol{u}_N)}{\partial t}$$

- For consistency, must also evaluate fluxes using projected entropy variables $\widetilde{\boldsymbol{u}} = \boldsymbol{u}(\Pi_N \boldsymbol{v}(\boldsymbol{u}_N))$.

$$(\mathbf{v}_i - \mathbf{v}_j)^T \boldsymbol{f}_S(\mathbf{u}_i, \mathbf{u}_j) \neq \psi(\mathbf{u}_i) - \psi(\mathbf{u}_j) \quad \text{if } \mathbf{v}_i \neq \mathbf{v}(\mathbf{u}_i).$$

Challenge 1 for modal formulations: entropy projection

- Test functions must be polynomial. Entropy variables are not.

- If $\boldsymbol{u}_N$ is polynomial, testing with L^2 projection of entropy variables $\Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)$ recovers rate of change of entropy

$$\int_{D^k} \Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)^T \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \underbrace{\boldsymbol{v}\left(\boldsymbol{u}_N\right)^T}_{\frac{\partial S(\boldsymbol{u})}{\partial \boldsymbol{u}}} \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \frac{\partial S(\boldsymbol{u}_N)}{\partial t}$$

- For consistency, must also evaluate fluxes using projected entropy variables $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(\Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)\right)$.

$$\left(\mathbf{v}_i - \mathbf{v}_j\right)^T \boldsymbol{f}_S\left(\mathbf{u}_i, \mathbf{u}_j\right) \neq \psi(\mathbf{u}_i) - \psi(\mathbf{u}_j) \quad \text{if } \mathbf{v}_i \neq \mathbf{v}\left(\mathbf{u}_i\right).$$

Challenge 1 for modal formulations: entropy projection

- Test functions must be polynomial. Entropy variables are not.

- If $\boldsymbol{u}_N$ is polynomial, testing with L^2 projection of entropy variables $\Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)$ recovers rate of change of entropy

$$\int_{D^k} \Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)^T \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \underbrace{\boldsymbol{v}\left(\boldsymbol{u}_N\right)^T}_{\frac{\partial S(\boldsymbol{u})}{\partial \boldsymbol{u}}} \frac{\partial \boldsymbol{u}_N}{\partial t} = \int_{D^k} \frac{\partial S(\boldsymbol{u}_N)}{\partial t}$$

- For consistency, must also evaluate fluxes using projected entropy variables $\widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(\Pi_N \boldsymbol{v}\left(\boldsymbol{u}_N\right)\right)$.

$$\left(\mathbf{v}_i - \mathbf{v}_j\right)^T \boldsymbol{f}_S\left(\mathbf{u}_i, \mathbf{u}_j\right) \neq \psi(\mathbf{u}_i) - \psi(\mathbf{u}_j) \quad \text{if } \mathbf{v}_i \neq \mathbf{v}\left(\mathbf{u}_i\right).$$

Illustration of entropy projection: degree $N = 4$, 8 elements

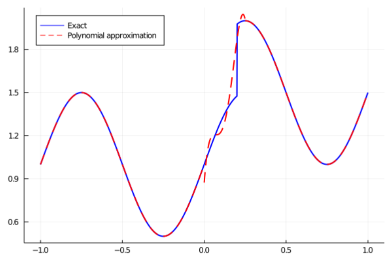
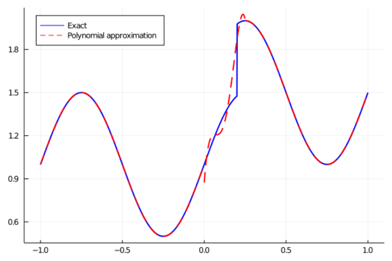


