Shock Capturing Methods for Flux Reconstruction

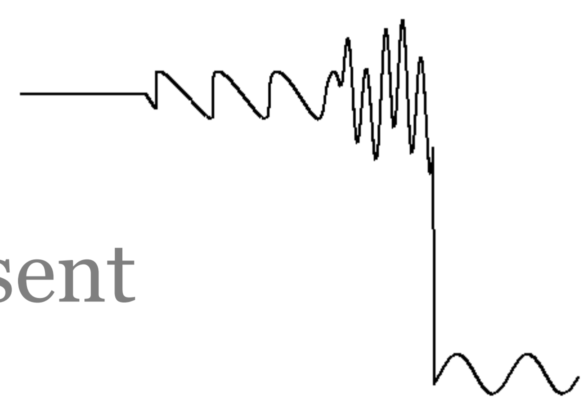
Will Trojak

Scope

- Why Shock Capturing?
- Invariance Preserving Methodology
- Preliminary Results
- Summary and Future Developments

Why Shock Capturing?

- More Physics
- Currently parametric
- P-adaptive methods present several issues

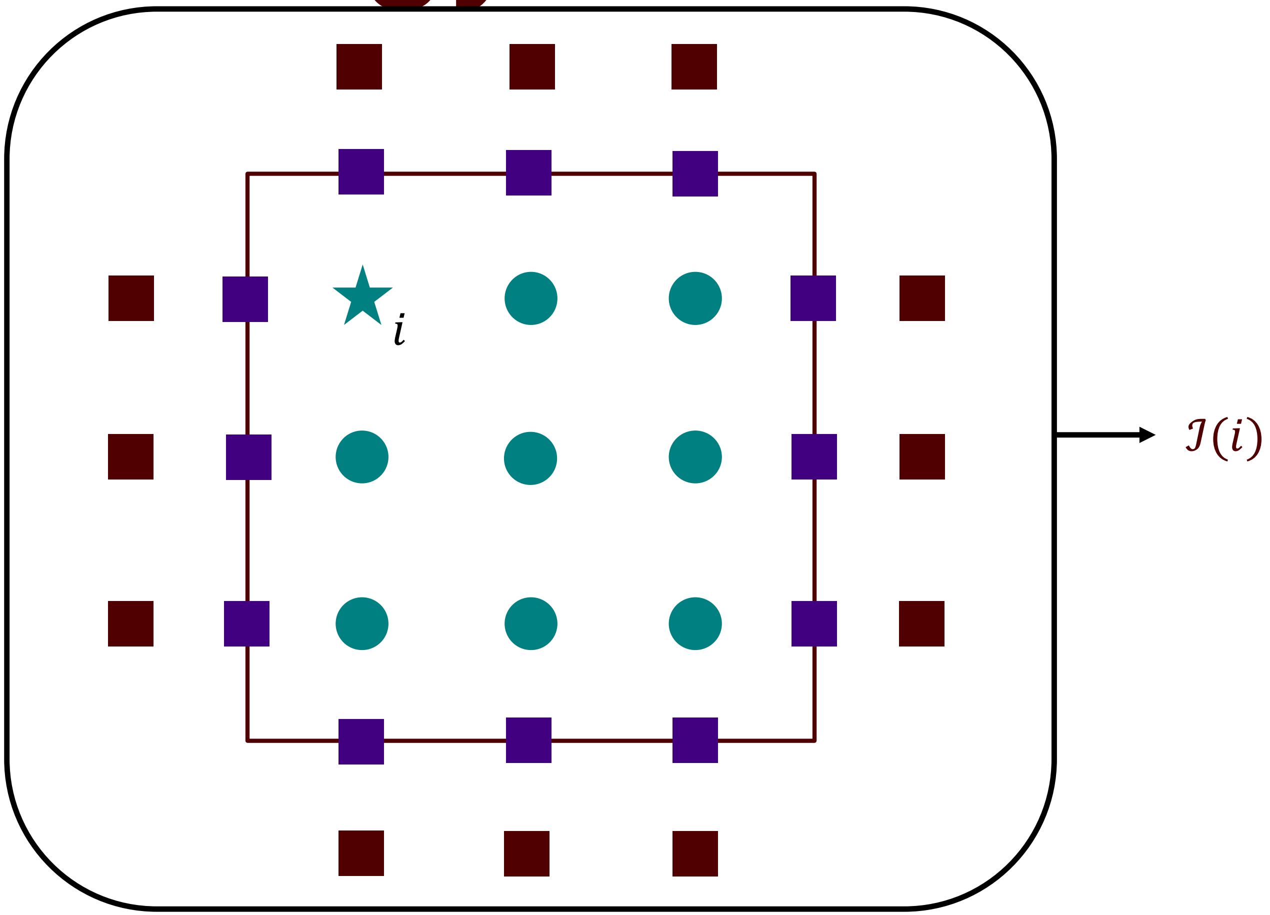


$p = 0$, DoF $= 20k$

Current Shock Capturing

Method	Parametric	Op. Cheap	"Stable"
AV/Per-Olof	✓	✓	✗?
Filtering	✓	✓	✓?
Adaptation/ Moving mesh	✗?	✗	✓?

Methodology



Methodology

So... $$\partial_t \boldsymbol{u}_i = -\sum_{j\in\mathcal{I}(i)} \mathbb{f}_j \cdot \boldsymbol{c}_{ij} + \sum_{j\in\mathcal{I}(i)} d_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i)$$

