Space-Time Discontinuous Petrov-Galerkin Finite Elements for Transient Computational Fluid Dynamics

Truman Ellis

Advisors: Leszek Demkowicz, Robert Moser

THE UNIVERSITY OF TEXAS AT AUSTIN

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

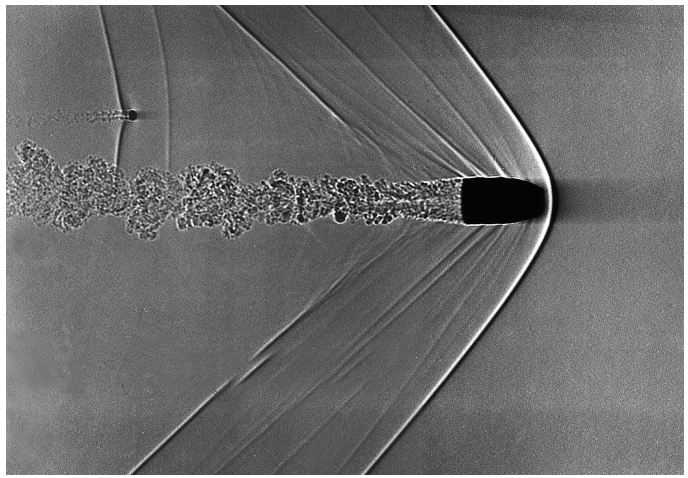
Navier-Stokes Equations

Numerical Challenges

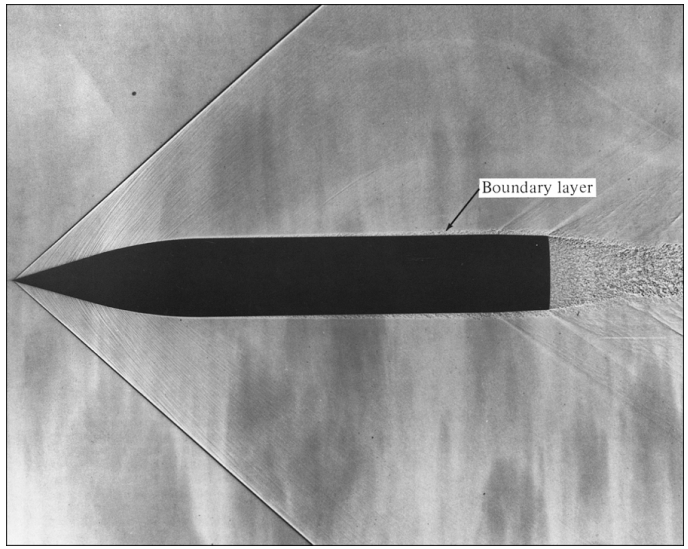
THE UNIVERSITY OF TEXAS AT AUSTIN

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)
- Nonlinear convergence and uniqueness of solutions
- Stability of numerical schemes
  - Coarse/adaptive grids
  - Higher order



