Space-Time Discontinuous Petrov-Galerkin Finite Elements for Transient Fluid Mechanics

Truman Ellis[a], Leszek Demkowicz[b], Jesse Chan[c], Nate Roberts[a] and Robert Moser[b]

[a] Sandia Laboratories
[b] ICES, University of Texas at Austin
[c] Dept. of Math., Rice University

Workshop on Space-Time Methods for PDEs
RICAM, Linz, Nov. 7-11, 2016

Table of Contents

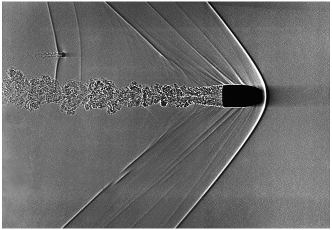
Table of Contents

Navier-Stokes Equations

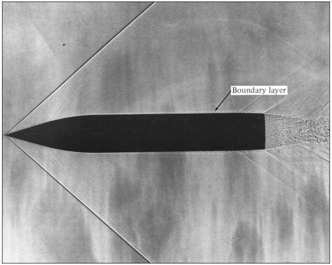
Numerical Challenges

Robust simulation of unsteady fluid dynamics remains a challenging issue.

- Resolving solution features (sharp, localized viscous-scale phenomena)
  - Shocks
  - Boundary layers - resolution needed for drag/load
  - Turbulence (non-localized)
- Stability of numerical schemes
  - Nonlinearity
  - Nature of PDE changes for different flow regimes
  - Coarse/adaptive grids
  - Higher order



Shock

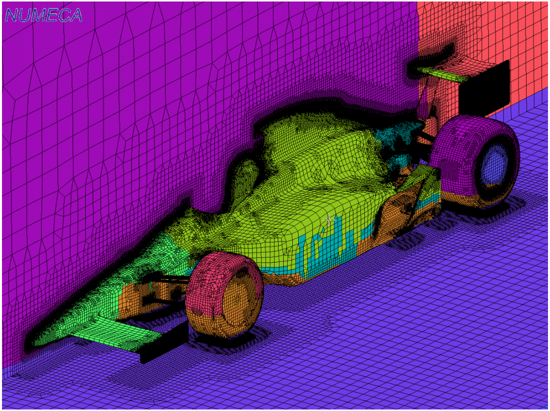


Boundary layer

Motivation

Initial Mesh Design is Expensive and Time-Consuming

- Surface mesh must accurately represent geometry
- Volume mesh needs sufficient resolution for asymptotic regime
- Engineers often forced to work by trial and error
- Bad in the context of HPC

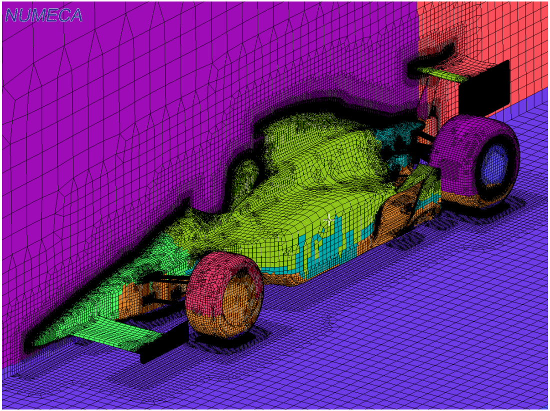


Formula 1 Mesh by Numeca

Motivation

Initial Mesh Design is Expensive and Time-Consuming

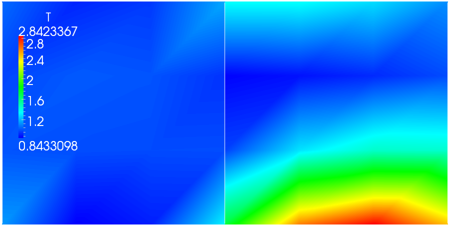
- Surface mesh must accurately represent geometry
- Volume mesh needs sufficient resolution for asymptotic regime
- Engineers often forced to work by trial and error
- Bad in the context of HPC
- We desire an automated computational technology



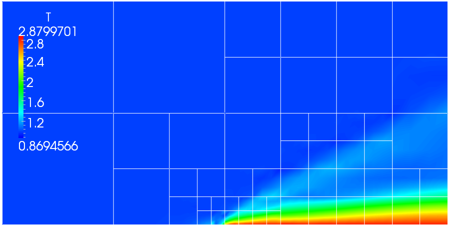
Formula 1 Mesh by Numeca

DPG on Coarse Meshes

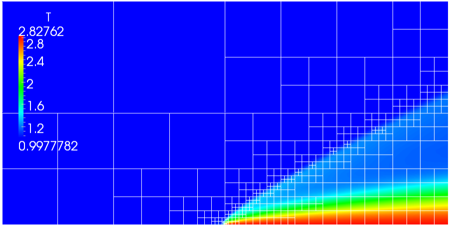
Adaptive Solve of the Carter Plate Problem[1] $Re = 1000$



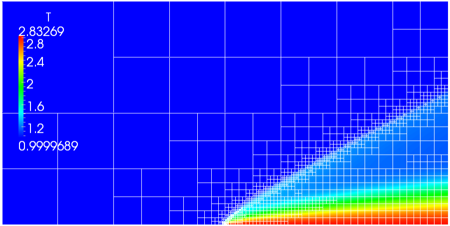
Temperature on Initial Mesh



Temperature after 4 Refinements



Temperature after 8 Refinements



Temperature after 11 Refinements

[1] J.L. Chan. ``A DPG Method for Convection-Diffusion Problems''. PhD thesis. University of Texas at Austin, 2013.

Lessons from Other Methods

Streamline Upwind Petrov-Galerkin: Adaptively changing the test space can produce a method with better stability.

Discontinuous Galerkin: Discontinuous basis functions are a legitimate option for finite element methods.

Hybridized DG: Mesh interface unknowns can facilitate static condensation -- reducing the number of DOFs in the global solve.

Least-Squares FEM: The finite element method is most powerful in a minimum residual context (i.e. as a Ritz method).

Space-Time FEM: Highly adaptive methods should have adaptive time integration. Superior framework for problems with moving boundaries. Requires a method that is both temporally and spatially stable.

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

THE UNIVERSITY OF TEXAS AT AUSTIN

DPG is a Minimum Residual Method

Find $u \in U$ such that

$$b(u, v) = l(v) \quad \forall v \in V$$

with operator $B : U \to V'$ defined by $b(u, v) = \langle Bu, v \rangle_{V' \times V}$.
This gives the operator equation

$$Bu = l \quad \in V' .$$

We wish to minimize the residual $Bu - l \in V'$:

$$u_h = \underset{w_h \in U_h}{\arg\min} \frac{1}{2} \left\| Bw_h - l \right\|_{V'}^2 .$$

Dual norms are not computationally tractable. Inverse Riesz map moves the residual to a more accessible space:

$$u_h = \underset{w_h \in U_h}{\arg\min} \frac{1}{2} \left\| R_V^{-1} (Bw_h - l) \right\|_V^2 .$$

Overview of DPG

Petrov-Galerkin with Optimal Test Functions

Taking the Gâteaux derivative to be zero in all directions $\delta u \in U_h$ gives,

$$\left(R_V^{-1}(Bu_h - l), R_V^{-1}B\delta u\right)_V = 0, \quad \forall \delta u \in U,$$

which by definition of the Riesz map is equivalent to

$$\left\langle Bu_h - l, R_V^{-1}B\delta u_h\right\rangle = 0 \quad \forall \delta u_h \in U_h\,,$$

with optimal test functions $v_{\delta u_h} := R_V^{-1}B\delta u_h$ for each trial function δu_h.

Resulting Petrov-Galerkin System

This gives a simple bilinear form

$$b(u_h, v_{\delta u_h}) = l(v_{\delta u_h}),$$

with $v_{\delta u_h} \in V$ that solves the auxiliary problem

$$(v_{\delta u_h}, \delta v)_V = \langle R_V v_{\delta u_h}, \delta v\rangle = \langle B\delta u_h, \delta v\rangle = b(\delta u_h, \delta v) \quad \forall \delta v \in V.$$

Identifying the error representation function:

$$\psi := R_V^{-1}(Bu_h - l)$$

allows us to develop an alternative interpretation of DPG.

DPG as a Mixed Problem

Find $\psi \in V$, $u_h \in U_h$ such that

$$\begin{aligned} (\psi, \delta v)_V - b(u_h, \delta v) &= -l(\delta v) \qquad && \forall \delta v \quad && \in V \\ b(\delta u_h, \psi) &= 0 && \forall \delta u_h && \in U_h \end{aligned}$$

In this unconventional saddle-point problem, the approximate solution u_h comes from a finite-dimensional trial space and plays the role of the Lagrange multiplier for the error representation function

Overview of DPG

DPG is the Most Stable Petrov-Galerkin Method

Babuška's theorem guarantees that *discrete stability and approximability imply convergence*. If bilinear form $b(u, v)$, with $M := \|b\|$ satisfies the discrete inf-sup condition with constant γ_h,

$$\sup_{v_h \in V_h} \frac{|b(u, v)|}{\|v_h\|_V} \geq \gamma_h \|u_h\|_U ,$$

then the Galerkin error satisfies the bound

$$\|u_h - u\|_U \leq \frac{M}{\gamma_h} \inf_{w_h \in U_h} \|w_h - u\|_U .$$

Optimal test function realize the supremum guaranteeing that $\gamma_h \geq \gamma$.

Energy Norm

If we use the energy norm, $\|u\|_E := \|Bu\|_{V'}$ in the error estimate, then $M = \gamma = 1$. Babuška's theorem implies that the minimum residual method is the most stable Petrov-Galerkin method (assuming exact optimal test functions).

Overview of DPG

Other Features

Discontinuous Petrov-Galerkin

- Continuous test space produces global solve for optimal test functions
- Discontinuous test space results in an embarrassingly parallel solve

Hermitian Positive Definite Stiffness Matrix

Property of all minimum residual methods

$$b(u_h, v_{\delta u_h}) = (v_{u_h}, v_{\delta u_h})_V = \overline{(v_{\delta u_h}, v_{u_h})_V} = \overline{b(\delta u_h, v_{u_h})}$$

Error Representation Function

Energy norm of Galerkin error (residual) can be computed without exact solution

$$\|u_h - u\|_E = \|B(u_h - u)\|_{V'} = \|Bu_h - l\|_{V'} = \left\|R_V^{-1}(Bu_h - l)\right\|_V$$

Overview of DPG

High Performance Computing

Eliminates human intervention

- Stability
- Robustness
- Adaptivity
- Automaticity
- Compute intensive
- Embarrassingly parallel local solves
- Factorization recyclable
- Low communication
- SPD stiffness matrix
- Multiphysics

Stampede Supercomputer at TACC

Mira Supercomputer at Argonne

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

THE UNIVERSITY OF TEXAS AT AUSTIN

DPG for Convection-Diffusion

Start with the strong-form PDE.

$$\nabla \cdot (\boldsymbol{\beta} u) - \epsilon \Delta u = g$$

Rewrite as a system of first-order equations.

$$\frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u = \mathbf{0}$$
$$\nabla \cdot (\boldsymbol{\beta} u - \boldsymbol{\sigma}) = g$$

Multiply by test functions and integrate by parts over each element, K.

$$\frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle u, \tau_n \rangle_{\partial K} = 0$$
$$-(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \langle (\boldsymbol{\beta} u - \boldsymbol{\sigma}) \cdot \mathbf{n}, v \rangle_{\partial K} = (g, v)_K$$

Use the ultraweak (DPG) formulation to obtain bilinear form $b(u, v) = l(v)$.

$$\frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle \hat{u}, \tau_n \rangle_{\partial K}$$
$$- (\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \langle \hat{t}, v \rangle_{\partial K} = (g, v)_K$$

Locally Conservative DPG

Local Conservation for Convection-Diffusion

The local conservation law in convection diffusion is

$$\int_{\partial K} \hat{t} = \int_K g \,,$$

which is equivalent to having $\mathbf{v}_K := \{v, \boldsymbol{\tau}\} = \{1_K, \mathbf{0}\}$ in the test space. In general, this is not satisfied by the optimal test functions. Following Moro et al[2] (also Chang and Nelson[3]), we can enforce this condition with Lagrange multipliers:

$$L(u_h, \boldsymbol{\lambda}) = \frac{1}{2} \left\| R_V^{-1}(Bu_h - l) \right\|_V^2 - \sum_K \lambda_K \underbrace{\langle Bu_h - l, \mathbf{v}_K \rangle}_{\langle \hat{t}, 1_K \rangle_{\partial K} - \langle g, 1_K \rangle_K} \,,$$

where $\boldsymbol{\lambda} = \{\lambda_1, \cdots, \lambda_N\}$.

[2] D. Moro, N.C. Nguyen, and J. Peraire. ``A Hybridized Discontinuous Petrov-Galerkin Scheme for Scalar Conservation Laws''. In: *Int. J. Num. Meth. Eng.* (2011).

[3] C.L. Chang and J.J. Nelson. ``Least-Squares Finite Element Method for the Stokes Problem with Zero Residual of Mass Conservation''. In: *SIAM J. Num. Anal.* 34 (1997), pp. 480–489.

Finding the critical points of $L(u, \boldsymbol{\lambda})$, we get the following equations.

Locally Conservative Saddle Point System

$$\frac{\partial L(u_h, \boldsymbol{\lambda})}{\partial u_h} = b(u_h, R_V^{-1} B\delta u_h) - l(R_V^{-1} B\delta u_h) - \sum_K \lambda_K b(\delta u_h, \mathbf{v}_K) = 0 \quad \forall \delta u_h \in U_h$$

$$\frac{\partial L(u_h, \boldsymbol{\lambda})}{\partial \lambda_K} = -b(u_h, \mathbf{v}_K) + l(\mathbf{v}_K) = 0 \quad \forall K$$

A few consequences:

- Minimization problem turns into a constrained minimization problem.
- Optimal test function are in the orthogonal complement of constants.

Locally Conservative DPG

THE UNIVERSITY OF TEXAS AT AUSTIN

Optimal Test Functions

For each $\mathbf{u} = \{u, \boldsymbol{\sigma}, \hat{u}, \hat{t}\} \in \mathbf{U}_h$, find $\mathbf{v_u} = \{v_\mathbf{u}, \boldsymbol{\tau}_\mathbf{u}\} \in \mathbf{V}$ such that

$$(\mathbf{v_u}, \mathbf{w})_\mathbf{V} = b(\mathbf{u}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}$$

where $\mathbf{V}$ becomes $\mathbf{V}_{p+\Delta p}$ in order to make this computationally tractable. We recently developed this modification to the *robust test norm*[4] which behaves better in the presence of singularities.