Now... $$\sum_{j\in\mathcal{I}(i)} \boldsymbol{c}_{ij} = 0 = \sum_{j\in\mathcal{I}(i)} d_{ij}$$

Methodology

$$\partial_t \boldsymbol{u}_i = -\sum_{j \in \mathcal{I}(i)} (\mathbb{f}_j + \mathbb{f}_i) \cdot \boldsymbol{c}_{ij} - d_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i)$$

Which we solve by using…

$$d_{ij} = \max(\lambda_{max}(\boldsymbol{n}_{ij}, \boldsymbol{u}_i, \boldsymbol{u}_j)|\boldsymbol{c}_{ij}|, \lambda_{max}(\boldsymbol{n}_{ji}, \boldsymbol{u}_j, \boldsymbol{u}_i)|\boldsymbol{c}_{ji}|)$$

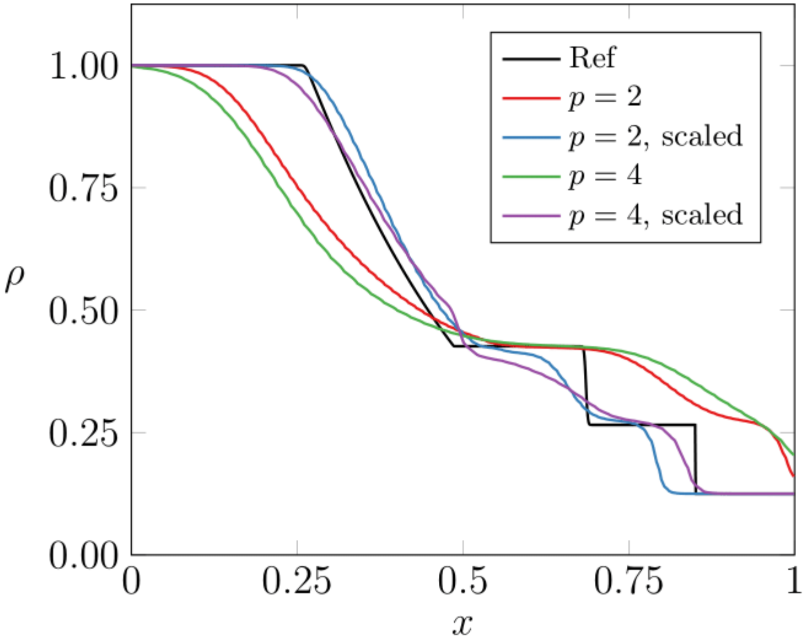
Methodology

$$\partial_t \boldsymbol{u}_i = -S\left(\sum_{j\in\mathcal{I}(i)} (\mathbb{f}_j + \mathbb{f}_i)\cdot \boldsymbol{c}_{ij} - d_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i)\right)$$

