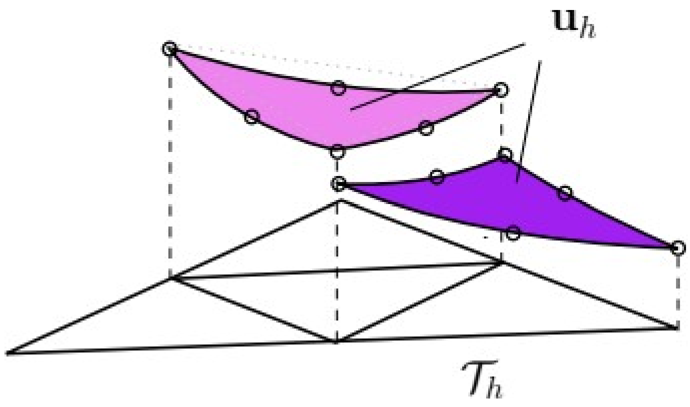
A C++ HDG Solver for the 2D Compressible Euler and Navier-Stokes Equations

Ben Couchman

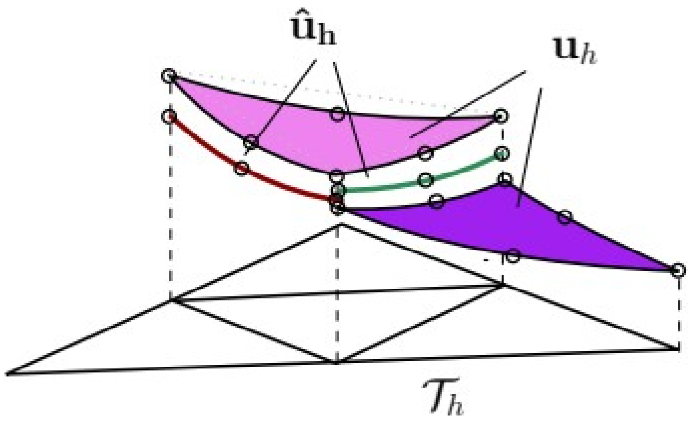
Motivation for HDG

- DG is more compact high order than Finite Volume
  - Higher order polynomials in cell
- **Problem:** Repeated solution variables at faces
  - More expensive



Motivation for HDG

- **Solution:** Add solution variables at faces
- Now can solve face variables globally
  - Element variables solve trivially parellelizable



Formulation

- Nonlinear advection-diffusion PDE

$$\nabla \cdot \left[\mathbf{F}^{adv}\left(\mathbf{u}\right) - \kappa\left(\mathbf{u}\right)\nabla\mathbf{u}\right] = \mathbf{f}\left(\mathbf{x}\right)$$

- Split into coupled 1st order PDEs

$$\nabla \cdot \left[\mathbf{F}^{adv}\left(\mathbf{u}\right) - \kappa\left(\mathbf{u}\right)\mathbf{q}\right] = \mathbf{f}\left(\mathbf{x}\right)$$

$$\nabla\mathbf{u} = \mathbf{q}$$

- Make sure we're conservative

$$\left[\mathbf{F}^{adv}\left(\hat{\mathbf{u}}\right) + \tau\left(\mathbf{u} - \hat{\mathbf{u}}\right)\right]_{LHS} = \left[\mathbf{F}^{adv}\left(\hat{\mathbf{u}}\right) + \tau\left(\mathbf{u} - \hat{\mathbf{u}}\right)\right]_{RHS}$$

Formulation

- In weak form,

$$(\mathbf{q}, \mathbf{v})_{\mathcal{T}_h} + (\mathbf{u}, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} + \langle \hat{\mathbf{u}}, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} = 0$$