Convection-Diffusion Test Norm

$$\|(v, \boldsymbol{\tau})\|^2_{\mathbf{V},\Omega_h} = \left\| \min\left\{ \frac{1}{\sqrt{\epsilon}}, \frac{1}{\sqrt{|K|}} \right\} \boldsymbol{\tau} \right\|^2 + \|\nabla \cdot \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v\|^2$$
$$+ \|\boldsymbol{\beta} \cdot \nabla v\|^2 + \epsilon \|\nabla v\|^2 \quad \underbrace{+ \|v\|^2}_{\text{No longer necessary}}$$

[4] J. Chan et al. ``A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms''. In: *Comp. Math. Appl.* 67.4 (2014), pp. 771 –795.

Locally Conservative DPG

Optimal Test Functions

For each $\mathbf{u} = \{u, \boldsymbol{\sigma}, \hat{u}, \hat{t}\} \in \mathbf{U}_h$, find $\mathbf{v_u} = \{v_\mathbf{u}, \boldsymbol{\tau}_\mathbf{u}\} \in \mathbf{V}$ such that

$$(\mathbf{v_u}, \mathbf{w})_\mathbf{V} = b(\mathbf{u}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}$$

where $\mathbf{V}$ becomes $\mathbf{V}_{p+\Delta p}$ in order to make this computationally tractable. We recently developed this modification to the *robust test norm*[4] which behaves better in the presence of singularities.

Convection-Diffusion Test Norm

$$\|(v, \boldsymbol{\tau})\|^2_{\mathbf{V},\Omega_h} = \left\| \min\left\{\frac{1}{\sqrt{\epsilon}}, \frac{1}{\sqrt{|K|}}\right\} \boldsymbol{\tau} \right\|^2 + \|\nabla \cdot \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v\|^2 + \|\boldsymbol{\beta} \cdot \nabla v\|^2 + \epsilon \|\nabla v\|^2 + \underbrace{\left(\frac{1}{|K|}\int_K v\right)^2}_{\text{Scaling term}}$$

[4] J. Chan et al. ``A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms''. In: *Comp. Math. Appl.* 67.4 (2014), pp. 771 –795.

Locally Conservative DPG

Stability and Robustness Analysis[5]

- We follow Brezzi's theory for an abstract mixed problem:

$$\begin{cases} \boldsymbol{u} \in \boldsymbol{U}, p \in Q \\ a(\boldsymbol{u}, \boldsymbol{w}) + c(p, \boldsymbol{w}) & = l(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{U} \\ c(q, \boldsymbol{u}) & = g(q) \quad \forall q \in Q \end{cases}$$

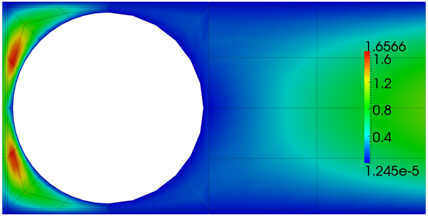
  where a, c, l, g denote the appropriate bilinear and linear forms.
- $a(\boldsymbol{u}, \boldsymbol{w}) = b(\boldsymbol{u}, R_V^{-1} B\boldsymbol{w}) = (R_V^{-1} B\boldsymbol{u}, R_V^{-1} B\boldsymbol{w})_V$
- $c(p, \boldsymbol{w}) = \sum_K \lambda_K \langle \hat{t}, 1_K \rangle_{\partial K}$
- Locally conservative DPG satisfies inf-sup and inf-sup in kernel conditions.
- Robustness is proved by switching to energy norm in Brezzi analysis.

[5] T.E. Ellis, L.F. Demkowicz, and J.L. Chan. ``Locally Conservative Discontinuous Petrov-Galerkin Finite Elements For Fluid Problems''. In: *Comp. Math. Appl.* 68.11 (2014), pp. 1530 –1549.

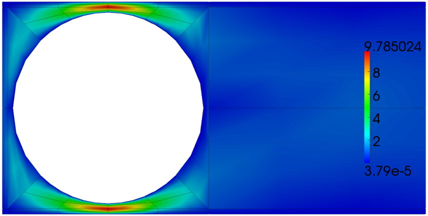
Numerical Experiments

Stokes Flow Around a Cylinder

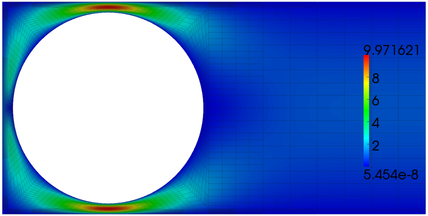
Horizontal Velocity



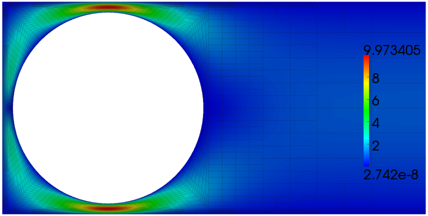
1 Refinement



1 Refinement



6 Refinements

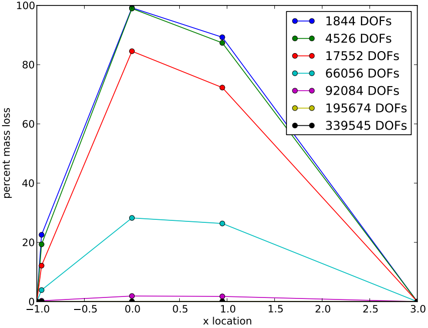


6 Refinements

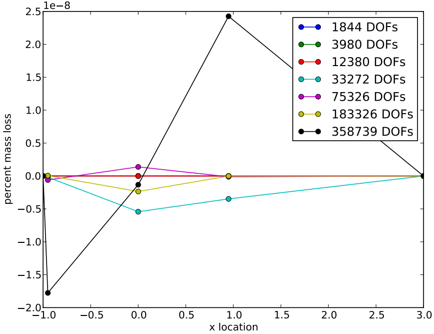
Nonconservative

Conservative

Percent Mass Loss at $x = [-1, -0.95, 0, 0.95, 3]$



Nonconservative

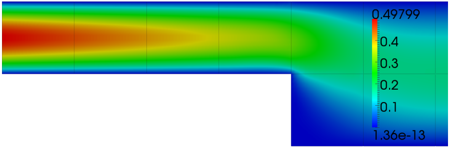


Conservative

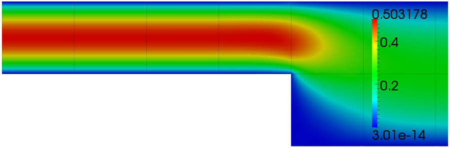
Numerical Experiments

Stokes Flow Over a Backward Facing Step

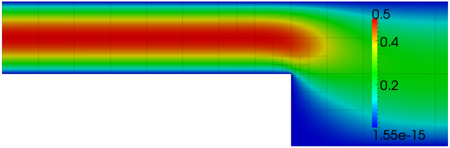
Horizontal Velocity



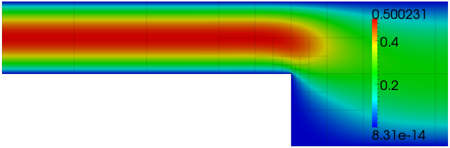
Initial Mesh



Initial Mesh



8 Refinements



8 Refinements

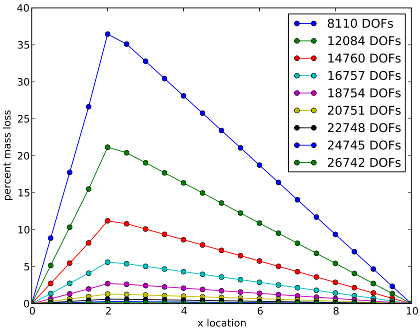
Nonconservative

Conservative

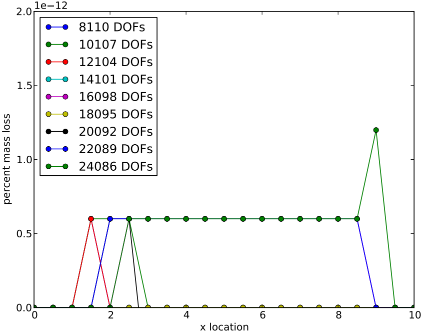
Numerical Experiments

Stokes Flow Over a Backward Facing Step

Percent Mass Loss at $x = [0, 0.5, \cdots, 9.5, 10]$



Nonconservative



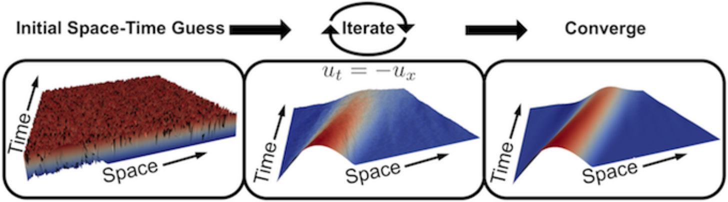
Conservative

Table of Contents

Space-Time DPG

Extending DPG to Transient Problems

- Time stepping techniques are not ideally suited to highly adaptive grids
- Space-time FEM proposed as a solution
  - ✓ Unified treatment of space and time
  - ✓ Local space-time adaptivity (local time stepping)
  - ✓ Parallel-in-time integration (space-time multigrid)
  - ✗ Spatially stable FEM methods may not be stable in space-time
  - ✗ Need to support higher dimensional problems
- DPG provides necessary stability and adaptivity



Courtesy of XBraid by LLNL

Equation is parabolic in space-time.

$$\frac{\partial u}{\partial t} + \boldsymbol{\beta} \cdot \nabla u - \epsilon \Delta u = f$$

This is just a composition of a constitutive law and conservation of mass.

$$\boldsymbol{\sigma} - \epsilon \nabla u = 0$$

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\beta} u - \boldsymbol{\sigma}) = f$$

We can rewrite this in terms of a space-time divergence.

$$\frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u = 0$$

$$\nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix} = f$$

Space-Time DPG for Convection-Diffusion

Ultra-Weak Formulation with Discontinuous Test Functions

Multiply by test function and integrate by parts over space-time element K.

$$\left(\frac{1}{\epsilon}\boldsymbol{\sigma},\boldsymbol{\tau}\right)_K+\left(u,\nabla\cdot\boldsymbol{\tau}\right)_K-\left\langle\hat{u},\boldsymbol{\tau}\cdot\boldsymbol{n}_x\right\rangle_{\partial K}=0$$

$$-\left(\begin{pmatrix}\boldsymbol{\beta}u-\boldsymbol{\sigma}\\ u\end{pmatrix},\nabla_{xt}v\right)_K+\left\langle\hat{t},v\right\rangle_{\partial K}=f$$

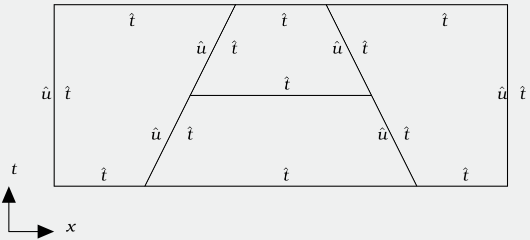
where

$$\hat{u}:=\operatorname{tr}(u)$$

$$\hat{t}:=\operatorname{tr}(\boldsymbol{\beta}u-\boldsymbol{\sigma})\cdot\boldsymbol{n}_x+\operatorname{tr}(u)\cdot n_t$$

- Trace $\hat{u}$ defined on spatial boundaries
- Flux $\hat{t}$ defined on all boundaries

Support of Trace Variables



Space-Time Convection-Diffusion

L^2 Equivalent Norms

Bilinear form with group variables:

$$b\left(\left(u,\hat{u}\right),v\right)=\left(u,A_h^*v\right)_{L^2(\Omega_h)}+\langle\widehat{u},[\![v]\!]\rangle_{\Gamma_h}$$

For conforming v^* satisfying $A^*v^*=u$

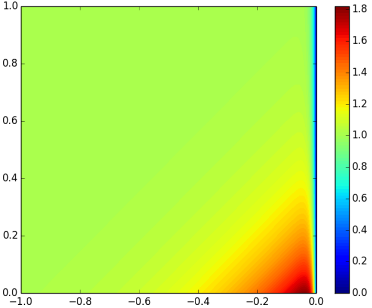
$$\|u\|^2_{L^2(\Omega_h)}=b(u,v^*)=\frac{b(u,v^*)}{\|v^*\|_V}\|v^*\|_V$$
$$\leq\sup_{v^*\neq 0}\frac{|b(u,v^*)|}{\|v^*\|}\|v^*\|=\|u\|_E\|v^*\|_V$$

Necessary robustness condition:

$$\|v^*\|_V\lesssim\|u\|_{L^2(\Omega_h)}$$
$$\Rightarrow\|u\|_{L^2(\Omega_h)}\lesssim\|u\|_E$$

Analytical Solution

$$e^{-lt}(e^{\lambda_1(x-1)}-e^{\lambda_2(x-1)})$$
$$+\left(1-e^{\frac{1}{\epsilon}x}\right)$$

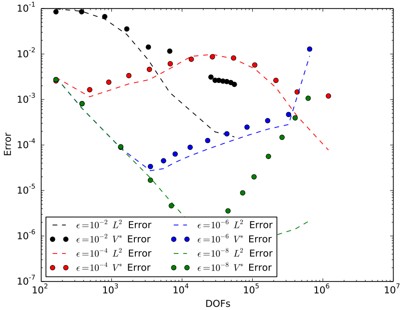


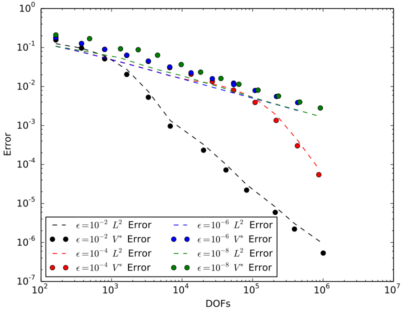
Space-Time Convection-Diffusion

THE UNIVERSITY OF TEXAS AT AUSTIN

L^2 Equivalent Norms

A norm should be: bounded by $\|u\|_{L^2(\Omega_h)}$, have good conditioning, not produce boundary layers in the optimal test function.