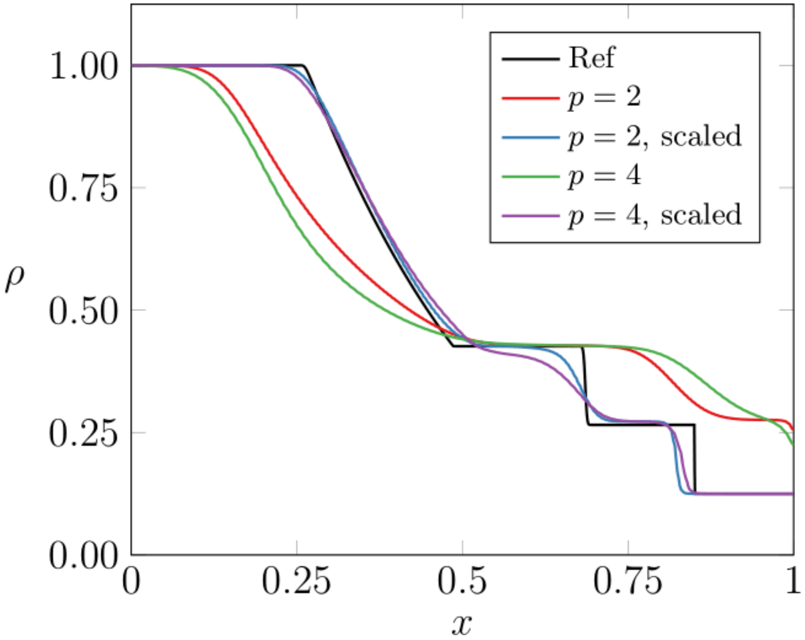
$$S = \frac{||\nabla \cdot \mathbb{f}||}{||\nabla \cdot \mathbb{f} + \boldsymbol{AV}||}$$

Any Quick Questions on Methodology?

Test 1 (Sod)