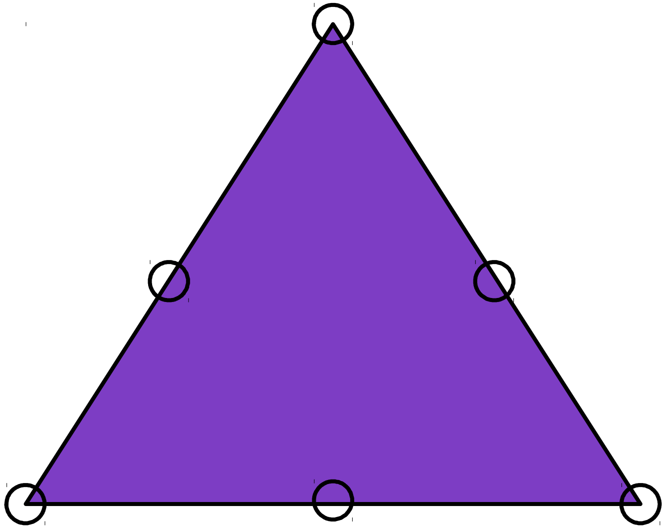
$$(\kappa \nabla \cdot \mathbf{q}, \mathbf{w})_{\mathcal{T}_h} + \left(\mathbf{F}^{adv}(\mathbf{u}), \nabla \mathbf{w}\right)_{\mathcal{T}_h} + \langle \mathbf{F}^{adv}(\hat{\mathbf{u}}) \cdot \mathbf{n} + \tau (\mathbf{u} - \hat{\mathbf{u}}), \mathbf{w} \rangle_{\partial \mathcal{T}_h} = (\mathbf{f}, \mathbf{w})_{\mathcal{T}_h}$$

$$\langle \kappa \mathbf{q} + \mathbf{F}^{adv}(\hat{\mathbf{u}}) \cdot \mathbf{n} + \tau (\mathbf{u} - \hat{\mathbf{u}}), \boldsymbol{\mu} \rangle_{\partial \mathcal{T}_h} = 0$$

$$\forall (\mathbf{w}, \mathbf{w}, \boldsymbol{\mu}) \in V_{h,p} \times W_{h,p} \times M_{h,p}$$

Implementation

- Equally spaced nodal bases
  - Conditioning problems for very high orders
- Curved (isoparametric) elements
- Tensor product Gaussian quadrature
  - Exact for polynomials up to 4p



Implementation

- After discretization

$$\begin{bmatrix} A & B & E \\ C & D & L \\ M & N & P \end{bmatrix} \begin{bmatrix} \delta q \\ \delta u \\ \delta \hat{u} \end{bmatrix} = \begin{bmatrix} H \\ F \\ G \end{bmatrix}$$

Implementation

- A B C D are block diagonal
- Decompose into two problems
- Global problem: $\mathbb{A} = P - \begin{bmatrix} M & N \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} E \\ L \end{bmatrix}$

$$\mathbb{B} = G - \begin{bmatrix} M & N \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} H \\ F \end{bmatrix}$$

$$\mathbb{A}\delta\hat{u} = \mathbb{B}$$

Implementation

- And a local problem (for each element)

$$\begin{bmatrix} \delta q \\ \delta u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \left(\begin{bmatrix} H \\ F \end{bmatrix} - \begin{bmatrix} E \\ L \end{bmatrix} \delta \hat{u} \right)$$

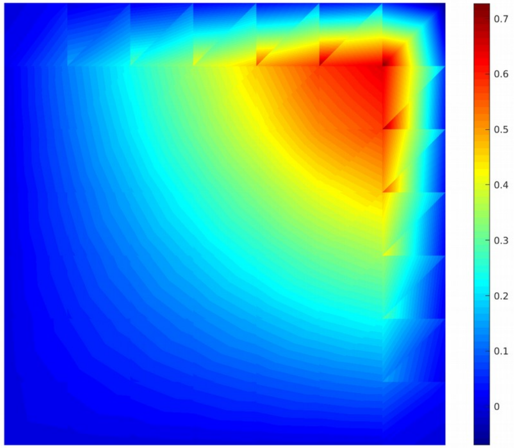
- Global and locals solves at each Newton iteration