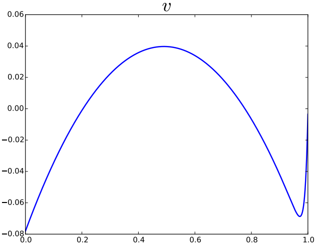
$$\|(v,\boldsymbol{\tau})\|^2 = \left\|\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|^2 + \left\|\frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v\right\|^2 + \|v\|^2 + \|\boldsymbol{\tau}\|^2$$

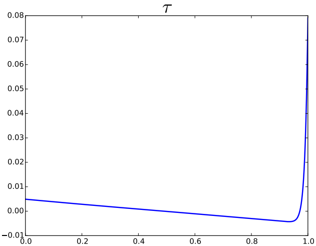
$$\|(v,\boldsymbol{\tau})\|^2 = \left\|\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|^2 + \min\left(\frac{1}{h^2},\frac{1}{\epsilon}\right)\|\boldsymbol{\tau}\|^2 + \epsilon\|\nabla v\|^2 + \|\boldsymbol{\beta}\cdot\nabla v\|^2 + \|v\|^2$$

Steady Convection-Diffusion

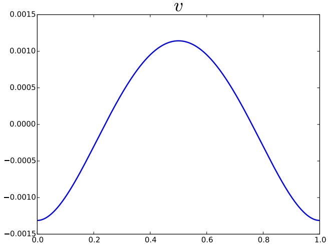
Ideal Optimal Shape Functions

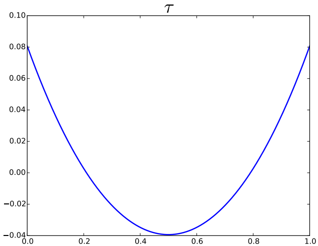
Graph Norm





Coupled Robust Norm

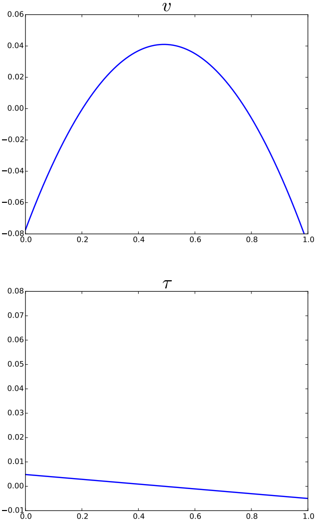




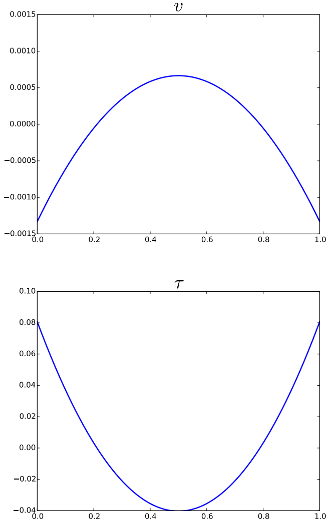
Steady Convection-Diffusion

Approximated ($p = 3$) Optimal Shape Functions

Graph Norm



Coupled Robust Norm



The following norms are robust for steady convection-diffusion.

The robust norm was derived in[6]:

$$\|(v,\boldsymbol{\tau})\|^2 = \|\boldsymbol{\beta}\cdot\nabla v\|^2 + \epsilon\,\|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2}, 1\right)\|v\|^2 + \|\nabla\cdot\boldsymbol{\tau}\|^2 + \min\left(\frac{1}{h^2}, \frac{1}{\epsilon}\right)\|\boldsymbol{\tau}\|^2 .$$

The case for the coupled robust norm was made in[7]:

$$\|(v,\boldsymbol{\tau})\|^2 = \|\boldsymbol{\beta}\cdot\nabla v\|^2 + \epsilon\,\|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2}, 1\right)\|v\|^2 + \|\nabla\cdot\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v\|^2 + \min\left(\frac{1}{h^2}, \frac{1}{\epsilon}\right)\|\boldsymbol{\tau}\|^2 .$$

[6] J. Chan et al. ``A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms''. In: *Comp. Math. Appl.* 67.4 (2014), pp. 771 –795.

[7] J.L. Chan. ``A DPG Method for Convection-Diffusion Problems''. PhD thesis. University of Texas at Austin, 2013.

Space-Time Convection-Diffusion

Two Robust Norms for Transient Convection-Diffusion

Let $\tilde{\boldsymbol{\beta}} := \begin{pmatrix} \boldsymbol{\beta} \\ 1 \end{pmatrix}$ and $\nabla_{xt} v := \begin{pmatrix} \nabla v \\ \frac{\partial v}{\partial t} \end{pmatrix}$.

The following norms are robust for space-time convection-diffusion.

Robust Norm:

$$\|(v, \boldsymbol{\tau})\|^2 = \left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v\right\|^2 + \epsilon \|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2}, 1\right) \|v\|^2 + \|\nabla \cdot \boldsymbol{\tau}\|^2 + \min\left(\frac{1}{h^2}, \frac{1}{\epsilon}\right) \|\boldsymbol{\tau}\|^2 \,.$$

Coupled Robust Norm

$$\|(v, \boldsymbol{\tau})\|^2 = \left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v\right\|^2 + \epsilon \|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2}, 1\right) \|v\|^2 + \left\|\nabla \cdot \boldsymbol{\tau} - \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v\right\|^2 + \min\left(\frac{1}{h^2}, \frac{1}{\epsilon}\right) \|\boldsymbol{\tau}\|^2 \,.$$

Consider the problem with homogeneous boundary conditions

$$
\begin{aligned}
\frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\
\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} u - \nabla \cdot \boldsymbol{\sigma} &= f \\
\beta_n u - \epsilon \frac{\partial u}{\partial n} &= 0 \text{ on } \Gamma_- \\
u &= 0 \text{ on } \Gamma_+ \\
u &= u_0 \text{ on } \Gamma_0 .
\end{aligned}
$$

The adjoint operator A^* is given by

$$
A^*(v, \boldsymbol{\tau}) = \left(\frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v, -\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v + \nabla \cdot \boldsymbol{\tau} \right) .
$$

Robust Norms for Transient Convection-Diffusion

Controlling Different Field Variables

We decompose the continuous adjoint problem $A^*(\boldsymbol{\tau}, v) = (\boldsymbol{\sigma}, u)$ into

Continuous part with forcing u

$$\frac{1}{\epsilon}\boldsymbol{\tau}_1 + \nabla v_1 = 0$$
$$-\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \nabla \cdot \boldsymbol{\tau}_1 = u$$

$$\boldsymbol{\tau}_1 \cdot \boldsymbol{n}_x = 0 \text{ on } \Gamma_-$$
$$v_1 = 0 \text{ on } \Gamma_+$$
$$v_1 = 0 \text{ on } \Gamma_T$$

Continuous part with forcing $\boldsymbol{\sigma}$

$$\frac{1}{\epsilon}\boldsymbol{\tau}_2 + \nabla v_2 = \boldsymbol{\sigma}$$
$$-\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2 + \nabla \cdot \boldsymbol{\tau}_2 = 0$$

$$\boldsymbol{\tau}_2 \cdot \boldsymbol{n}_x = 0 \text{ on } \Gamma_-$$
$$v_2 = 0 \text{ on } \Gamma_+$$
$$v_2 = 0 \text{ on } \Gamma_T$$

Robust Norms for Transient Convection-Diffusion

Proved Bounds at Our Disposal

Proofs of these lemmas can be found in[8].

Lemma (1)

If $\nabla \cdot \boldsymbol{\beta} = 0$*, we can bound*

$$\|v\|^2 + \epsilon \|\nabla v\|^2 \leq \|u\|^2 + \epsilon \|\boldsymbol{\sigma}\|^2$$

where $v = v_1 + v_2$.

Lemma (2)

If $\left\|\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta} \boldsymbol{I}\right\|_{L^\infty} \leq C_{\boldsymbol{\beta}}$*, we can bound*

$$\left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1\right\| \lesssim \|u\| .$$

[8] T. Ellis, J. Chan, and L. Demkowicz. ``Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations,x Eds. G.R. Barrenechea et al.'' In: vol. 114. Lecture Notes in Computational Science and Engineering. in print, see also ICES Report 2015/21. Springer, 2016. Chap. Robust DPG Methods for Transient Convection-Diffusion.

THE UNIVERSITY OF TEXAS AT AUSTIN

Bound on $\|(v_1, \boldsymbol{\tau}_1)\|$

Lemma (2) $\Rightarrow$ $$\left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1\right\| \lesssim \|u\|$$

Lemma (2) $\Rightarrow$ $$\|\nabla \cdot \boldsymbol{\tau}_1\| \leq \|u\| + \left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1\right\| \lesssim 2\|u\|$$

Lemma (2) $\Rightarrow$ $$\left\|\nabla \cdot \boldsymbol{\tau}_1 - \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1\right\| = \|u\|$$

Lemma (1) $\Rightarrow$ $$\|v_1\|^2 + \epsilon \|\nabla v_1\|^2 \leq \|u\|^2$$

Lemma (1) $\Rightarrow$ $$\frac{1}{\epsilon}\|\boldsymbol{\tau}_1\| = \epsilon \|\nabla v_1\| \leq \|u\|$$

We can guarantee robust control

$$\|(u, 0)\|_{L^2(\Omega_h)} \lesssim \|(u, \boldsymbol{\sigma})\|_E \, .$$

Bound on $\|(v_2, \boldsymbol{\tau}_2)\|$

Definition $\Rightarrow$ $\left\|\nabla \cdot \boldsymbol{\tau}_2 - \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2\right\| = 0 \leq \|\boldsymbol{\sigma}\|$

Lemma (1) $\Rightarrow$ $\|v_2\|^2 + \epsilon \|\nabla v_2\|^2 \leq \epsilon \|\boldsymbol{\sigma}\|^2$

Lemma (1) $\Rightarrow$ $\frac{1}{\epsilon}\|\boldsymbol{\tau}_2\| = \|\boldsymbol{\sigma}\| + \epsilon \|\nabla v_2\| = (1+\epsilon)\|\boldsymbol{\sigma}\|$

We have not been able to prove bounds on $\left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2\right\|$ or $\|\nabla \cdot \boldsymbol{\tau}_2\|$.

We can **not** guarantee robust control

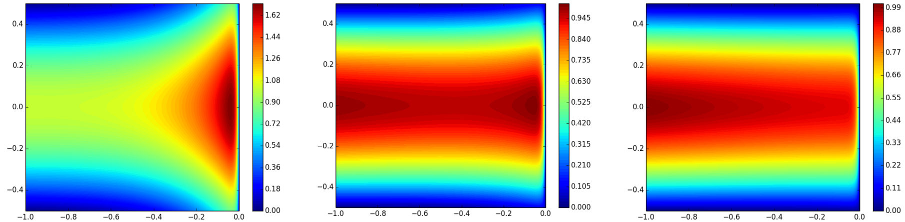
$$\|(0, \boldsymbol{\sigma})\|_{L^2(\Omega_h)} \not\lesssim \|(u, \boldsymbol{\sigma})\|_E \ .$$

Robust Norms for Transient Convection-Diffusion

Transient Analytical Solution

Transient impulse decays to Eriksson-Johnson steady state solution.

$$u = \exp(-lt)\left[\exp(\lambda_1 x) - \exp(\lambda_2 x)\right] + \cos(\pi y)\frac{\exp(s_1 x) - \exp(r_1 x)}{\exp(-s_1) - \exp(-r_1)}$$

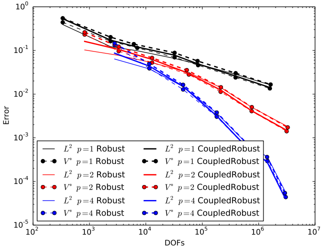


$t = 0.0$ $t = 0.5$ $t = 1.0$

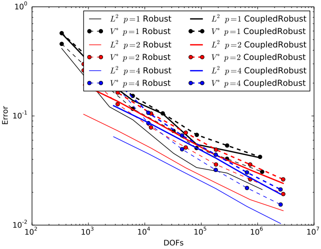
Robust Norms for Transient Convection-Diffusion

THE UNIVERSITY OF TEXAS AT AUSTIN

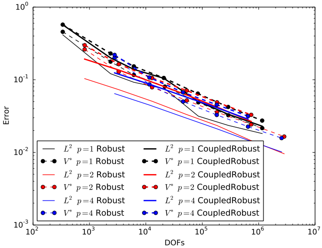
Robust Convergence to Analytical Solution



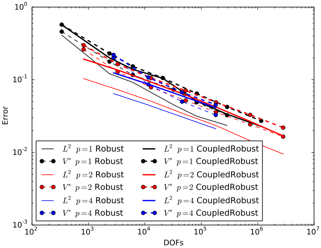
$\epsilon = 10^{-2}$



$\epsilon = 10^{-4}$



$\epsilon = 10^{-6}$



$\epsilon = 10^{-8}$

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

Space-time divergence form:

$$\frac{1}{\nu}\mathbb{D} - \nabla \boldsymbol{u} = 0$$

$$\nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I} \\ \boldsymbol{u} \end{pmatrix} = \boldsymbol{f}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

Multiply by $\mathbb{S} \in \mathbb{H}(\text{div}, Q)$, $\boldsymbol{v} \in \boldsymbol{H}^1_{xt}(Q)$, $q \in H^1(Q)$, and integrate by parts:

$$\left(\frac{1}{\nu}\mathbb{D}, \mathbb{S}\right) + (\boldsymbol{u}, \nabla \cdot \mathbb{S}) - \langle \hat{\boldsymbol{u}}, \mathbb{S} \cdot \boldsymbol{n}_x \rangle = 0$$

$$-\left(\begin{pmatrix} \boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I} \\ \boldsymbol{u} \end{pmatrix}, \nabla_{xt}\boldsymbol{v}\right) + \langle \hat{\boldsymbol{t}}, \boldsymbol{v} \rangle = (\boldsymbol{f}, \boldsymbol{v})$$

$$-(\boldsymbol{u}, \nabla q) + \langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, q \rangle = 0$$

Recall the adjoint and robust norm for convection-diffusion:

$$\left(\boldsymbol{\sigma}, \frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v\right) + \left(u, \nabla \cdot \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v - \frac{\partial v}{\partial t}\right)$$

$$\begin{aligned}\|(v, \boldsymbol{\tau})\|_{V,K}^2 := &\left\|\boldsymbol{\beta} \cdot \nabla v + \frac{\partial v}{\partial t}\right\|_K^2 + \epsilon \|\nabla v\|_K^2 + \min\left(\frac{\epsilon}{h^2}, 1\right)\|v\|_K^2 \\ &+ \|\nabla \cdot \boldsymbol{\tau}\|_K^2 + \min\left(\frac{1}{\epsilon}, \frac{1}{h^2}\right)\|\boldsymbol{\tau}\|_K^2\end{aligned}$$

For incompressible Navier-Stokes the adjoint comes from:

$$\begin{aligned}&\left(\Delta\mathbb{D}, \frac{1}{\nu}\mathbb{S} + \nabla \boldsymbol{v}\right) + \left(\Delta \boldsymbol{u}, \nabla \cdot \mathbb{S} - \nabla q - \left(\tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v} + \tilde{\boldsymbol{u}} \cdot (\nabla \boldsymbol{v})^T + \frac{\partial \boldsymbol{v}}{\partial t}\right)\right) \\ &+ (p, -\nabla \cdot \boldsymbol{v})\end{aligned}$$

Norms for Navier-Stokes come from analogy

Convection-Diffusion		Navier-Stokes
ϵ	$\rightarrow$	ν
$\boldsymbol{\tau}$	$\rightarrow$	$\mathbb{S}$
∇v	$\rightarrow$	$\nabla \boldsymbol{v}$
$\nabla \cdot \boldsymbol{\tau}$	$\rightarrow$	$\nabla \cdot \mathbb{S} - \nabla q$
$\boldsymbol{\beta} \cdot \nabla v + \frac{\partial v}{\partial t}$	$\rightarrow$	$\tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v} + \tilde{\boldsymbol{u}} \cdot (\nabla \boldsymbol{v})^T + \frac{\partial \boldsymbol{v}}{\partial t}$