Illustration of entropy projection: degree $N = 4$, 8 elements



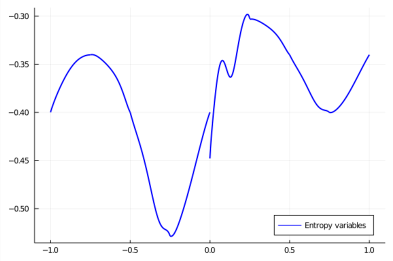


Illustration of entropy projection: degree $N = 4$, 8 elements

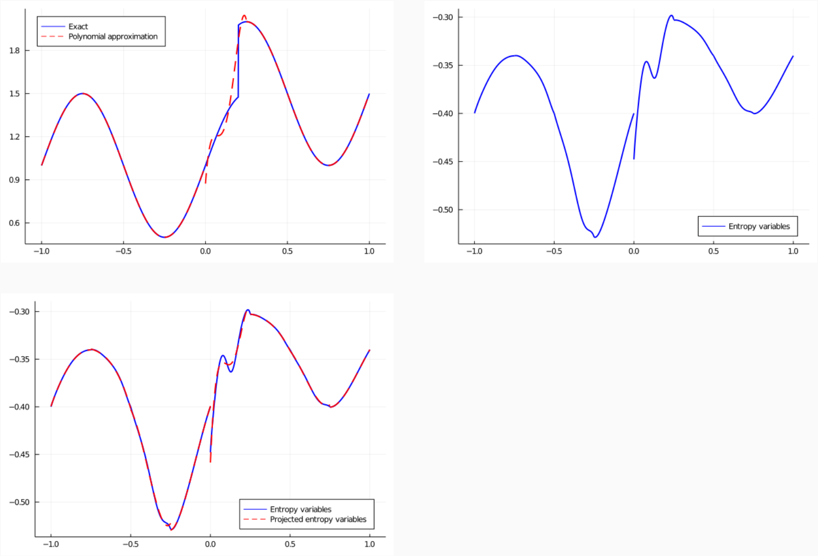
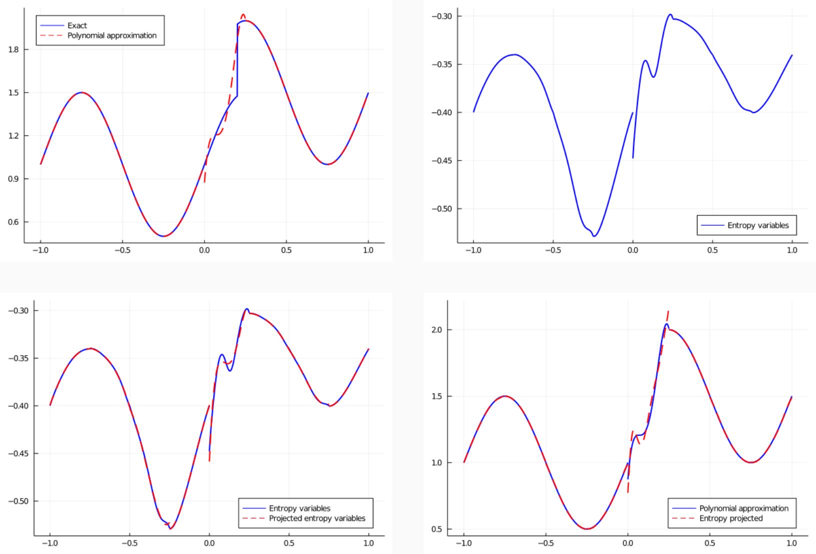
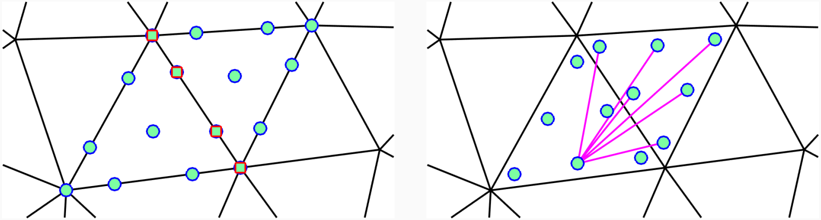


Illustration of entropy projection: degree $N = 4$, 8 elements



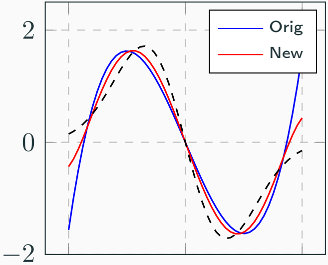
Challenge 2 for modal formulations: interface coupling



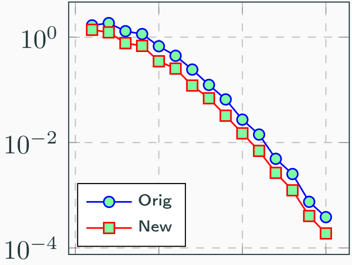
Entropy stable interface coupling with/without boundary nodes

- Interface fluxes must be designed to cancel other boundary terms in the discrete entropy balance.
- Entropy stable interface fluxes previously involved all-to-all coupling between nodes on different elements.