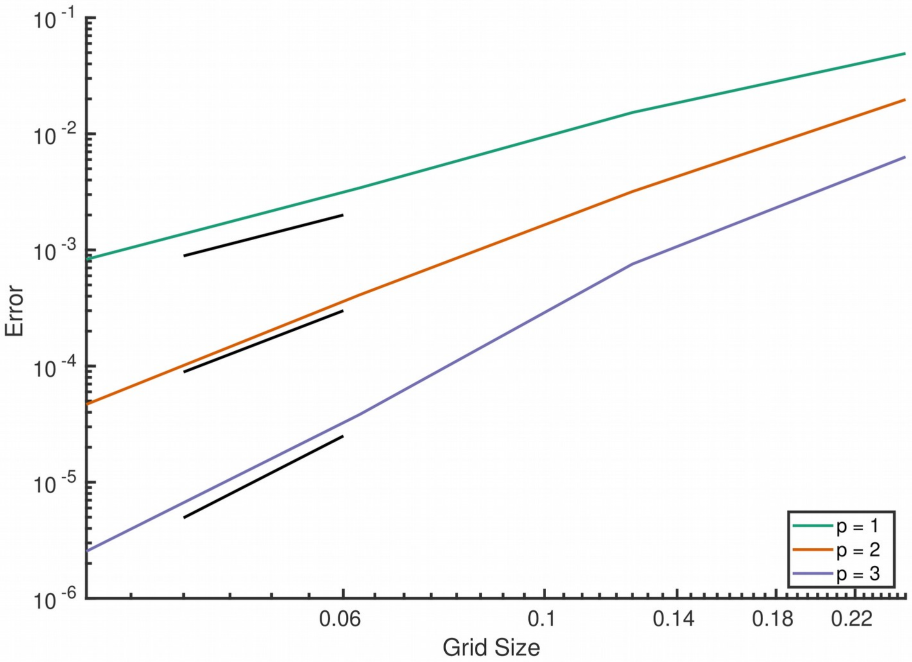
Burgers

$$\mathbf{F}^{adv}(\mathbf{u}) = \begin{bmatrix} \frac{1}{2}u^2 \\ \frac{1}{2}u^2 \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$\nabla \cdot \left[\mathbf{F}^{adv}(\mathbf{u}) - \kappa(\mathbf{u}) \nabla \mathbf{u}\right] = \mathbf{f}(\mathbf{x})$$