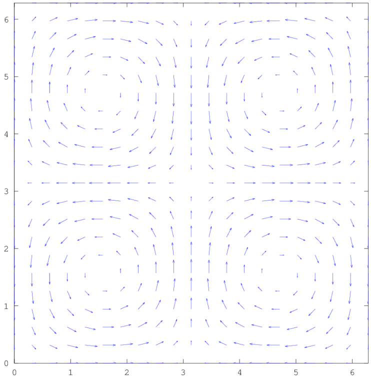
Robust norm:

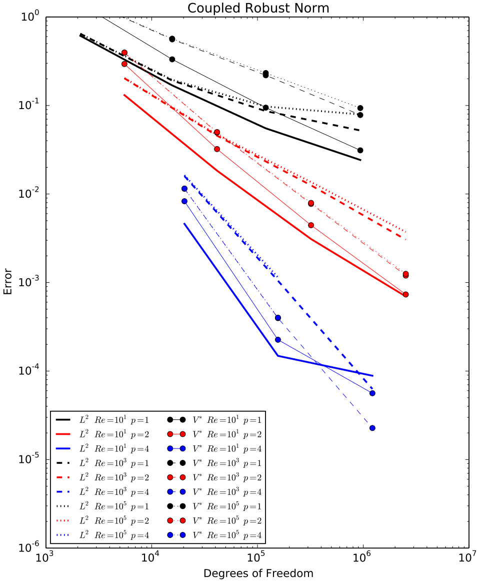
$$
\begin{aligned}
\|(\boldsymbol{v}, \mathbb{D}, q)\|_{V,K}^2 := & \left\| \tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v} + \tilde{\boldsymbol{u}} \cdot (\nabla \boldsymbol{v})^T + \frac{\partial \boldsymbol{v}}{\partial t} \right\|_K^2 + \nu \|\nabla \boldsymbol{v}\|_K^2 \\
& + \min\left(\frac{\nu}{h^2}, 1\right) \|\boldsymbol{v}\|_K^2 + \|\nabla \cdot \mathbb{S} - \nabla q\|_K^2 \\
& + \min\left(\frac{1}{\nu}, \frac{1}{h^2}\right) \|\mathbb{S}\|_K^2 + \|\nabla \cdot \boldsymbol{v}\|_K^2 + \|q\|_K^2
\end{aligned}
$$

Space-Time Incompressible Navier-Stokes

Taylor-Green Vortex

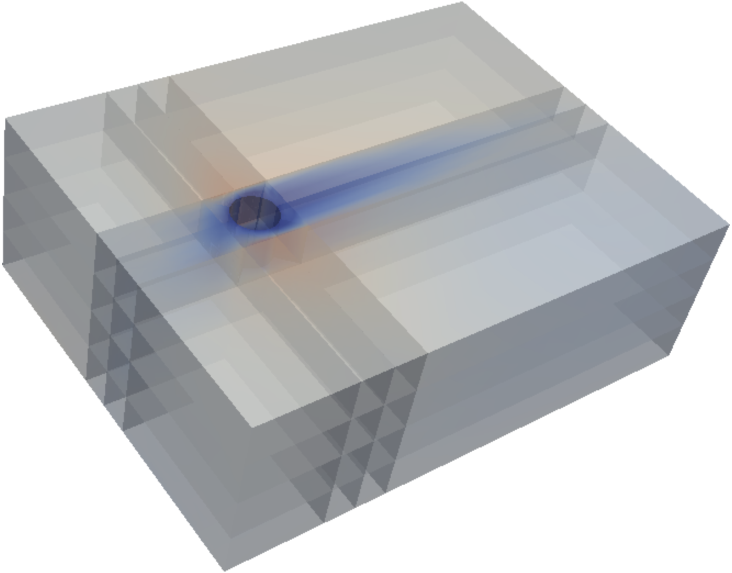
$$\boldsymbol{u} = e^{-\frac{2}{Re}t}\begin{pmatrix} \sin x \cos y \\ -\cos x \sin y \end{pmatrix}$$





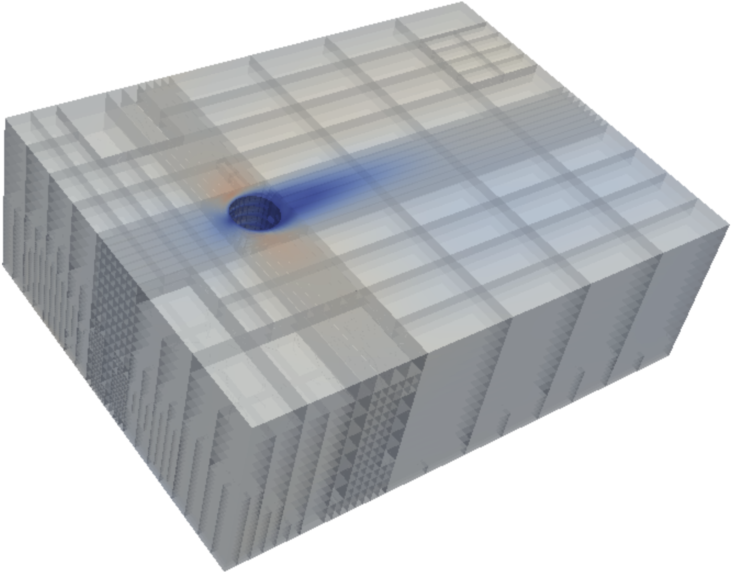
Space-Time Incompressible Navier-Stokes

Flow Over a Cylinder, Initial Mesh



Space-Time Incompressible Navier-Stokes

Flow Over a Cylinder, 4 Refinements



Space-Time Incompressible Navier-Stokes

Solve Times and Strong Scaling

Transient Flow Over a Cylinder

			1 Node	4 Nodes		32 Nodes	
Ref	Elems	DOFs	Time	Time	Scaling vs 1	Time	Scaling vs 4
0	80	31,304	1,772	453	3.91	451	1.01
1	605	225,908	8,190	3,574	2.29	717	4.98
2	3,013	1,081,598	32,008	12,076	2.65	2,648	4.56
3	9,726	3,429,384		28,744		6,319	4.54
4	11,742	4,144,674				8,510	

Computations on Lonestar, 1 node = 24 processors

32,008 seconds = 8.8 hours

28,744 seconds = 8.0 hours

8,510 seconds = 2.4 hours

Table of Contents

Space-Time Compressible Navier-Stokes

First Order System with Primitive Variables

Assuming Stokes hypothesis, ideal gas law, and constant viscosity:

$$\frac{1}{\mu}\mathbb{D} - \nabla \boldsymbol{u} = 0$$

$$\frac{Pr}{C_p \mu}\boldsymbol{q} + \nabla T = 0$$

$$\nabla_{xt} \cdot \begin{pmatrix} \rho \boldsymbol{u} \\ \rho \end{pmatrix} = f_c$$

$$\nabla_{xt} \cdot \begin{pmatrix} \rho \boldsymbol{u} \otimes \boldsymbol{u} + \rho R T \mathbb{I} - \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\ \rho \boldsymbol{u} \end{pmatrix} = \boldsymbol{f}_m$$

$$\nabla_{xt} \cdot \begin{pmatrix} \rho \boldsymbol{u}\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \rho R T \boldsymbol{u} + \boldsymbol{q} - \boldsymbol{u} \cdot \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\ \rho\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) \end{pmatrix} = f_e$$

Space-Time Compressible Navier-Stokes

Compact Notation

Conserved quantities

$$C_c := \rho$$

$$\boldsymbol{C}_m := \rho \boldsymbol{u}$$

$$C_e := \rho(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u})$$

Euler fluxes

$$\boldsymbol{F}_c := \rho \boldsymbol{u}$$

$$\mathbb{F}_m := \rho \boldsymbol{u} \otimes \boldsymbol{u} + \rho R T \mathbb{I}$$

$$\boldsymbol{F}_e := \rho \boldsymbol{u}\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \rho R T \boldsymbol{u}$$

Viscous fluxes

$$\boldsymbol{K}_c := \boldsymbol{0}$$

$$\mathbb{K}_m := \mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}$$

$$\boldsymbol{K}_e := -\boldsymbol{q} + \boldsymbol{u}\cdot\left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right)$$

Viscous variables

$$\mathbb{M}_{\mathbb{D}} := \mathbb{D}$$

$$\boldsymbol{M}_{\boldsymbol{q}} := \frac{Pr}{C_p}\boldsymbol{q}$$

$$\boldsymbol{G}_{\mathbb{D}} := 2\boldsymbol{u}$$

$$G_{\boldsymbol{q}} := -T$$

Use change of variables to get conservation or entropy variables.

Space-Time Compressible Navier-Stokes

Conservation Variables (Popular for Time-Stepping)

Change of variables:

$$\rho = \rho$$
$$\boldsymbol{m} = \rho \boldsymbol{u}$$
$$E = \rho \left(C_v T + \frac{1}{2} \boldsymbol{u} \cdot \boldsymbol{u} \right)$$

Euler fluxes:

$$\boldsymbol{F}_c^c = \boldsymbol{m}$$
$$\mathbb{F}_m^c = \frac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho} + (\gamma - 1) \left(E - \frac{\boldsymbol{m} \cdot \boldsymbol{m}}{2\rho} \right) \mathbb{I}$$
$$\boldsymbol{F}_e^c = \gamma E \frac{\boldsymbol{m}}{\rho} - (\gamma - 1) \frac{\boldsymbol{m} \cdot \boldsymbol{m}}{2\rho^2} \boldsymbol{m}$$

Space-Time Compressible Navier-Stokes

THE UNIVERSITY OF TEXAS AT AUSTIN

Entropy Variables (Symmetrize the Bubnov-Galerkin Stiffness Matrix)

Change of variables:

$$V_c = \frac{-E + (E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m})\left(\gamma + 1 - \ln\left[\frac{(\gamma-1)(E-\frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m})}{\rho^\gamma}\right]\right)}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}}$$

$$\boldsymbol{V}_m = \frac{\boldsymbol{m}}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}}$$

$$V_e = \frac{-\rho}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}}$$

Euler fluxes:

$$\boldsymbol{F}^e_c = \left[\frac{\gamma-1}{(-V_e)^\gamma}\right]^{\frac{1}{\gamma-1}} \exp\left[\frac{-\gamma + V_c - \frac{1}{2V_e}\boldsymbol{V}_m\cdot\boldsymbol{V}_m}{\gamma-1}\right]\boldsymbol{V}_m$$

$$\mathbb{F}^e_m = \left[\frac{\gamma-1}{(-V_e)^\gamma}\right]^{\frac{1}{\gamma-1}} \exp\left[\frac{-\gamma + V_c - \frac{1}{2V_e}\boldsymbol{V}_m\cdot\boldsymbol{V}_m}{\gamma-1}\right]\left(-\frac{\boldsymbol{V}_m\otimes\boldsymbol{V}_m}{V_e} + (\gamma-1)\mathbb{I}\right)$$

$$\boldsymbol{F}^e_e = \left[\frac{\gamma-1}{(-V_e)^\gamma}\right]^{\frac{1}{\gamma-1}} \exp\left[\frac{-\gamma + V_c - \frac{1}{2V_e}\boldsymbol{V}_m\cdot\boldsymbol{V}_m}{\gamma-1}\right]\frac{\boldsymbol{V}_m}{V_e}\left(\frac{1}{2V_e}\boldsymbol{V}_m\cdot\boldsymbol{V}_m - \gamma\right)$$

Space-Time Compressible Navier-Stokes

Define Group Variables

Group terms

$$C := \{C_c\,,\ \boldsymbol{C}_m\,,\ C_e\}$$
$$F := \{\boldsymbol{F}_c\,,\ \mathbb{F}_m\,,\ \boldsymbol{F}_e\}$$
$$K := \{\boldsymbol{K}_c\,,\ \mathbb{K}_m\,,\ \boldsymbol{K}_e\}$$
$$M := \{\mathbb{M}_{\mathbb{D}}\,,\ \boldsymbol{M}_{\boldsymbol{q}}\}$$
$$G := \{\boldsymbol{G}_{\mathbb{D}}\,,\ G_{\boldsymbol{q}}\}$$
$$f := \{f_c\,, \boldsymbol{f}_m\,, f_e\}$$

Group variables

$$W := \{\rho\,,\ \boldsymbol{u}\,,\ T\}$$
$$\hat{W} := \left\{2\hat{\boldsymbol{u}}\,,\ -\hat{T}\right\}$$
$$\Sigma := \{\mathbb{D}, \boldsymbol{q}\}$$
$$\hat{t} := \{\hat{t}_e\,,\ \hat{\boldsymbol{t}}_m,\ ,\ \hat{t}_e\}$$
$$\Psi := \{\mathbb{S}\,,\ \boldsymbol{\tau}\}$$
$$V := \{v_c\,,\ \boldsymbol{v}_m,\ ,\ v_e\}$$

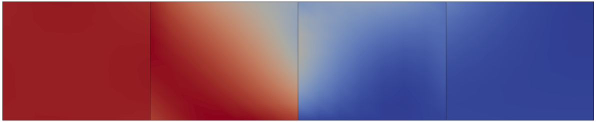
Navier-Stokes variational formulation is

$$\left(\frac{1}{\mu}M, \Psi\right) + (G, \nabla\cdot\Psi) - \left\langle \hat{W}, \Psi\cdot\boldsymbol{n}_x\right\rangle = 0$$
$$-\left(\begin{pmatrix} F-K \\ C \end{pmatrix}, \nabla_{xt}V\right) + \langle\hat{t}, V\rangle = (f, V)$$

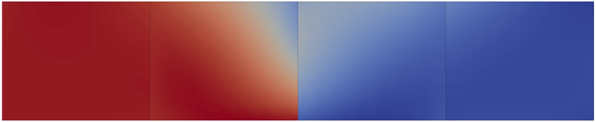
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

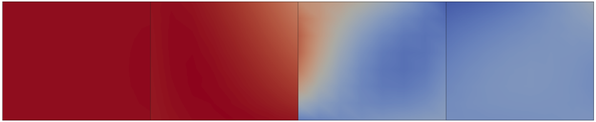
Mesh 1



Primitive Variables



Conservation Variables

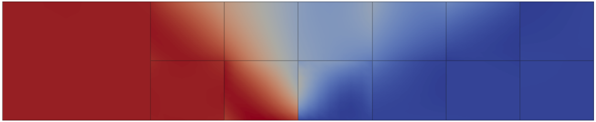


Entropy Variables

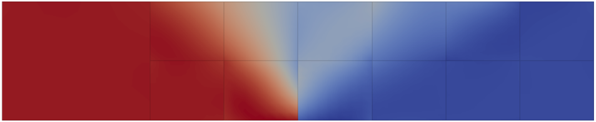
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