Efficient interface fluxes via "hybridization"



(a) Approximated derivatives



(b) L^2 error, degree $N = 1, \ldots, 15$

- Use an expanded *hybridized* SBP operator, where $\mathbf{E}$ is a face extrapolation matrix.

$$\mathbf{Q}_h = \frac{1}{2}\begin{bmatrix} \mathbf{Q} - \mathbf{Q}^T & \mathbf{E}^T\mathbf{B} \\ -\mathbf{B}\mathbf{E} & \mathbf{B} \end{bmatrix}, \qquad \frac{\partial}{\partial x} \approx \mathbf{M}^{-1}\begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \mathbf{Q}_h$$

- Akin to adding correction terms similar to "$\mathbf{E}\boldsymbol{f}(\mathbf{u}) - \boldsymbol{f}(\mathbf{E}\mathbf{u})$".

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

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

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

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j\right), \qquad \widetilde{\mathbf{u}} = \text{ evaluate } \boldsymbol{u}\left(\Pi_N \boldsymbol{v}(\mathbf{u})\right).$$

- Entropy stability if conservative ($\mathbf{Q}_h\mathbf{1} = \mathbf{0}$). Equivalent to a weak version of the generalized SBP property.

$$\mathbf{Q}^T\mathbf{1} = \mathbf{E}^T\mathbf{B}\mathbf{1} \qquad \text{(related to quadrature accuracy)}$$

Chan (2019). *Skew-symmetric entropy stable modal discontinuous Galerkin formulations*.

Chan (2018). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods*.

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

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

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

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j\right), \qquad \widetilde{\mathbf{u}} = \text{ evaluate } \boldsymbol{u}\left(\Pi_N \boldsymbol{v}(\mathbf{u})\right).$$

- Entropy stability if conservative ($\mathbf{Q}_h\mathbf{1} = \mathbf{0}$). Equivalent to a weak version of the generalized SBP property.

$$\mathbf{Q}^T\mathbf{1} = \mathbf{E}^T\mathbf{B}\mathbf{1} \qquad \text{(related to quadrature accuracy)}$$

Chan (2019). *Skew-symmetric entropy stable modal discontinuous Galerkin formulations*.
Chan (2018). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods*.

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

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

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

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j\right), \qquad \widetilde{\mathbf{u}} = \text{ evaluate } \boldsymbol{u}\left(\Pi_N \boldsymbol{v}(\mathbf{u})\right).$$

- Entropy stability if conservative ($\mathbf{Q}_h\mathbf{1} = \mathbf{0}$). Equivalent to a weak version of the generalized SBP property.

$$\mathbf{Q}^T\mathbf{1} = \mathbf{E}^T\mathbf{B}\mathbf{1} \qquad \text{(related to quadrature accuracy)}$$

Chan (2019). *Skew-symmetric entropy stable modal discontinuous Galerkin formulations.*

Chan (2018). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods.*

Entropy stable schemes using hybridized SBP operators

- Replace SBP operator with hybridized SBP operator

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

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

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j\right), \qquad \widetilde{\mathbf{u}} = \text{ evaluate } \boldsymbol{u}\left(\Pi_N \boldsymbol{v}(\mathbf{u})\right).$$

- Entropy stability if conservative ($\mathbf{Q}_h\mathbf{1} = \mathbf{0}$). Equivalent to a weak version of the generalized SBP property.

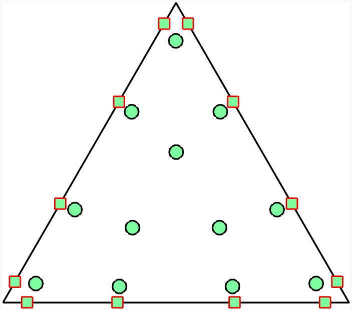
$$\mathbf{Q}^T\mathbf{1} = \mathbf{E}^T\mathbf{B}\mathbf{1} \qquad \text{(related to quadrature accuracy)}$$

Chan (2019). *Skew-symmetric entropy stable modal discontinuous Galerkin formulations*.

Chan (2018). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods*.

Example: high order DG on triangular meshes

- Degree N polynomial approximation + degree $\geq 2N$ volume/face quadratures.
- Uniform 32×32 mesh: degree $N = 3$, CFL $.125$, only dissipation from DG interface penalization.



Results computed on larger periodic domain ("natural" boundary conditions not entropy stable).

Recent work

Recent developments for modal ESDG formulations

- Non-conforming quadrilateral and hexahedral meshes (with Bencomo, Fernandez, Carpenter)
- Networks and multi-dimensional domains (with Philip Wu)
- Projection-based reduced order modeling
- Jacobian matrices, time-implicit solvers (with Christina Taylor)
- Compressible Navier-Stokes (with Yimin Lin, Warburton)

Recent work

Nonlinear conservation laws on networks and multi-dimensional domains (with Philip Wu)

Motivation: network-like domains

Flows on network-like domains often behave like one-dimensional channels (with some 2D effects at interfaces).