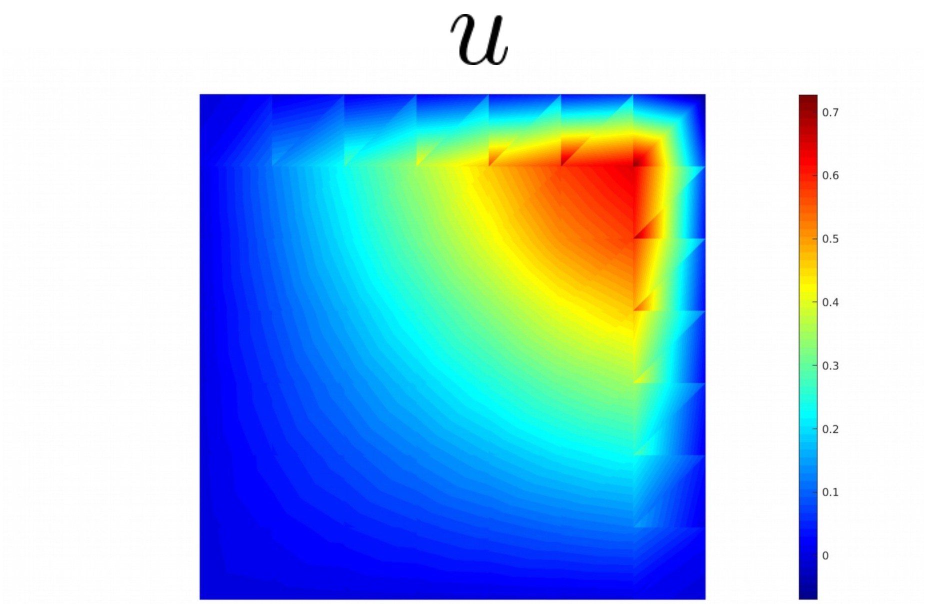




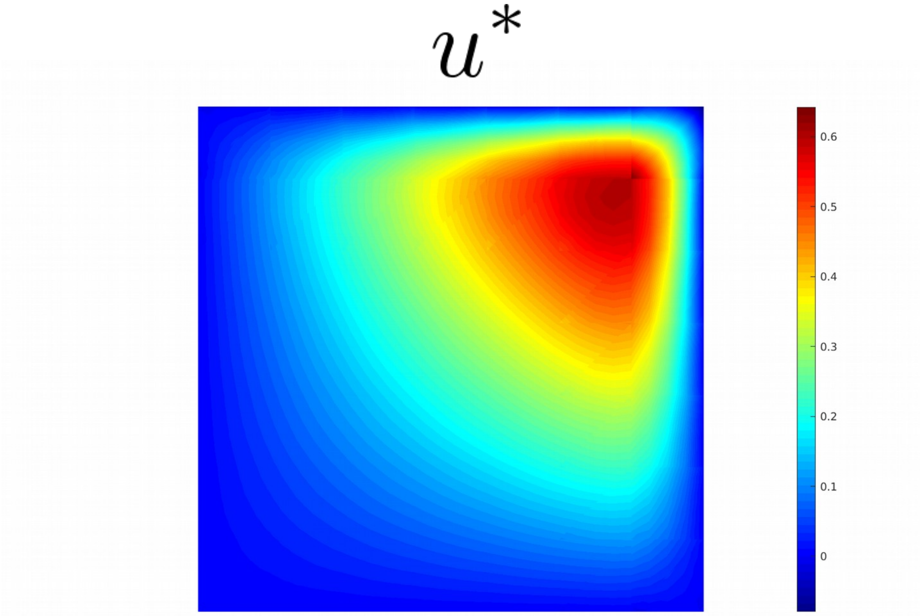
Can we do better?

- $\mathbf{q}$ is a (p+1) order accurate approximation of $\nabla \mathbf{u}$
- Obtain a new approximation for u
  - Solve: $\nabla \mathbf{u}^* = \mathbf{q}$
  - Subject to: $\int_{\mathcal{T}_h} \mathbf{u} \, d\Omega = \int_{\mathcal{T}_h} \mathbf{u}^* \, d\Omega$
- $\mathbf{u}^*$ is (p+2) order accurate