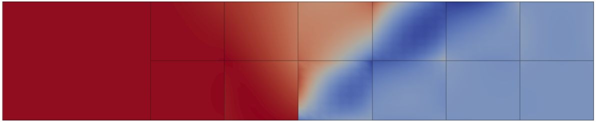
Mesh 2



Primitive Variables



Conservation Variables

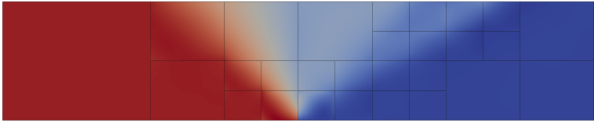


Entropy Variables

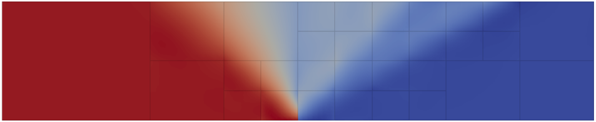
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

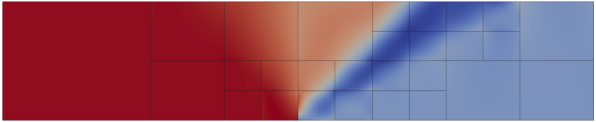
Mesh 3



Primitive Variables



Conservation Variables

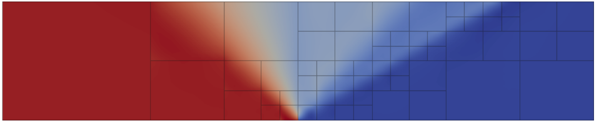


Entropy Variables

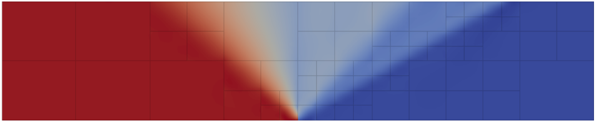
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

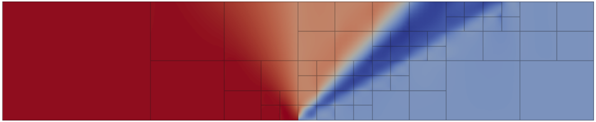
Mesh 4



Primitive Variables



Conservation Variables

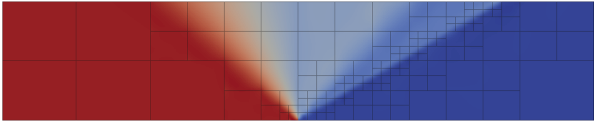


Entropy Variables

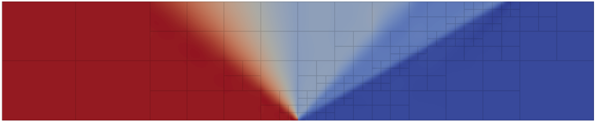
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

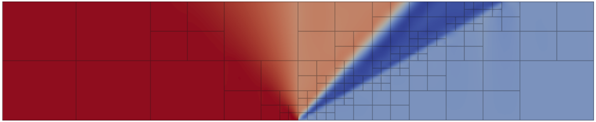
Mesh 5



Primitive Variables



Conservation Variables



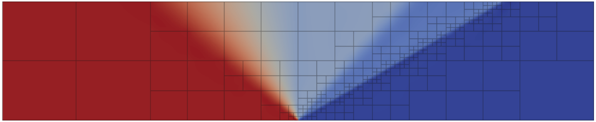
Entropy Variables

Space-Time Compressible Navier-Stokes

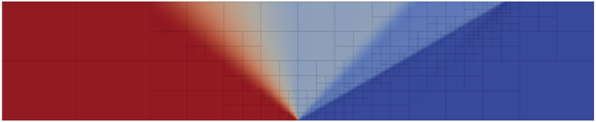
THE UNIVERSITY OF TEXAS AT AUSTIN

Sod Shock Tube with $\mu = 10^{-5}$

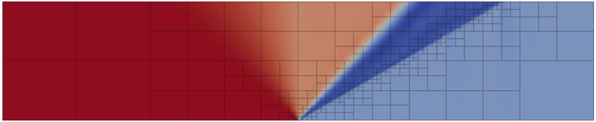
Mesh 6



Primitive Variables



Conservation Variables

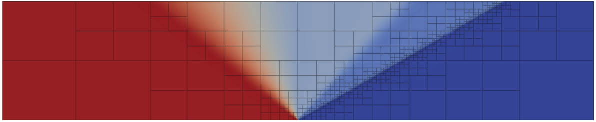


Entropy Variables

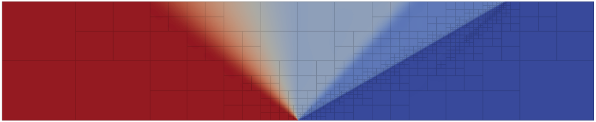
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

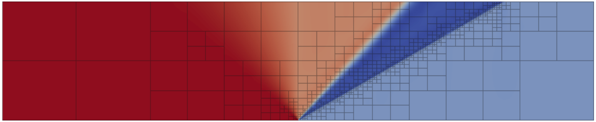
Mesh 7



Primitive Variables



Conservation Variables

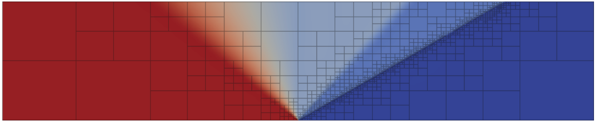


Entropy Variables

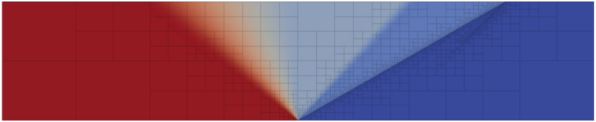
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

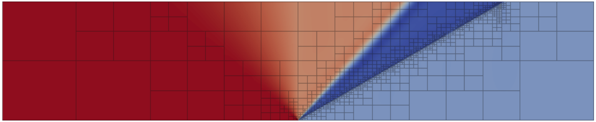
Mesh 8



Primitive Variables



Conservation Variables



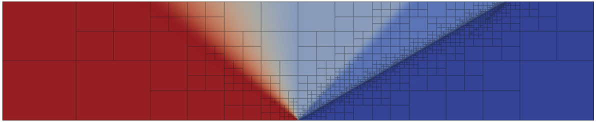
Entropy Variables

Space-Time Compressible Navier-Stokes

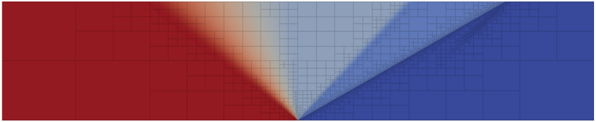
THE UNIVERSITY OF TEXAS AT AUSTIN

Sod Shock Tube with $\mu = 10^{-5}$

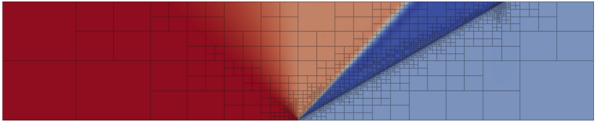
Mesh 9



Primitive Variables



Conservation Variables

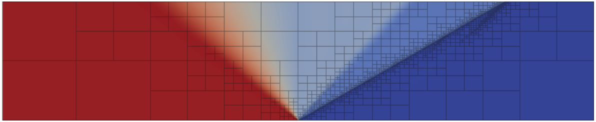


Entropy Variables

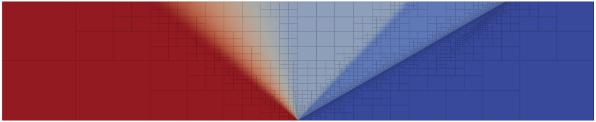
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

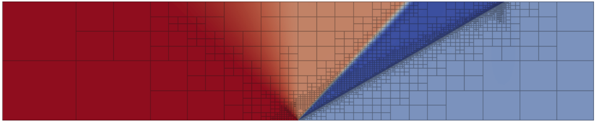
Mesh 10



Primitive Variables



Conservation Variables



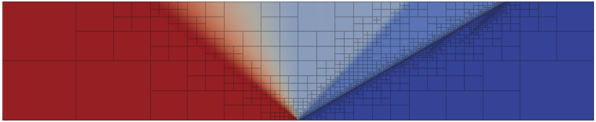
Entropy Variables

Space-Time Compressible Navier-Stokes

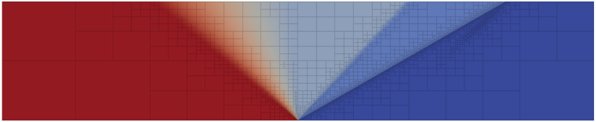
Sod Shock Tube with $\mu = 10^{-5}$

THE UNIVERSITY OF TEXAS AT AUSTIN

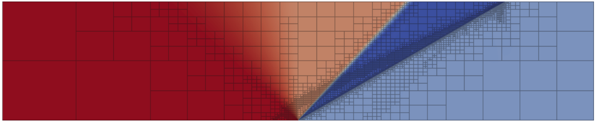
Mesh 11



Primitive Variables



Conservation Variables



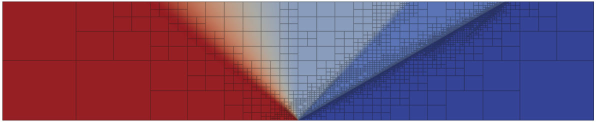
Entropy Variables

Space-Time Compressible Navier-Stokes

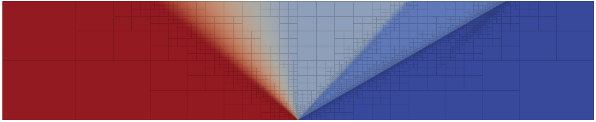
THE UNIVERSITY OF TEXAS AT AUSTIN

Sod Shock Tube with $\mu = 10^{-5}$

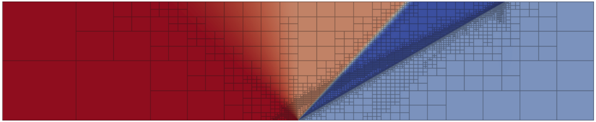
Mesh 12



Primitive Variables



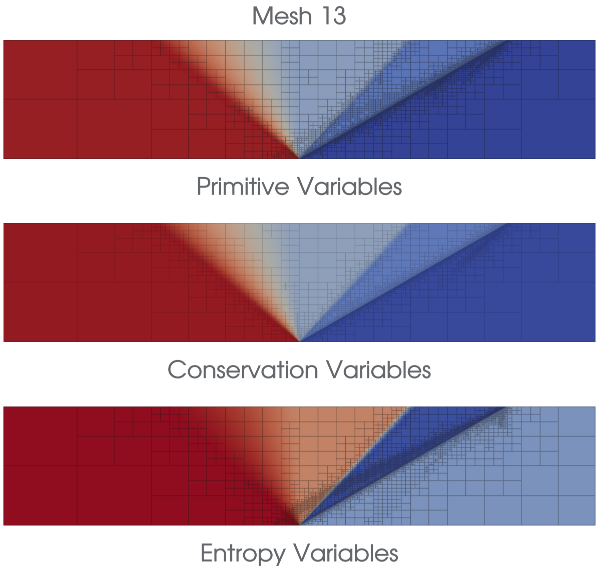
Conservation Variables



Entropy Variables

Space-Time Compressible Navier-Stokes

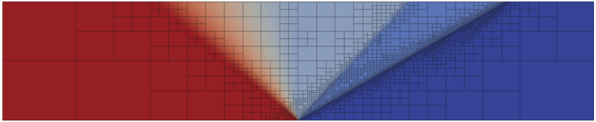
Sod Shock Tube with $\mu = 10^{-5}$



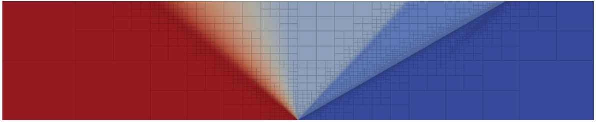
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

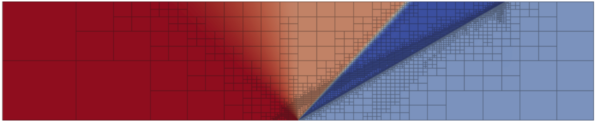
Mesh 14



Primitive Variables



Conservation Variables

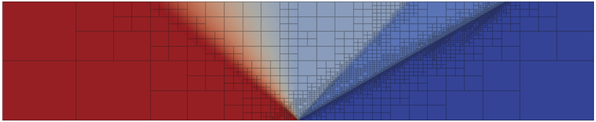


Entropy Variables

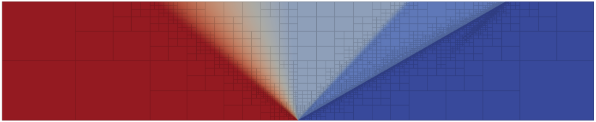
Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