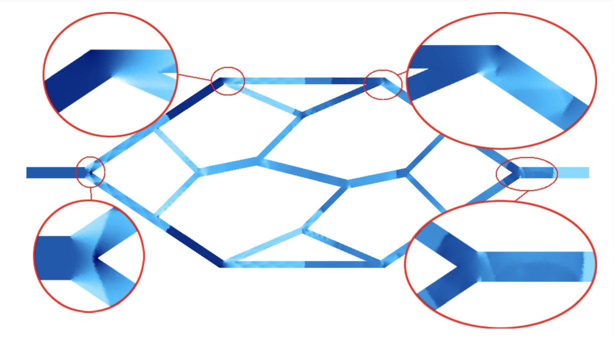


Figure from Bellamoli, Müller, Toro (2017)

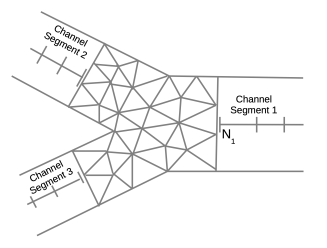
Network-like domains

Two approaches to treating network domains:

- Derive entropy stable junction fluxes by "mixing" 1D flux contributions with channel widths A_i at the ith channel.

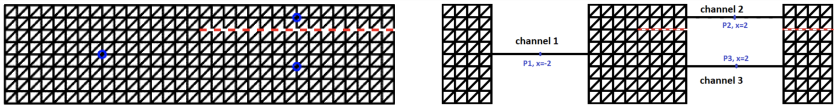
$$\boldsymbol{f}_{J,i} = \frac{1}{|I_J|}\sum_{j\in I_J} c_{ij}\boldsymbol{f}_S\left(\boldsymbol{u}_i,\boldsymbol{u}_j\right), \qquad \sum_{j\in I_J} c_{ij} = 1, \quad A_i c_{ij} = A_j c_{ji}.$$

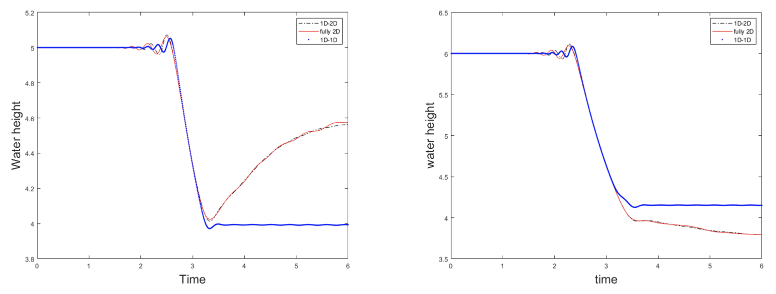
- Treat junctions as small 2D domains and construct entropy stable 1D-2D couplings.



Purely 1D junction model (shallow water)

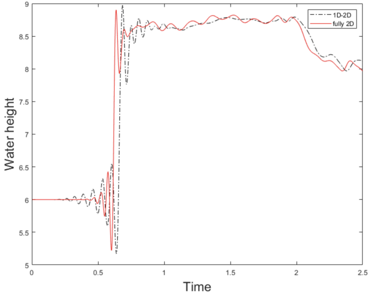
The 2D junction model closely matches the full 2D model, while the 1D junction model can fail to predict 2D behavior.

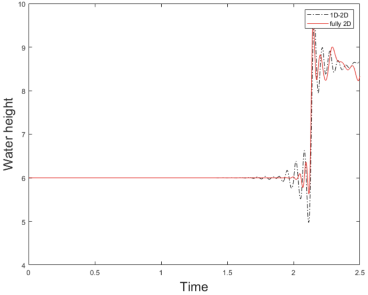


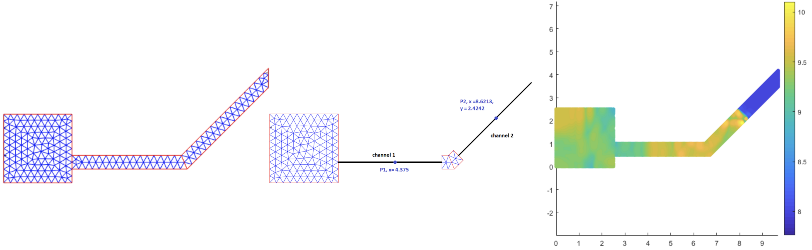


y-averaged water height at P2 and P3 for a domain with an internal barrier and discontinuous initial conditions in each channel.