Burgers

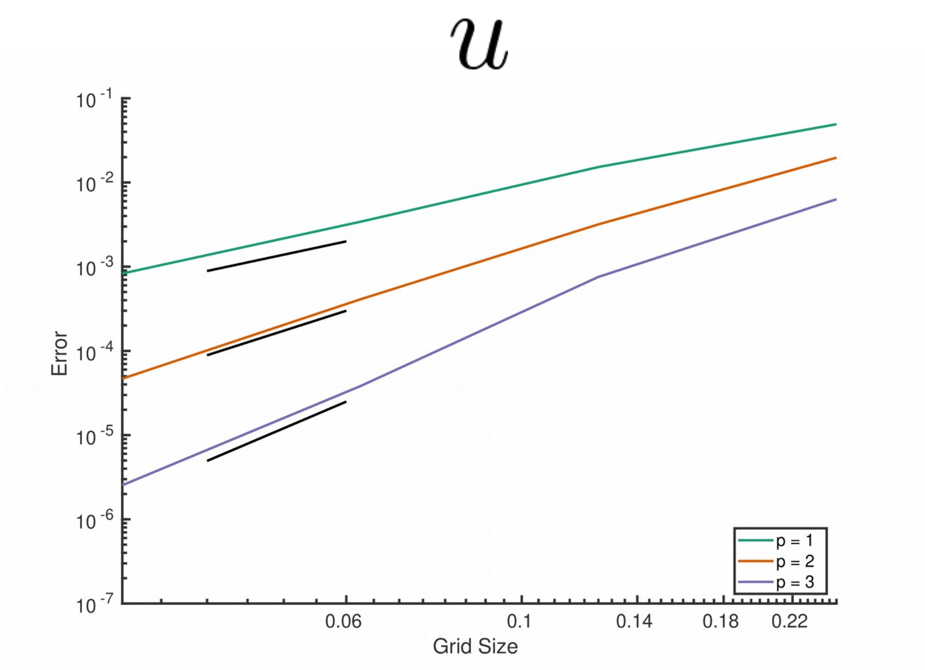


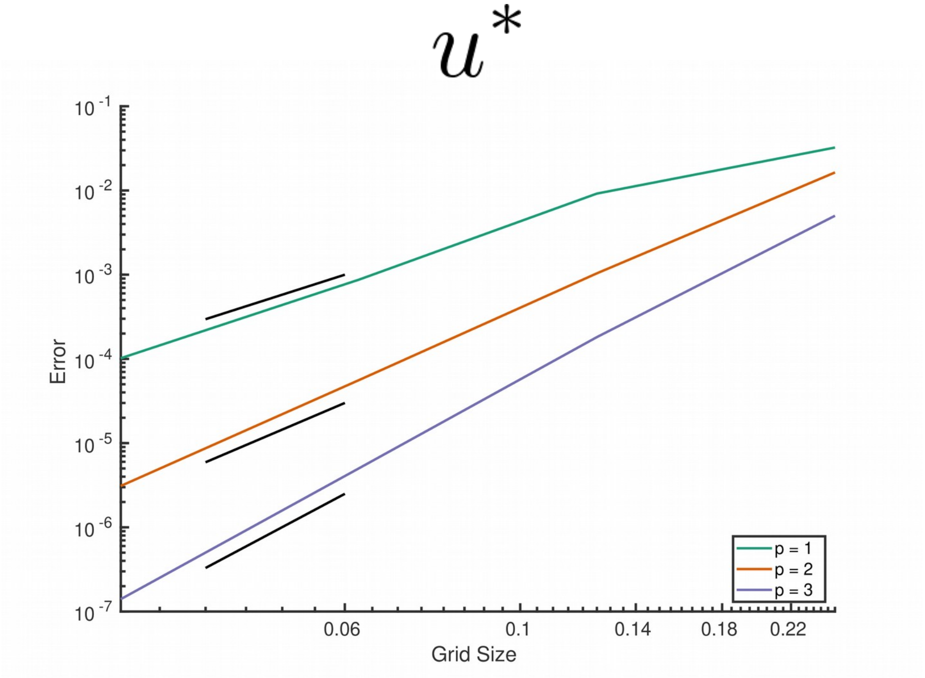
Error = 0.015



Error = 0.008

Burgers


u
10 -1
10 -2
10 -3
10 -4
10 -5
10 -6
10 -7
Error
0.06
0.1
0.14
0.18
0.22
Grid Size
p = 1
p = 2
p = 3