Shock

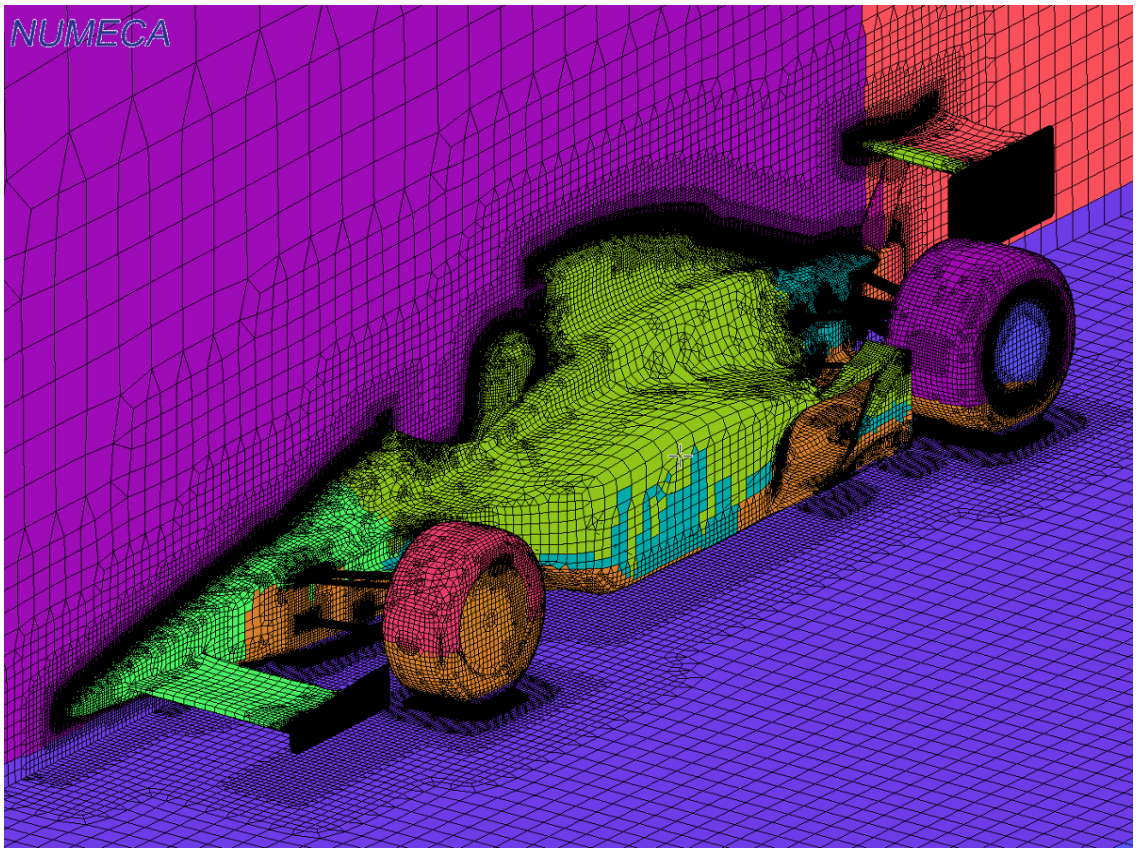


Boundary layer

Motivation

Initial Mesh Design is Expensive and Time-Consuming

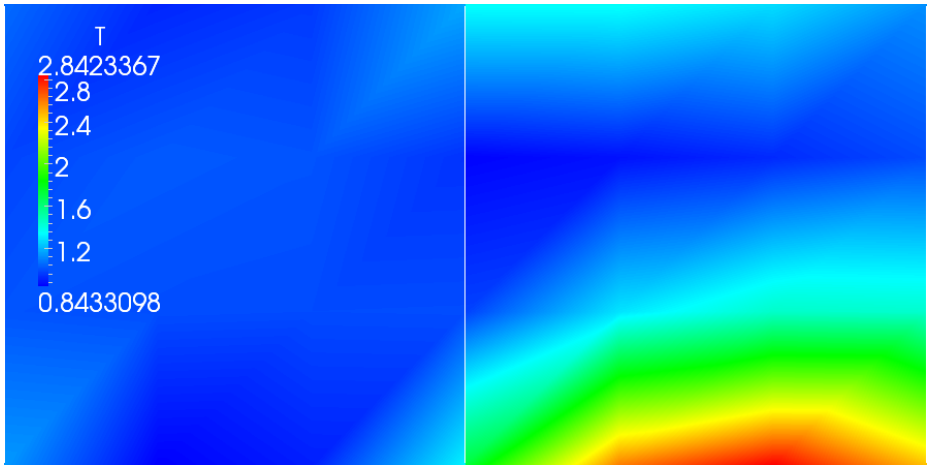
- Surface mesh must accurately represent geometry
- Volume mesh needs sufficient resolution for asymptotic regime
- Boundary layer meshes must respect y^+ guidelines
- Engineers often forced to work by trial and error
- Bad in the context of HPC