1D-2D modeling of a dam break (shallow water)







Recent work

Compressible Navier-Stokes equations (with Yimin Lin)

Compressible Navier-Stokes: discretization of viscous terms

The compressible Navier-Stokes equations are given by inviscid fluxes $\boldsymbol{f}_i(\boldsymbol{u})$ and viscous fluxes $\boldsymbol{g}_i(\boldsymbol{u})$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \boldsymbol{f}_i(\boldsymbol{u})}{\partial x_i} = \sum_{i=1}^{d} \frac{\partial \boldsymbol{g}_i(\boldsymbol{u})}{\partial x_i}.$$

Symmetrize viscous terms by transforming to entropy variables $\boldsymbol{v}(\boldsymbol{u})$

$$\sum_{i=1}^{d} \frac{\partial \boldsymbol{g}_i(\boldsymbol{u})}{\partial x_i} = \sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}(\boldsymbol{v})\frac{\partial \boldsymbol{v}}{\partial x_j}\right), \qquad \boldsymbol{K}_{ij} \succeq \boldsymbol{0}.$$

DG formulation and boundary conditions

Write viscous terms as a first order system

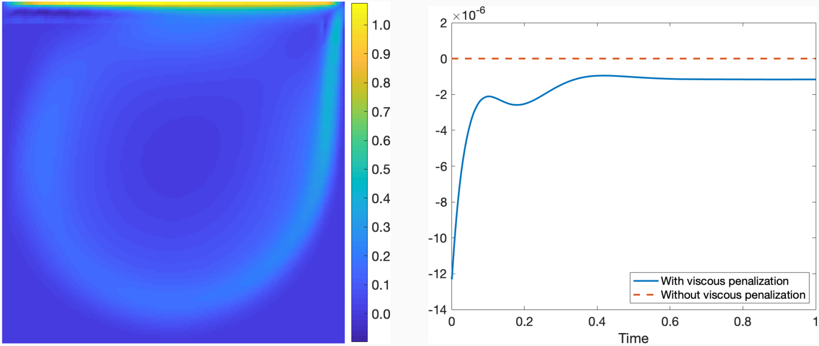
$$\boldsymbol{\Theta}_i = \frac{\partial \boldsymbol{v}}{\partial x_i}$$

$$\boldsymbol{\sigma}_i = \sum_{j=1}^{d} \boldsymbol{K}_{ij}(\boldsymbol{v})\Theta_j$$

$$\boldsymbol{g}_{\text{visc}} = \sum_{i=1}^{d} \frac{\partial \boldsymbol{\sigma}_i}{\partial x_i}.$$

- Entropy dissipative under standard DG discretizations and appropriate BCs on $\boldsymbol{u}$, entropy variables $\boldsymbol{v}$, and $\boldsymbol{\sigma}_i$
- Exactly entropy conservative for no-slip adiabatic and symmetry walls, entropy "mimetic" for no-slip isothermal walls.

Verification of entropy conservation/dissipation

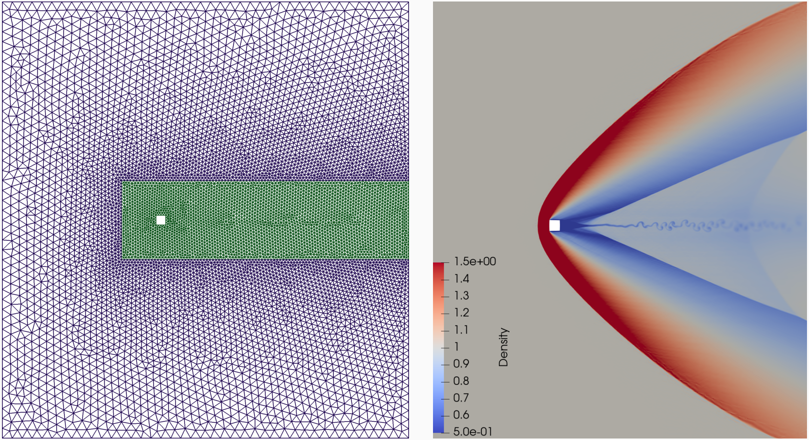


(a) Adiabatic lid-driven cavity, $\mathrm{Ma} = .1, \mathrm{Re} = 1000$

(b) Viscous entropy dissipation

Figure 1: Imposition of BCs guarantees correct energy dissipation, independently of solution resolution.