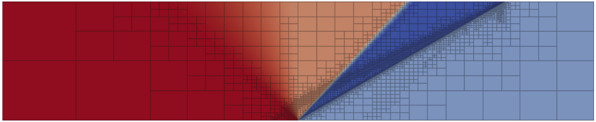
Mesh 15



Primitive Variables



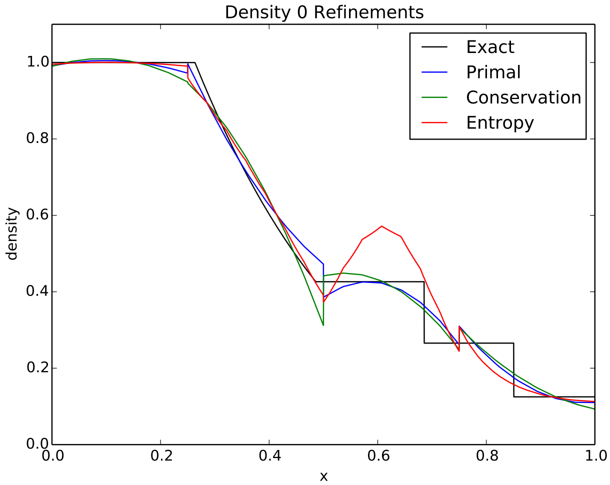
Conservation Variables

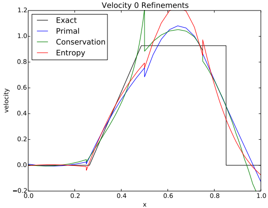


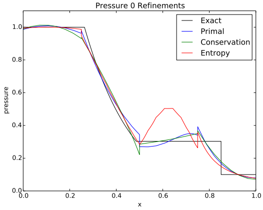
Entropy Variables

Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

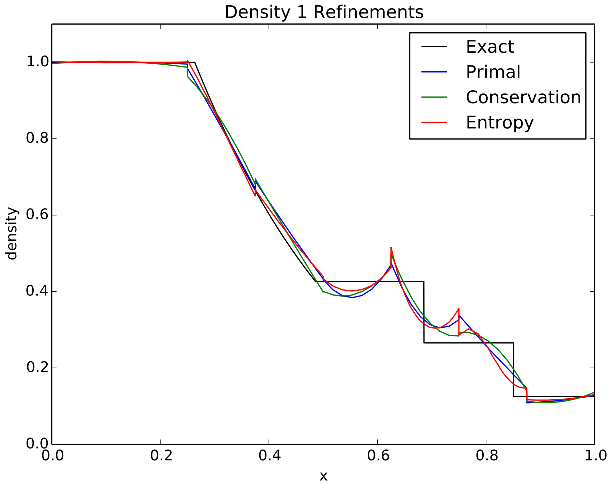


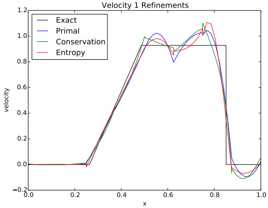


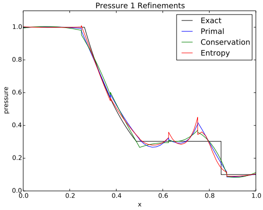


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

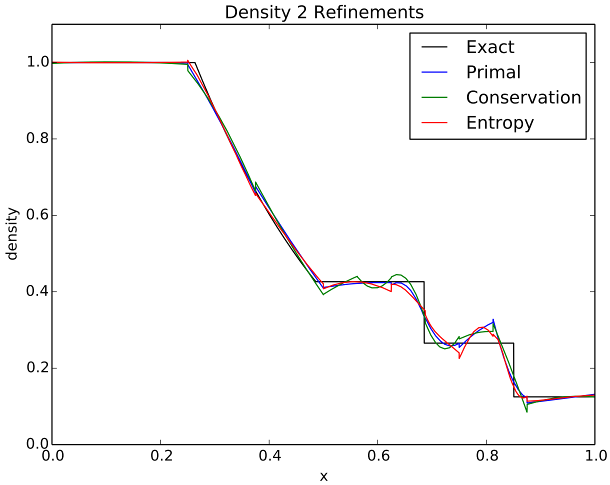


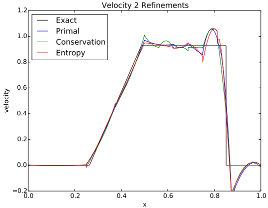


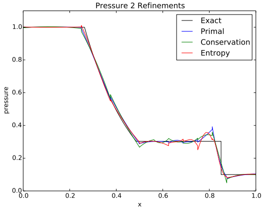


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

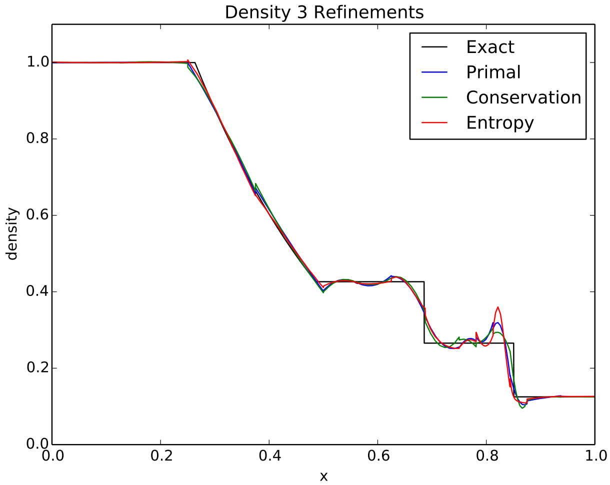


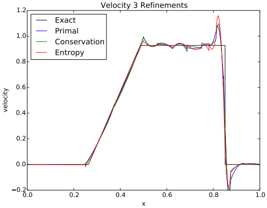


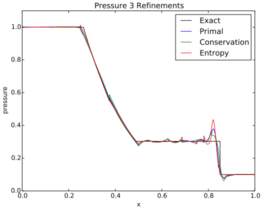


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

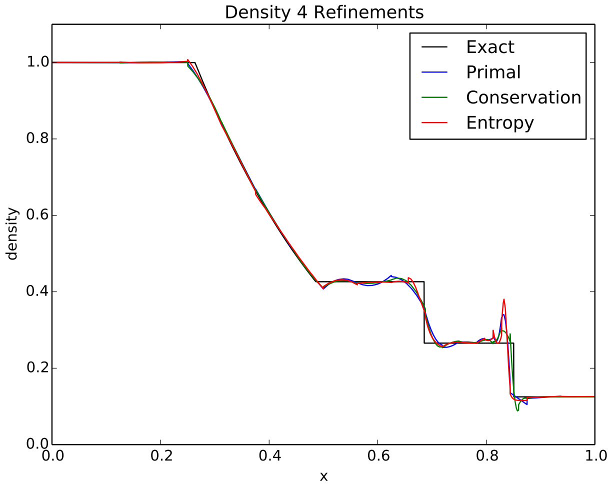


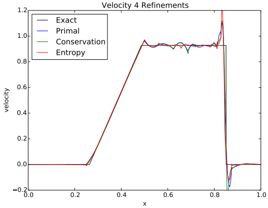


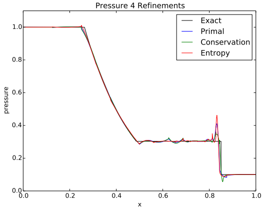


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

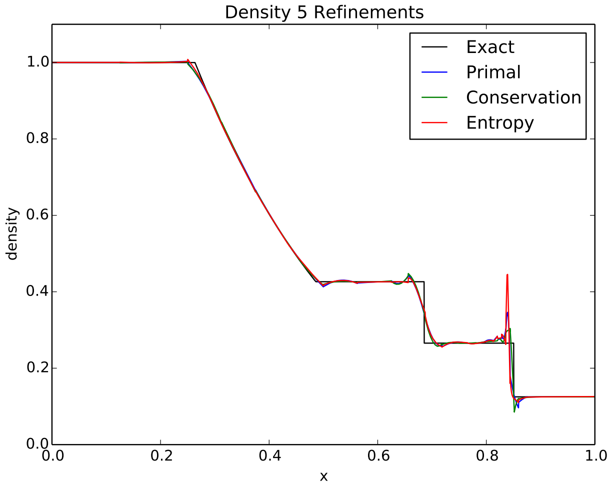


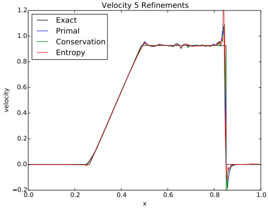


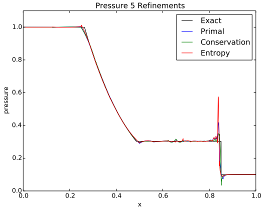


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

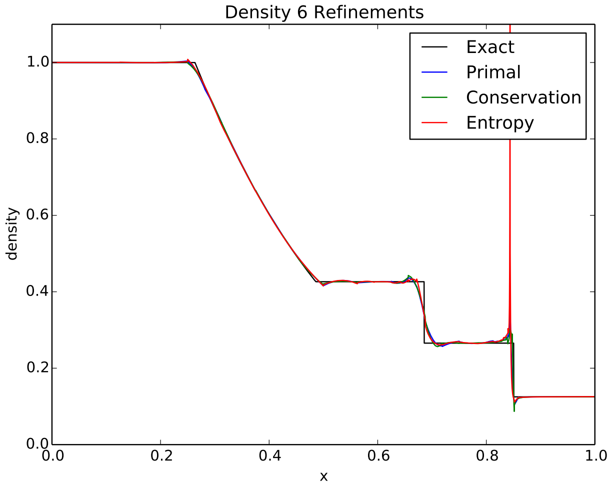


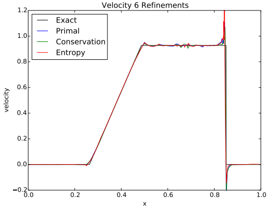


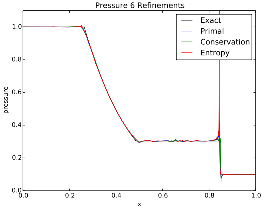


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

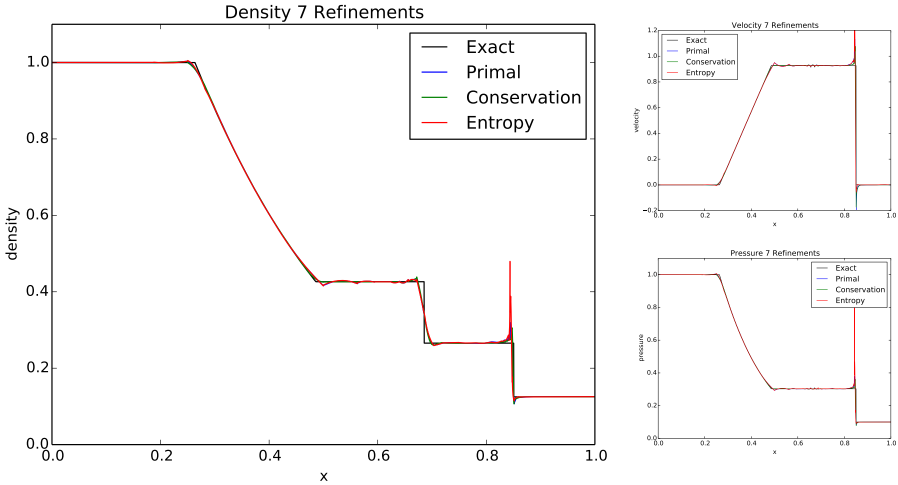






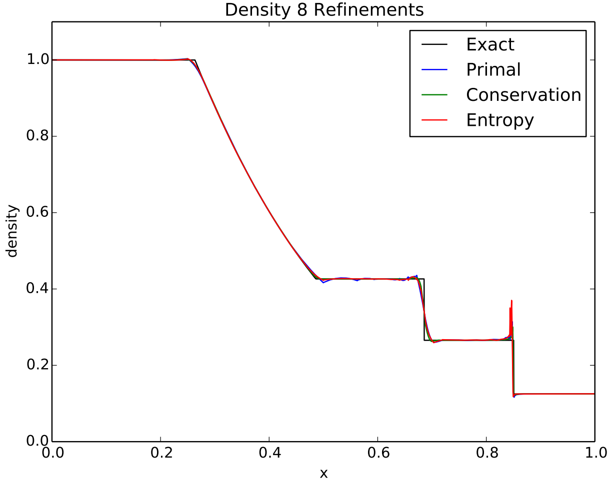
Space-Time Compressible Navier-Stokes

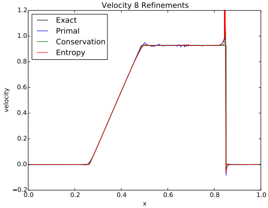
Sod Shock Tube with $\mu = 10^{-5}$

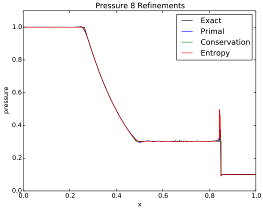


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

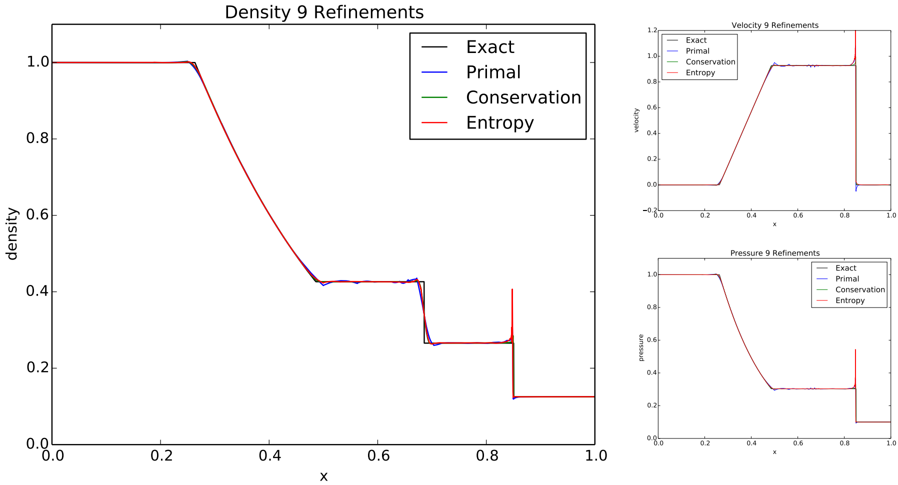






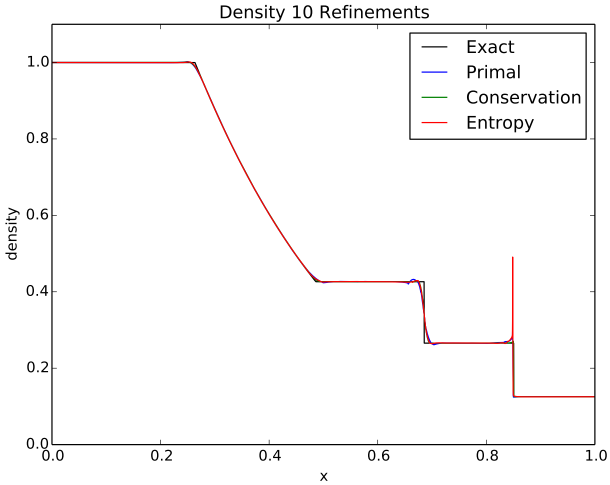
Space-Time Compressible Navier-Stokes

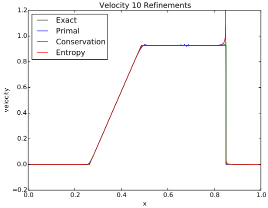
Sod Shock Tube with $\mu = 10^{-5}$

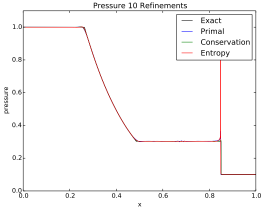


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

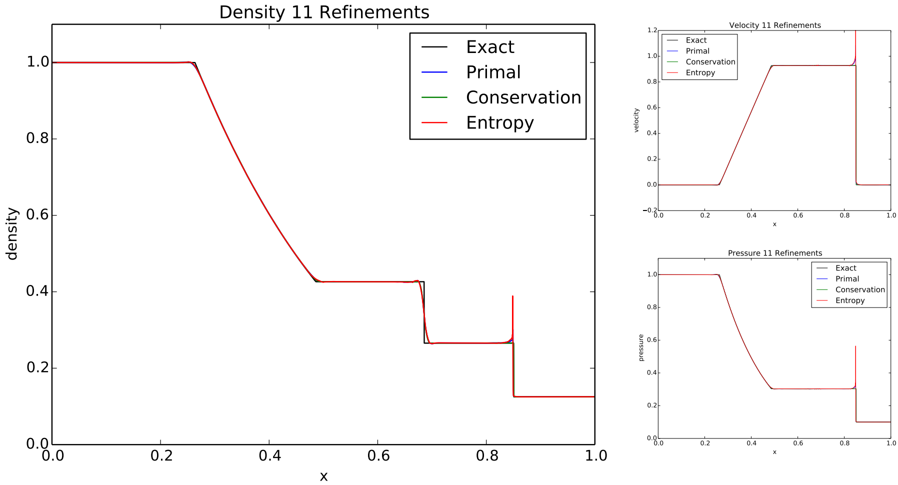






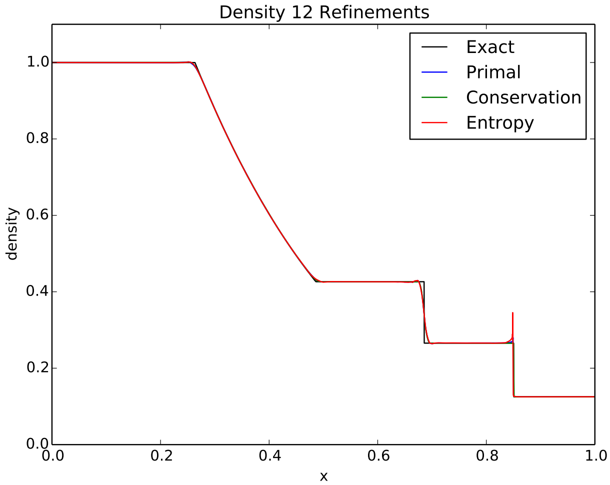
Space-Time Compressible Navier-Stokes

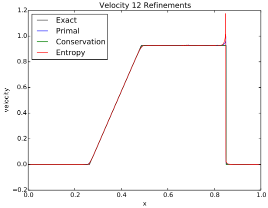
Sod Shock Tube with $\mu = 10^{-5}$

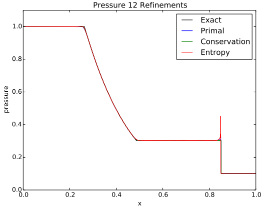


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

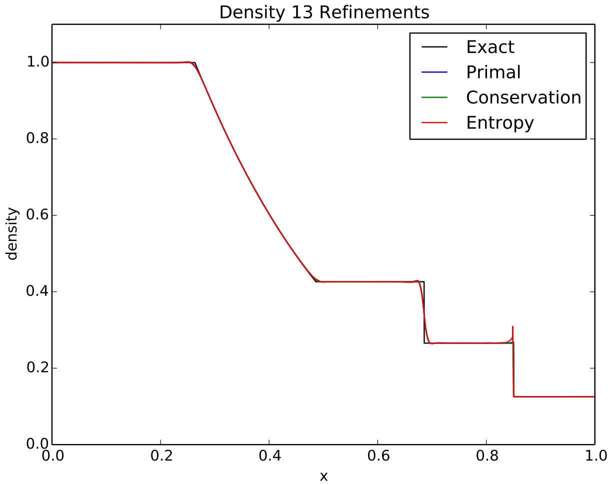


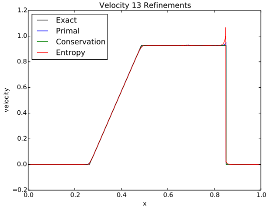


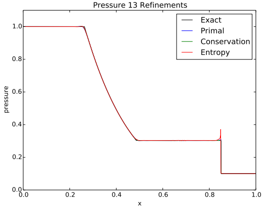


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$

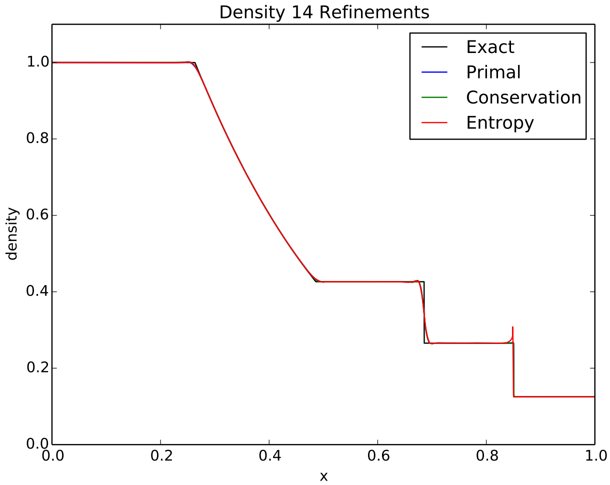


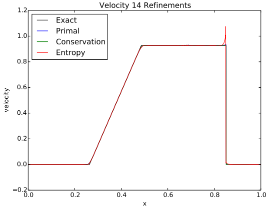


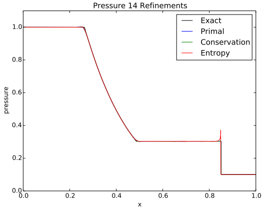


Space-Time Compressible Navier-Stokes

Sod Shock Tube with $\mu = 10^{-5}$







THE UNIVERSITY OF TEXAS AT AUSTIN

Let W, U, and V be the set of primitive, conservation, and entropy variables.
The entropy function

$$H = -\rho \log(p\rho^{-\gamma})$$

provides a natural residual for the Navier-Stokes equations.
$A_0 = H_{,UU} = V_{,U}$ is known as the symmetrizer and $(U, A_0 U)$ provides a natural metric for the linearized Euler equations.
In primitive variables:

$$(U, A_0 U) = (U_{,W} W, V_{,U} U_{,W} W) = (W, U_{,W}^T V_{,U} U_{,W} W) = (W, A_0(W) W)$$

where

$$A_0(W) = U_{,W}^T V_{,U} U_{,W} = \begin{bmatrix} \frac{\gamma-1}{\rho} & 0 & 0 \\ 0 & \frac{\rho}{C_v T} & 0 \\ 0 & 0 & \frac{\rho}{T^2} \end{bmatrix}$$