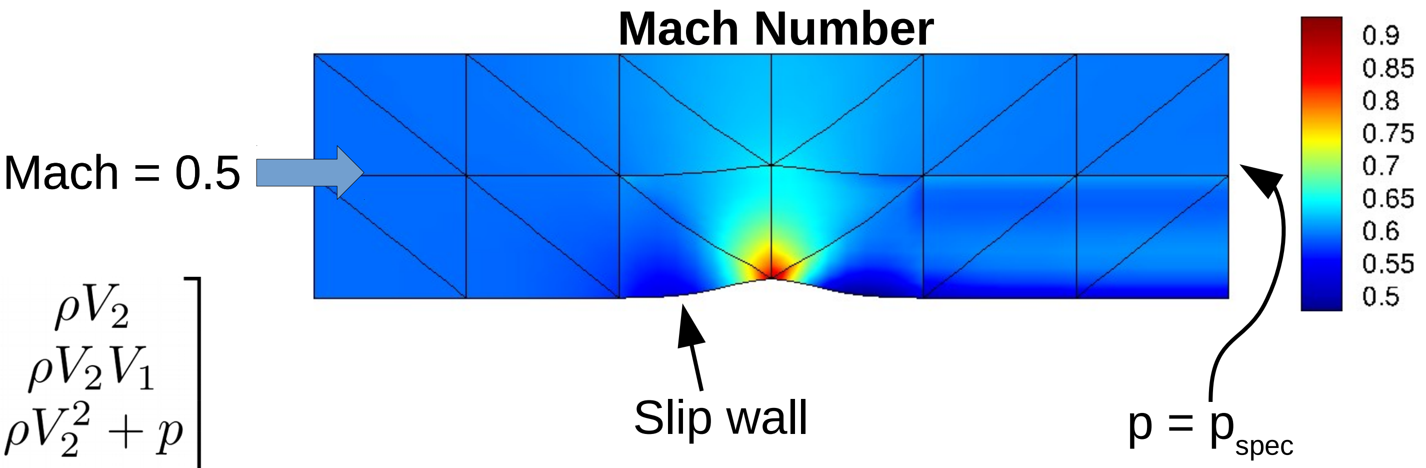

u^*
10 -1
10 -2
10 -3
10 -4
10 -5
10 -6
10 -7
Error
0.06
0.1
0.14
0.18
0.22
Grid Size
p = 1
p = 2
p = 3

Euler

$$\nabla \cdot \left[\mathbf{F}^{adv}(\mathbf{u}) - \kappa(\mathbf{u})\nabla\mathbf{u}\right] = \mathbf{f}(\mathbf{x})$$

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho V_1 \\ \rho V_2 \\ \rho E \end{bmatrix}$$

$$\mathbf{F}^{adv} = \begin{bmatrix} \rho V_1 & \rho V_2 \\ \rho V_1^2 + p & \rho V_2 V_1 \\ \rho V_1 V_2 & \rho V_2^2 + p \\ \rho V_1 H & \rho V_2 H \end{bmatrix}$$



Current Status

- HDG framework for solving

$$\nabla \cdot \left[\mathbf{F}^{adv}\left(\mathbf{u}\right) - \kappa\left(\mathbf{u}\right)\nabla\mathbf{u}\right] = \mathbf{f}\left(\mathbf{x}\right)$$

- Demonstrated design order accuracy on several PDEs
- Demonstrated post-processing to (p+2)

Future Work

- Limited robustness with Newton-Raphson
- Expand implementation to solve Navier-Stokes
- Limited application without shock-capturing

References

- Nguyen, N., Peraire, J., and Cockburn, B. (2009). *An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations.* Journal of Computational Physics 228 (2009), 3232-3254.
- Nguyen, N., Peraire, J., and Cockburn, B. (2009). *An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations*. Journal of Computational Physics 229 (2009), 8841-8855.
- Peraire, J., Nguyen, N., and Cockburn, B. (2010). *A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Naver-Stokes Equations.* 48th AIAA Aerospace Sciences Meeting, Orlando Florida: AIAA.
- Hesthaven, J.,and Warburton, T. (2008). *Nodal Discontinuous Galerkin Methods*. New York, New York: Springer.

Backup Slides

Poisson

$$\nabla \cdot \left[\mathbf{F}^{adv}(\mathbf{u}) - \kappa(\mathbf{u})\nabla\mathbf{u}\right] = \mathbf{f}(\mathbf{x})$$

$$\mathbf{F}^{adv}(\mathbf{u}) = 0$$

$$\kappa = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$