Flow over a square cylinder



(a) Mesh

(b) Zoom of ρ at $T_{\text{final}} = 100$

Figure 2: Mesh and density at $T_{\text{final}} = 100$ for $\mathrm{Re} = 10^4, \mathrm{Ma} = 1.5$, and a degree $N = 3$ approximation.

Recent work

Efficient computation of Jacobian matrices (with Christina Taylor)

Current methods for computing Jacobian matrices

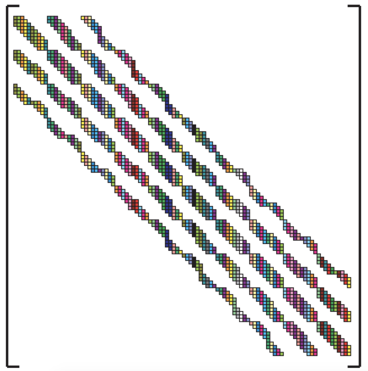


Figure from Gebremedhin, Manne, Pothen (2005), *What color is your Jacobian? Graph coloring for computing derivatives.*

- Implicit time-stepping: compute Jacobian matrices using automatic differentiation (AD)
- Graph coloring reduces costs, but only for sparse matrices
- Cost of AD scales with input and output dimensions.

Cost of computing dense Jacobian blocks

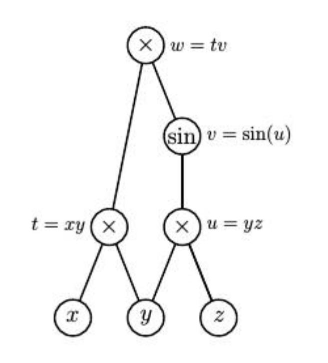


Image from Austin (2017), *How to Differentiate with a Computer.*

- Graph coloring AD expensive for dense matrix blocks (e.g., high order DG methods).
- Problem: cost of AD scales with input and output dimensions.
- Can reduce costs by passing AD through to the nonlinear flux function $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$.

Jacobian matrices for flux differencing

Theorem

Assume $\mathbf{Q} = \pm\mathbf{Q}^T$. *Consider a scalar "collocation" discretization*

$$\mathbf{r}(\mathbf{u}) = (\mathbf{Q} \circ \mathbf{F})\,\mathbf{1}, \qquad \mathbf{F}_{ij} = f_S(\mathbf{u}_i, \mathbf{u}_j).$$

The Jacobian matrix is then

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{u}} = (\mathbf{Q} \circ \partial\mathbf{F}_R) \pm \operatorname{diag}\left(\mathbf{1}^T(\mathbf{Q} \circ \partial\mathbf{F}_R)\right),$$

$$(\partial\mathbf{F}_R)_{ij} = \left.\frac{\partial f_S(u_L, u_R)}{\partial u_R}\right|_{\mathbf{u}_i, \mathbf{u}_j}.$$

AD is efficient for $O(1)$ inputs/outputs!

Separates discretization matrix $\mathbf{Q}$ and AD for flux contributions

Observations about flux differencing Jacobian formulas

Separates "template" matrix Q and flux contributions.

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{u}} = \left(\mathbf{Q} \circ \partial \mathbf{F}_R\right) \pm \operatorname{diag}\left(\mathbf{1}^T \left(\mathbf{Q} \circ \partial \mathbf{F}_R\right)\right),$$

$$\left(\partial \mathbf{F}_R\right)_{ij} = \left.\frac{\partial f_S(u_L, u_R)}{\partial u_R}\right|_{\mathbf{u}_i, \mathbf{u}_j}.$$

$O(1)$ inputs/outputs $\rightarrow$ AD is efficient. In Julia:

```
using ForwardDiff
f(uL,uR) = (1/6)*(uL^2 + uL*uR + uR^2)
dF(uL,uR) = ForwardDiff.derivative(uR->f(uL,uR),uR)
```

Extensions: systems of nonlinear conservation laws

Assume n fields; nonlinear term is now

$$\mathbf{r}(\mathbf{u}) = ((\mathbf{I}_n \otimes \mathbf{Q}) \circ \mathbf{F})\, \mathbf{1} = \begin{bmatrix} (\mathbf{Q} \circ \mathbf{F}_1)\, \mathbf{1} \\ \vdots \\ (\mathbf{Q} \circ \mathbf{F}_n)\, \mathbf{1} \end{bmatrix}, \qquad (\mathbf{F}_\ell)_{ij} = (\boldsymbol{f}_S(\mathbf{u}_i, \mathbf{u}_j))_\ell \,.$$