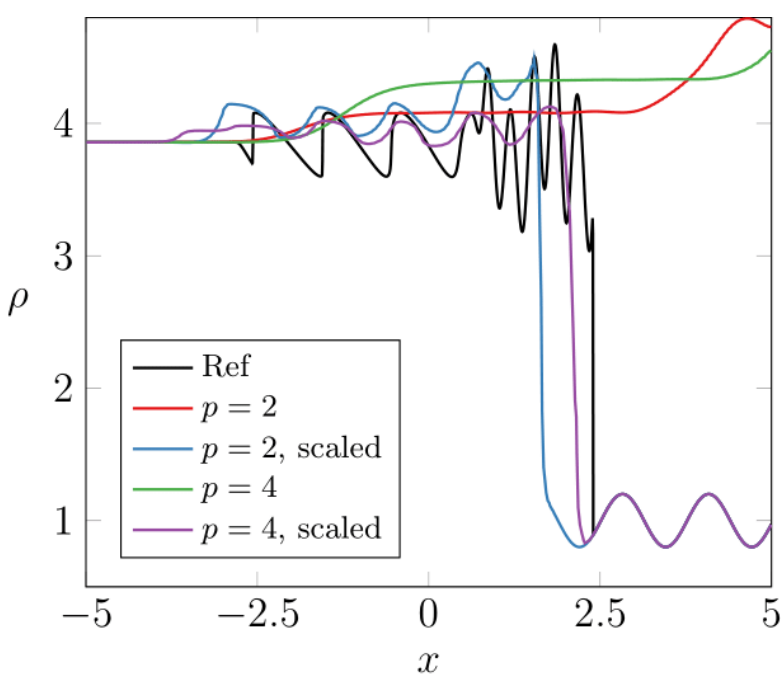
DoF = 255



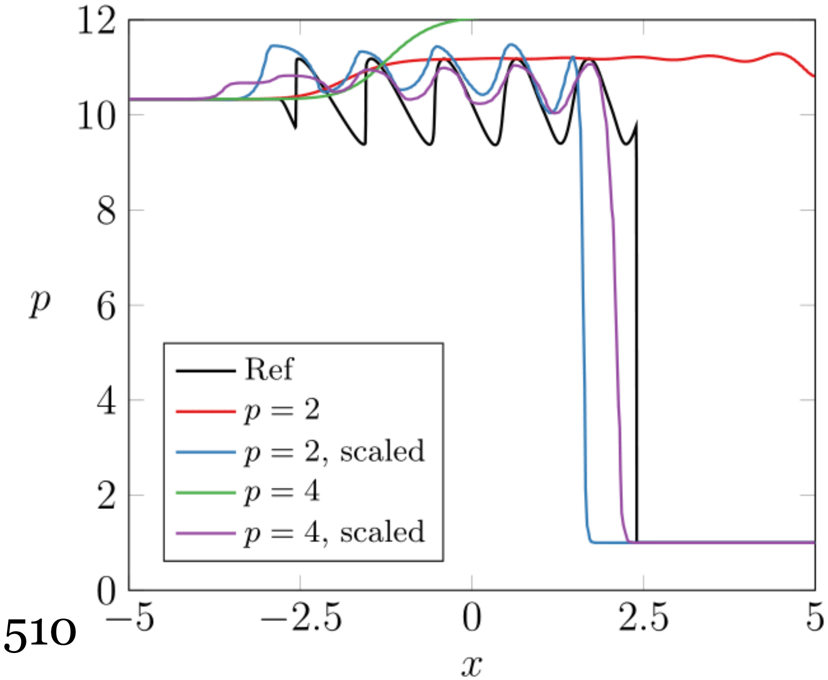
DoF = 510

$$\boldsymbol{w}_L = \begin{bmatrix} \rho = 1 \\ u = 0 \\ p = 1 \end{bmatrix}, \quad \boldsymbol{w}_R = \begin{bmatrix} \rho = 0.125 \\ u = 0 \\ p = 0.1 \end{bmatrix}$$

Test 2 (Shu-Osher)



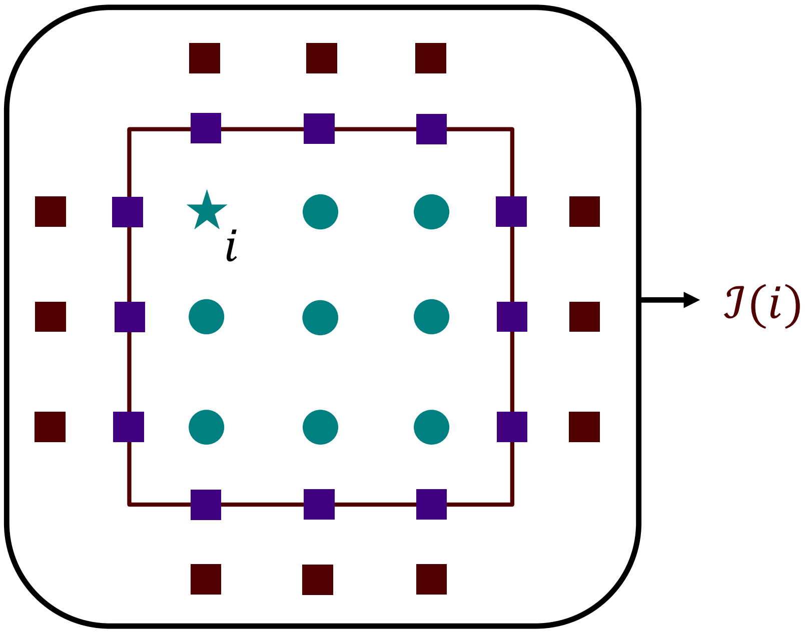
DoF = 510



$$\boldsymbol{w}_L = \begin{bmatrix} \rho \approx 3.8 \\ u \approx 2.6 \\ p \approx 10.3 \end{bmatrix}, \quad \boldsymbol{w}_R = \begin{bmatrix} \rho = 1 + 0.2 \sin 5x \\ u = 0 \\ p = 1 \end{bmatrix}$$

How can we improve this?