$(W, A_0 W)$ has units of density.

Bilinear form with group variables:

$$b\left(\left(W,\hat{W}\right),v\right)=\left(W,A_h^* v\right)_{L^2(\Omega_h)}+\left\langle \hat{W},[\![v]\!]\right\rangle_{\Gamma_h}$$

For conforming v^* satisfying $A^* v^* = A_0 W$

$$\left\|A_0^{\frac{1}{2}}W\right\|^2 = b(W,v^*) = \frac{b(W,v^*)}{\|v^*\|_V}\|v^*\|_V$$
$$\leq \sup_{v^*\neq 0}\frac{|b(W,v^*)|}{\|v^*\|}\|v^*\| = \|W\|_E\|v^*\|_V .$$

Necessary robustness condition:

$$\|v^*\|_V \lesssim \left\|A_0^{\frac{1}{2}}W\right\|_{L^2(\Omega_h)}$$
$$\Rightarrow \left\|A_0^{\frac{1}{2}}W\right\|_{L^2(\Omega_h)} \lesssim \|W\|_E$$

We load our adjoint equations with $A_0 W$:

$$\frac{1}{\mu}M^*(\Psi) + K^*(\nabla V) = 0$$

$$-\begin{pmatrix} F^* \\ C^* \end{pmatrix}(\nabla_{xt}V) + G^*(\nabla\Psi) = A_0 W$$

This leads to the entropy scaled robust norm:

$$\begin{aligned}\|(V,\Psi)\|^2_{V,K} := & \left\|A_0^{-\frac{1}{2}}(F^* + C^*)\right\|^2_K + \mu\left\|A_0^{-\frac{1}{2}}K^*\right\|^2_K \\ & + \min\left(\frac{\mu}{h^2}, 1\right)\left\|A_0^{-\frac{1}{2}}V\right\|^2_K + \left\|A_0^{-\frac{1}{2}}G^*\right\|^2_K \\ & + \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right)\left\|A_0^{-\frac{1}{2}}M^*\right\|^2_K\end{aligned}$$

THE UNIVERSITY OF TEXAS AT AUSTIN

We load our adjoint equations with $A_0 W$:

$$\frac{1}{\mu}M^*(\Psi) + K^*(\nabla V) = 0$$

$$-\begin{pmatrix} F^* \\ C^* \end{pmatrix}(\nabla_{xt} V) + G^*(\nabla \Psi) = A_0 W$$

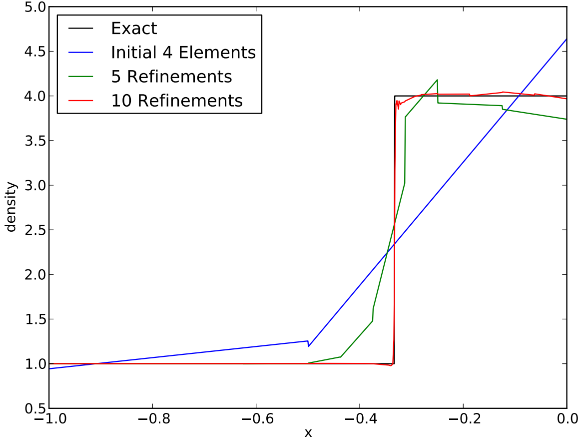
This leads to the entropy scaled robust norm:

$$\begin{aligned}\|(V,\Psi)\|^2_{V,K} := {}& \left\|A_0^{-\frac{1}{2}}(F^* + C^*)\right\|^2_K + \mu\left\|A_0^{-\frac{1}{2}}K^*\right\|^2_K \\ &+ \min\left(\frac{\mu}{h^2}, 1\right)\left\|A_0^{-\frac{1}{2}}V\right\|^2_K + \left\|A_0^{-\frac{1}{2}}G^*\right\|^2_K \\ &+ \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right)\left\|A_0^{-\frac{1}{2}}M^*\right\|^2_K\end{aligned}$$

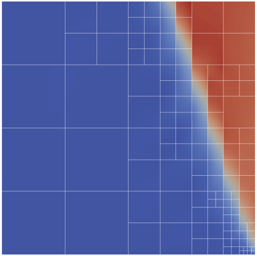
Numerical results were disappointing.

Space-Time Compressible Navier-Stokes

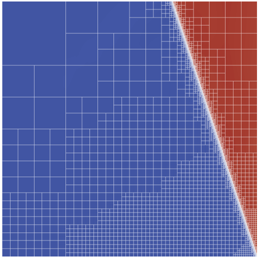
Noh Implosion with Primitive Variables, Robust Norm, $\mu = 10^{-3}$



After 5 refinements



After 10 refinements



Space-Time Compressible Navier-Stokes

Piston with $\mu = 10^{-2}$

$$\hat{t}_c = \sqrt{2}(-\rho u + \rho)$$

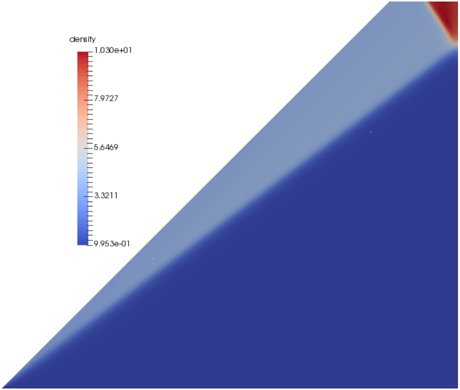
$$\hat{t}_m = \sqrt{2}(-\rho u^2 - \rho R T + \rho u)$$

$$\hat{t}_e = \sqrt{2}(-\rho u(C_v T + \frac{1}{2}u^2) - u\rho R T + \rho(C_v T + \frac{1}{2}u^2))$$

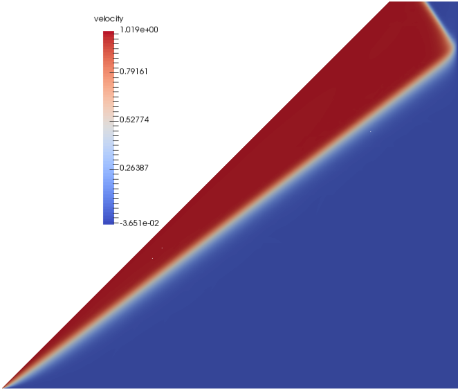
$$\hat{u} = 1$$

$$\hat{t}_c = 0$$

$$\hat{t}_m - \hat{t}_e = 0$$



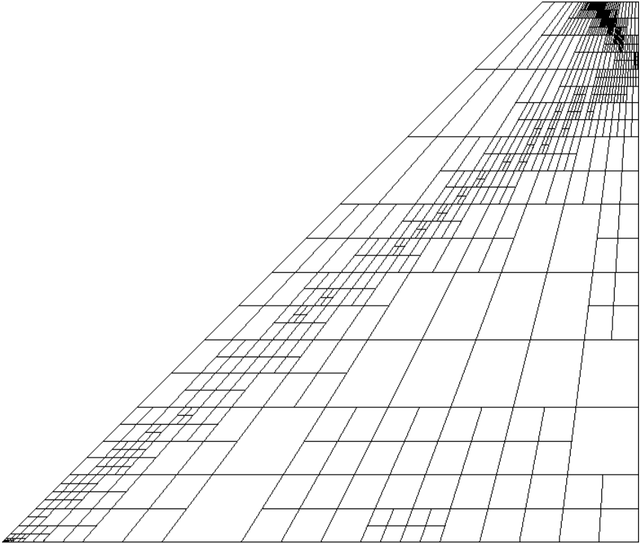
Density



Velocity

Space-Time Compressible Navier-Stokes

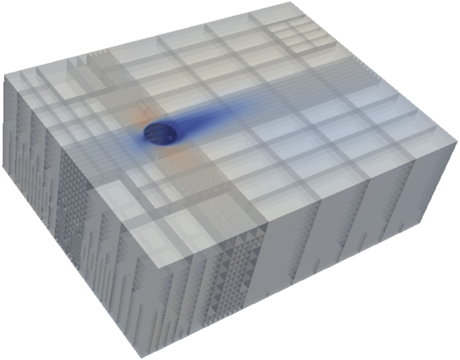
Piston with $\mu = 10^{-2}$



Mesh after 8 adaptive refinements

Future Directions

- **Improve performance:** line smoothing for multigrid
- **Shock capturing:** DPG makes no promises when it comes to Gibbs phenomenon
- **Non-Hilbert DPG:** L^1 is known to limit oscillations
- **Anisotropic refinements:** necessary for time slabs
- **More extensive 2D results:** shedding vortex problems, 2D shock problems
- **3D results:** will not be cheap



Incompressible Flow Over a Cylinder

Thank You!