Jacobian matrix involves Jacobian of $\boldsymbol{f}_S$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} \mathbf{F}_{1,\mathbf{u}_1} & \dots & \mathbf{F}_{1,\mathbf{u}_n} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{n,\mathbf{u}_1} & \dots & \mathbf{F}_{n,\mathbf{u}_n} \end{bmatrix}, \qquad \partial \mathbf{F}_{i,\mathbf{u}_j} \Longleftrightarrow \frac{\partial\left(\boldsymbol{f}_S\right)_i}{\partial \boldsymbol{u}_{R,j}}$$

$$\mathbf{F}_{i,\mathbf{u}_j} = \left(\mathbf{Q} \circ \partial \mathbf{F}_{i,\mathbf{u}_j}\right) \pm \operatorname{diag}\left(\mathbf{1}^T \left(\mathbf{Q} \circ \partial \mathbf{F}_{i,\mathbf{u}_j}\right)\right)$$

Extensions: dissipative terms

- Define anti-symmetric entropy dissipative flux (e.g., Lax-Friedrichs penalization $\frac{\lambda}{2}(\boldsymbol{u}_L - \boldsymbol{u}_R)$)

$$\boldsymbol{d}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = -\boldsymbol{d}_S(\boldsymbol{u}_R, \boldsymbol{u}_L)$$
$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{d}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) \geq 0.$$

- Dissipation matrix $\mathbf{K}$ (symmetric, non-negative entries)

$$\mathbf{d}(\mathbf{u}) = (\mathbf{K} \circ \mathbf{D})\,\mathbf{1}, \qquad \mathbf{D}_{ij} = \boldsymbol{d}_S(\mathbf{u}_i, \mathbf{u}_j).$$

- Jacobian of $\mathbf{d}(\mathbf{u})$ is similar to previous formulas

$$\frac{\partial \mathbf{d}}{\partial \mathbf{u}} = -(\mathbf{K} \circ \partial \mathbf{D}_R^T) + \operatorname{diag}\left(\left(\mathbf{K} \circ \partial \mathbf{D}_R^T\right)\mathbf{1}\right).$$

Extensions: modal DG methods (entropy projection)

$$\underbrace{\boxed{\mathbf{V}_h^T}}_{(N_p \times N_{\text{total}})} \underbrace{\boxed{\mathbf{Q} \circ \partial \mathbf{F}}}_{(N_{\text{total}} \times N_{\text{total}})} \underbrace{\boxed{\left.\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}\right|_{\mathbf{V}_h \mathbf{P} \mathbf{v}_q}}}_{(N_{\text{total}} \times N_{\text{total}})} \underbrace{\boxed{\mathbf{V}_h \mathbf{P}}}_{(N_{\text{total}} \times N_q)} \underbrace{\boxed{\left.\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}}\right|_{\mathbf{V}\widehat{\mathbf{u}}}}}_{(N_q \times N_q)} \underbrace{\boxed{\mathbf{V}}}_{(N_q \times N_p)}$$

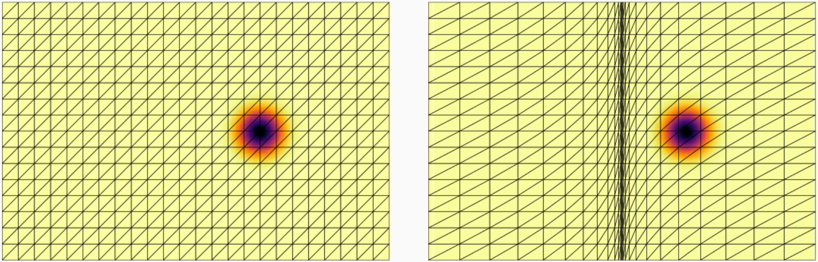
- Degrees of freedom are N_p "modal" coefficients $\widehat{\mathbf{u}}$.
- Fluxes use entropy projected variables $\boldsymbol{u}\left(\Pi_N \boldsymbol{v}(\boldsymbol{u}_h)\right)$.
- Requires projection/interpolation matrices + Jacobians of transforms between conservative/entropy variables.

Computational timings

Jacobian timings for $f_S(u_L, u_R) = \frac{1}{6}\left(u_L^2 + u_L u_R + u_R^2\right)$ and dense differentiation matrices $\mathbf{Q} \in \mathbb{R}^{N\times N}$.