Sparse Graph-Viscosity



Sparse Graph-Viscosity

Low Order…

$$\partial_t \boldsymbol{u}_i^l = -\sum_{j \in \mathcal{S}(i)} \mathbb{f}_j \cdot \hat{\boldsymbol{c}}_{ij} - d_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i)$$

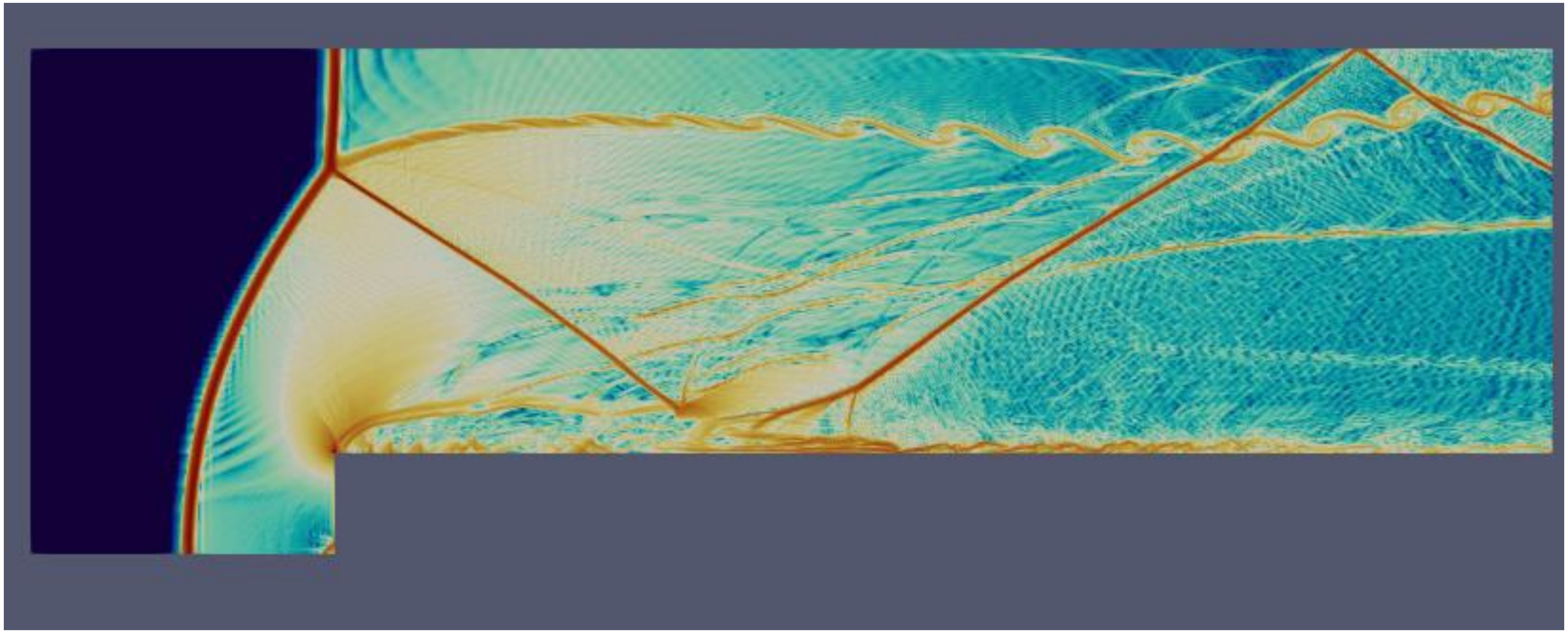
High Order…

$$\partial_t \boldsymbol{u}_i^h = -\sum_{j \in \mathcal{I}(i)} \mathbb{f}_j \cdot \boldsymbol{c}_{ij}$$

Summary and Future

- Non-parametric shock capturing method
- Developing sparse methods for FR
- Currently working on PyFR implementation
- Developing GPU accelerated convex limiting

Thanks to Tarik Dzanic for his work



Any Questions?

References

- “Second-Order Invariant Domain Preserving Approximation of the Euler Equations Using Convex Limiting” Jean-Luc Guermond, Murtazo Nazarov, Bojan Popov, and Ignacio Tomas, SIAM Journal on Scientific Computing 2018 40:5, A3211-A3239

- “Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting” Will Pazner, 2020 ArXiV 2004:08503