THE UNIVERSITY OF TEXAS AT AUSTIN

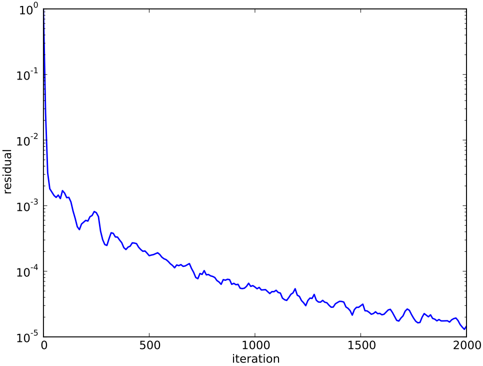
Recommended References

- J.L. Chan. ``A DPG Method for Convection-Diffusion Problems''. PhD thesis. University of Texas at Austin, 2013.
- D. Moro, N.C. Nguyen, and J. Peraire. ``A Hybridized Discontinuous Petrov-Galerkin Scheme for Scalar Conservation Laws''. In: *Int. J. Num. Meth. Eng.* (2011).
- C.L. Chang and J.J. Nelson. ``Least-Squares Finite Element Method for the Stokes Problem with Zero Residual of Mass Conservation''. In: *SIAM J. Num. Anal.* 34 (1997), pp. 480–489.
- J. Chan et al. ``A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms''. In: *Comp. Math. Appl.* 67.4 (2014), pp. 771 –795.
- T.E. Ellis, L.F. Demkowicz, and J.L. Chan. ``Locally Conservative Discontinuous Petrov-Galerkin Finite Elements For Fluid Problems''. In: *Comp. Math. Appl.* 68.11 (2014), pp. 1530 –1549.
- T. Ellis, J. Chan, and L. Demkowicz. ``Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations,x Eds. G.R. Barrenechea et al.'' In: vol. 114. Lecture Notes in Computational Science and Engineering. in print, see also ICES Report 2015/21. Springer, 2016. Chap. Robust DPG Methods for Transient Convection-Diffusion.
- L.F. Demkowicz and J. Gopalakrishnan. ``Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations (eds. X. Feng, O. Karakashian, Y. Xing)''. In: vol. 157. IMA Volumes in Mathematics and its Applications, 2014. Chap. An Overview of the DPG Method, pp. 149–180.
- N.V. Roberts. ``Camellia: A Software Framework for Discontinuous Petrov-Galerkin Methods''. In: *Comp. Math. Appl.* 68.11 (2014), pp. 1581 –1604.
- L.F. Demkowicz and N. Heuer. ``Robust DPG Method for Convection-Dominated Diffusion Problems''. In: *SIAM J. Numer. Anal.* 51.5 (2013), pp. 1514–2537.
- N. Roberts, T. Bui-Thanh, and L. Demkowicz. ``The DPG method for the Stokes problem''. In: *Comp. Math. Appl.* 67.4 (2014), pp. 966 –995.

Scaling Issues

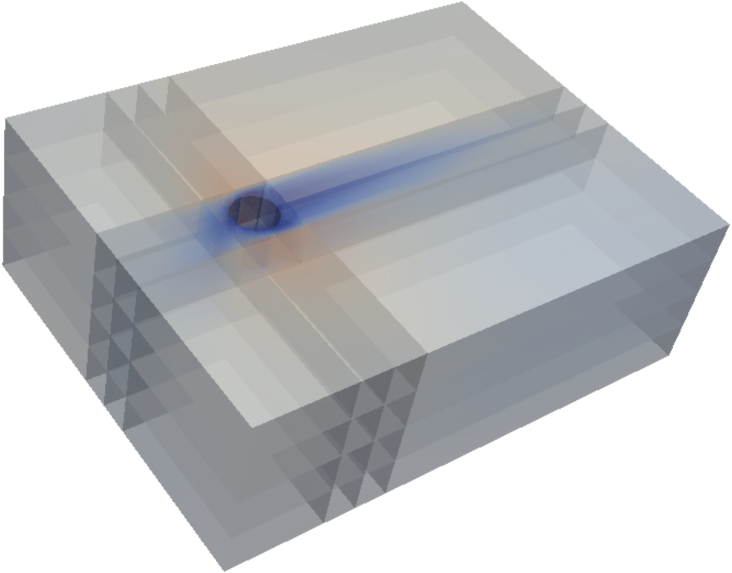
Multigrid and Convection-Diffusion

- Convection-diffusion, $\epsilon = 10^{-2}$, 64×64 mesh
- Conjugate gradient
- Geometric multigrid preconditioner
- Multiplicative V-cycle
- Overlapping additive Schwarz smoother
- Hierarchy of $p-$coarsening followed by $h-$coarsening



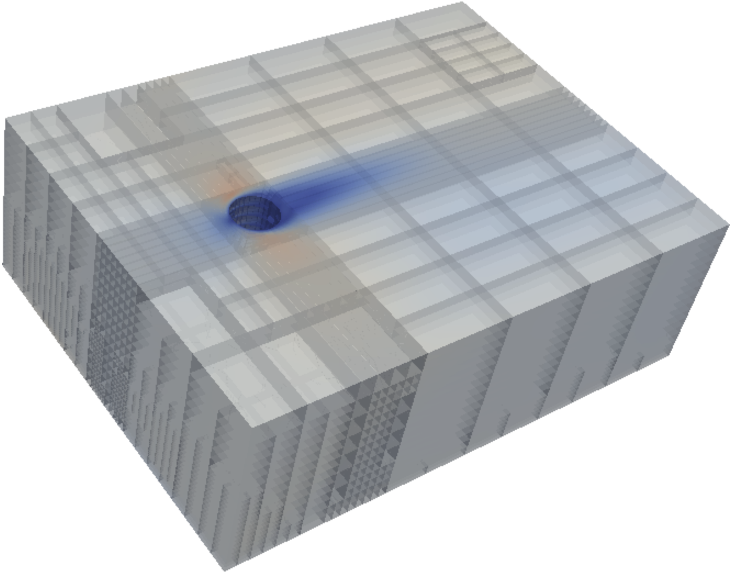
Scaling Issues

Incompressible Flow Over a Cylinder, Initial Mesh



Scaling Issues

Incompressible Flow Over a Cylinder, 4 Refinements



Scaling Issues

Solve Times and Strong Scaling

Transient Flow Over a Cylinder

			1 Node	4 Nodes		32 Nodes	
Ref	Elems	DOFs	Time	Time	Scaling vs 1	Time	Scaling vs 4
0	80	31,304	1,772	453	3.91	451	1.01
1	605	225,908	8,190	3,574	2.29	717	4.98
2	3,013	1,081,598	32,008	12,076	2.65	2,648	4.56
3	9,726	3,429,384		28,744		6,319	4.54
4	11,742	4,144,674				8,510	

Computations on Lonestar, 1 node = 24 processors

32,008 seconds = 8.8 hours

28,744 seconds = 8.0 hours

8,510 seconds = 2.4 hours

Scaling Issues

Solve Times and Strong Scaling

Taylor-Green Vortex

			1 Node	4 Nodes	
Ref	Elems	DOFs	Time	Time	Scaling vs 1
0	60	21,302	331	140	2.35
1	312	108,410	945	290	3.25
2	2,020	691,834	4,880	1,363	3.58
3	9,244	3,043,024		6,171	

Computations on Lonestar, 1 node = 24 processors

4,880 seconds = 1.4 hours

6,171 seconds = 1.7 hours

Assumptions:

- The maximum required spatial resolution is much finer than the required temporal resolution.
- Regions requiring high spatial resolution are concentrated in relatively compact parts of the domain.
- Only isotropic refinements are permitted.
- The number of time slabs is a power of 2.

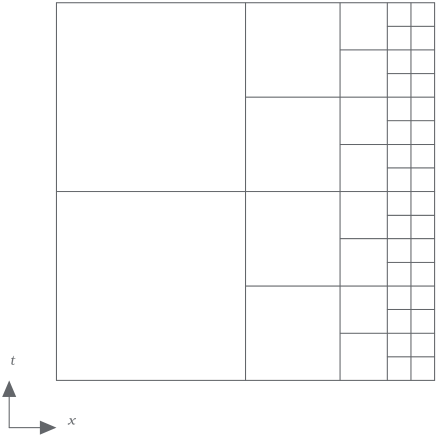
Test problem: convection diffusion with exact solution

$$u = 1 - e^{\frac{x}{\epsilon}}$$

on space-time domain $[-1, 0] \times [0, 1]$.

Scaling Issues

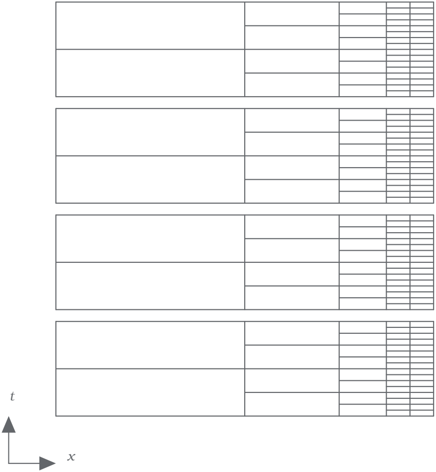
Space-Time Slabs



Single Slab Strategy

Scaling Issues

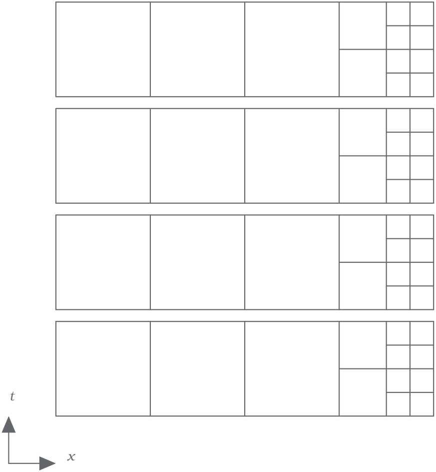
Space-Time Slabs



Naive Time Slab Strategy

Scaling Issues

Space-Time Slabs

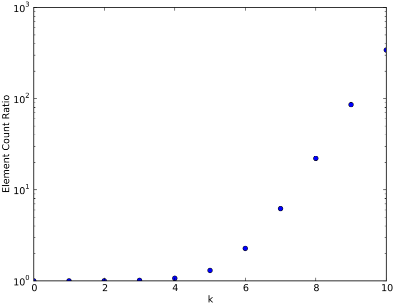


Smarter Time Slab Strategy

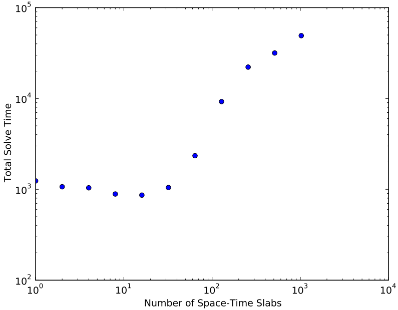
Scaling Issues

Space-Time Slabs

Number of time slabs = 2^k.



Ratio of total element counts



Total solve time using smart time slabs

- Without anisotropic refinements, time slabs don't significantly speed up computations.
- Time slabs could be useful for memory constrained problems.

Camellia: DPG for the Masses

Overview

Design Goal

Make DPG research and experimentation as simple as possible, while maintaining computational efficiency and scalability.

Built on Trilinos (Teuchos, Intrepid, Shards, Epetra, etc).

Mature support for:

- Rapid specification of DPG variational forms, inner products, etc.
- Distributed computation of stiffness matrix
- 1D - 3D geometries
- Curvilinear elements
- h- and p-refinements (anisotropic in h)

Experimental support for:

- Space-time computations
- Iterative solvers (tested up to 32,768 cores)

Convection-Diffusion in Three Slides

Building the Bilinear Form

```
VarFactory vf;
//fields:
VarPtr u = vf.fieldVar("u", L2);
VarPtr sigma = vf.fieldVar("sigma", VECTOR_L2);

// traces:
VarPtr u_hat = vf.traceVar("u_hat");
VarPtr t_n = vf.fluxVar("t_n");

// test:
VarPtr v = vf.testVar("v", HGRAD);
VarPtr tau = vf.testVar("tau", HDIV);

double eps = .01;
FunctionPtr beta_x = Function::constant(1);
FunctionPtr beta_y = Function::constant(2);
FunctionPtr beta = Function::vectorize(beta_x, beta_y);

BFPtr bf = Teuchos::rcp( new BF(vf) );

bf->addTerm((1/eps) * sigma, tau);
bf->addTerm(u, tau->div());
bf->addTerm(-u_hat, tau->dot_normal());

bf->addTerm(sigma - beta * u, v->grad());
bf->addTerm(t_n, v);

RHSPtr rhs = RHS::rhs();
```

Find $u \in L^2(\Omega_h)$, $\boldsymbol{\sigma} \in \boldsymbol{L}^2(\Omega_h)$, $\hat{u} \in H^{\frac{1}{2}}(\Gamma_h)$, $\hat{t}_n \in H^{-\frac{1}{2}}(\Gamma_h)$ such that

$$\frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau}) + (u, \nabla \cdot \boldsymbol{\tau}) - \langle \hat{u}, \boldsymbol{\tau} \cdot \boldsymbol{n} \rangle$$
$$- (\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v) + \langle \hat{t}_n, v \rangle = (f, v)$$

for all $v \in H^1(K)$, $\boldsymbol{\tau} \in \boldsymbol{H}(\text{div}, K)$,

where $\epsilon = 10^{-2}$, $\boldsymbol{\beta} = (1, 2)^T$ and

$f = 0$.

Convection-Diffusion in Three Slides

Boundary Conditions and Mesh

```
int k = 2;
int delta_k = 2;
MeshPtr mesh = MeshFactory::quadMesh(bf, k+1, delta_k);

BCPtr bc = BC::bc();

SpatialFilterPtr y_equals_one = SpatialFilter::matchingY(1.0);
SpatialFilterPtr y_equals_zero = SpatialFilter::matchingY(0);
SpatialFilterPtr x_equals_one = SpatialFilter::matchingX(1.0);
SpatialFilterPtr x_equals_zero = SpatialFilter::matchingX(0.0);

FunctionPtr zero = Function::zero();
FunctionPtr x = Function::xn(1);
FunctionPtr y = Function::yn(1);
bc->addDirichlet(t_n, y_equals_zero, -2 * (1-x));
bc->addDirichlet(t_n, x_equals_zero, -1 * (1-y));
bc->addDirichlet(u_hat, y_equals_one, zero);
bc->addDirichlet(u_hat, x_equals_one, zero);
```

Create a square mesh $[0, 1] \times [0, 1]$ with boundary conditions

- $\hat{t}_n = 2x - 2$ on $y = 0$
- $\hat{t}_n = x - 1$ on $x = 0$
- $\hat{u} = 0$ on $y = 1$
- $\hat{u} = 0$ on $x = 1$

Note

- Can subclass SpatialFilter to match any geometry
- Adding new mesh readers is straightforward

Convection-Diffusion in Three Slides

Test Norm, Solving, and Adaptivity

```
IPPtr ip = bf->graphNorm();

SolutionPtr soln = Solution::solution(mesh, bc, rhs, ip);

double threshold = 0.20;
RefinementStrategy refStrategy(soln, threshold);

int numRefs = 10;

ostringstream refName;
refName << "ConvectionDiffusion";
HDF5Exporter exporter(mesh,refName.str());

for (int refIndex=0; refIndex < numRefs; refIndex++) {
  soln->solve();

  double energyError = soln->energyErrorTotal();
  cout << "After␣" << refIndex << "␣refinements,␣energy␣error␣is␣" << energyError << endl;

  exporter.exportSolution(soln, vf, refIndex);

  if (refIndex != numRefs)
    refStrategy.refine();
}
```

Convection-Diffusion in Three Slides

Computed Solution