	N = 10	N = 25	N = 50
Direct automatic differentiation	5.666	60.388	373.633
`FiniteDiff.jl`	1.429	17.324	125.894
Jacobian formula (analytic deriv.)	.209	1.005	3.249
Jacobian formula (AD flux deriv.)	.210	1.030	3.259
Evaluation of $\mathbf{f}(\mathbf{u})$ (reference)	.120	.623	2.403

Implicit midpoint method for compressible Euler



(a) Uniform, L^2 error .0901 **(b)** [Anisotropic, L^2 error .0935

Figure 3: Solutions for a degree $N = 3$ modal DG method with $dt = .1$ on uniform and "squeezed" meshes.

Summary and future work

- High order entropy stable DG formulations improve robustness for shocks and under-resolved features.
- Current work: positive schemes with sub-cell resolution.

This work is supported by the NSF under awards DMS-1719818, DMS-1712639, and DMS-CAREER-1943186.

Thank you! Questions?

Chan, Lin, Warburton (2020). *Entropy stable modal discontinuous Galerkin schemes and wall boundary conditions for the compressible Navier-Stokes equation.*

Wu, Chan (2020). *Entropy stable discontinuous Galerkin methods for nonlinear conservation laws on networks and multi-dimensional domains.*

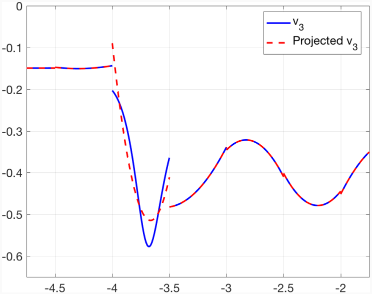
Chan, Taylor (2020). *Efficient computation of Jacobian matrices for ES-SBP schemes.*

Chan (2018). *On discretely entropy conservative and entropy stable discontinuous Galerkin methods.*

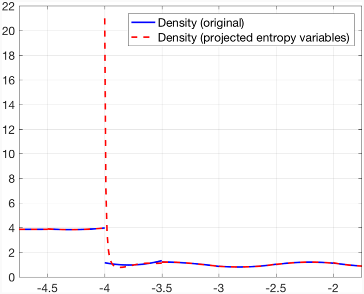
Additional slides

Loss of control with the entropy projection

- For $(N+1)$-point Lobatto, $\widetilde{\boldsymbol{u}} = \boldsymbol{u}$ at nodal points.
- For $(N+2)$-point Gauss, discrepancy between $\boldsymbol{v}(\boldsymbol{u})$ and projection on the boundary of elements.
- Still need positivity of thermodynamic quantities for stability!



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



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

Taylor-Green vortex

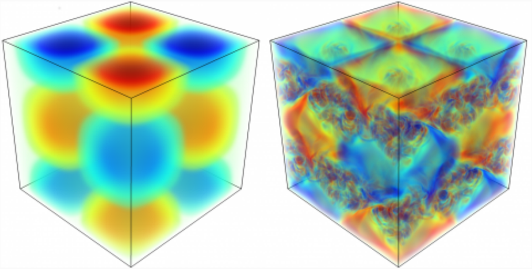
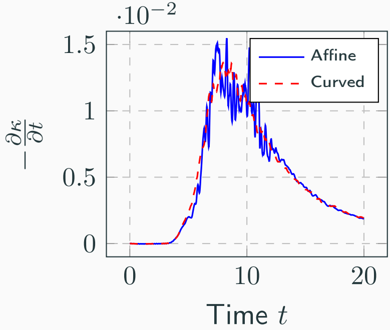


Figure 4: Isocontours of z-vorticity for Taylor-Green at $t = 0, 10$ seconds.

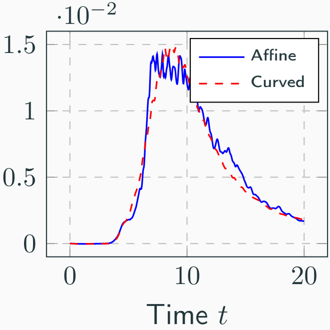
- Simple turbulence-like behavior (generation of small scales).
- Inviscid Taylor-Green: tests robustness w.r.t. under-resolved solutions.

https://how4.cenaero.be/content/bs1-dns-taylor-green-vortex-re1600.

3D inviscid Taylor-Green vortex



(a) GLL collocation



(b) Gauss collocation

Figure 5: Kinetic energy dissipation rate for entropy stable GLL and Gauss collocation schemes with $N = 7$ and $h = \pi/8$.