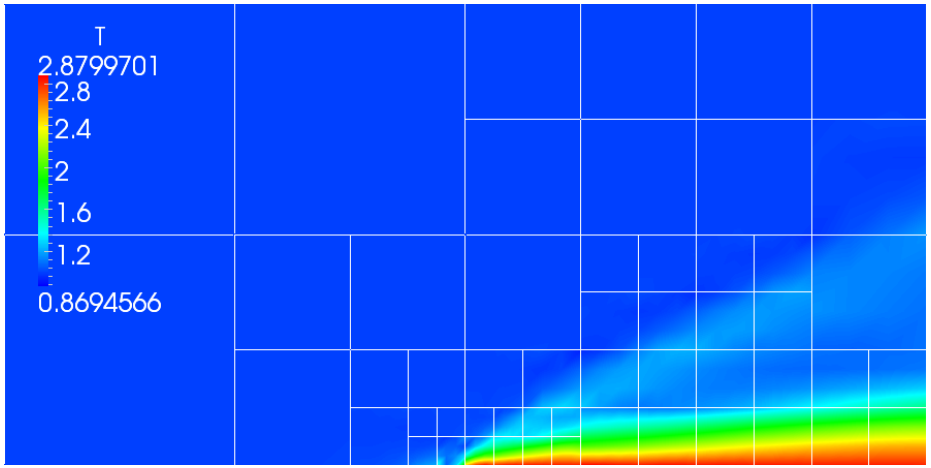
Formula 1 Mesh by Numeca

DPG on Coarse Meshes

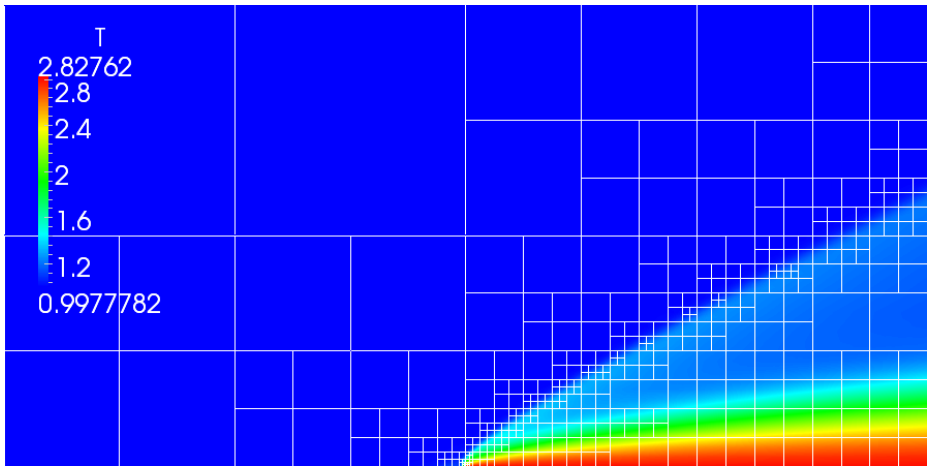
Adaptive Solve of the Carter Plate Problem[1] $Re = 1000$



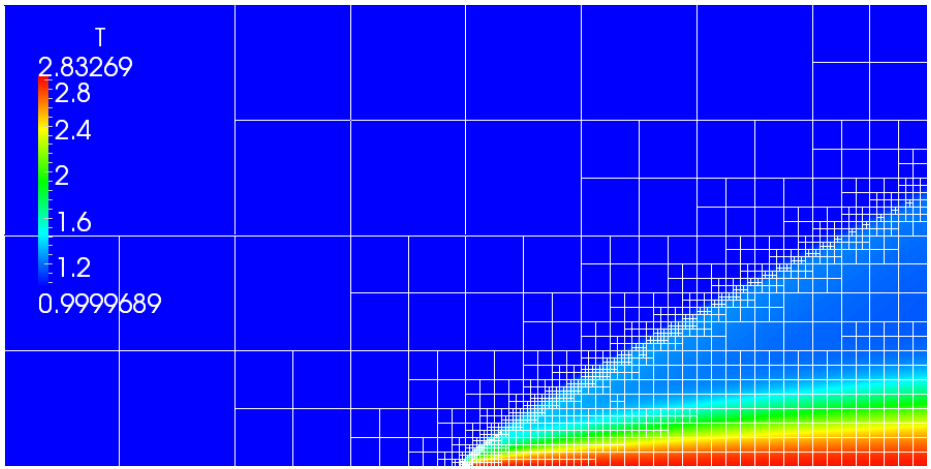
Temperature on Initial Mesh



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

THE UNIVERSITY OF TEXAS AT AUSTIN

Streamline Upwind Petrov-Galerkin: Changing your 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 the a minimum residual context (i.e. as a Ritz method).

Space-Time FEM: Highly adaptive methods should have adaptive time integration. Space-time is a very natural framework in which to do moving mesh simulations.

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

Overview of DPG

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 $Bu - l \in V'$:

$$u_h = \arg\min_{w_h \in U_h} \frac{1}{2} \|Bu - l\|_{V'}^2 .$$

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

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

Overview of DPG

THE UNIVERSITY OF TEXAS AT AUSTIN

DPG is a Minimum Residual Method

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

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

Overview of DPG

THE UNIVERSITY OF TEXAS AT AUSTIN

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[2]

Other Features

THE UNIVERSITY OF TEXAS AT AUSTIN

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

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

[2] **DPGOverview2**.

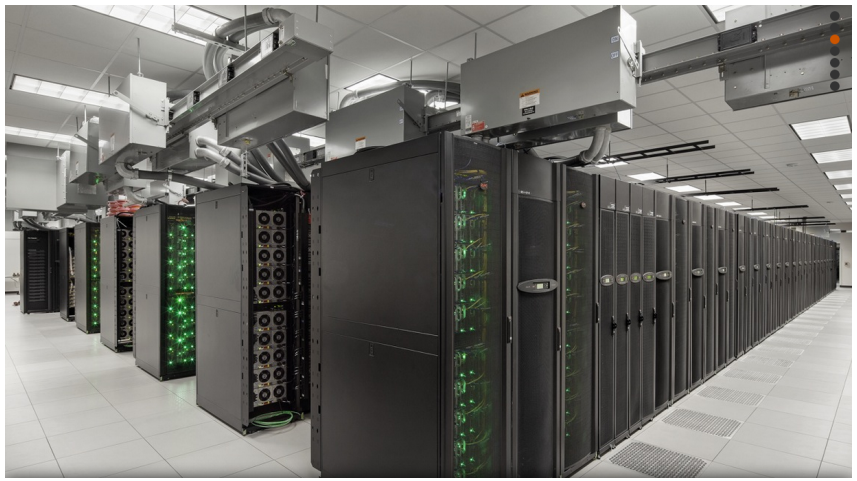
Overview of DPG

High Performance Computing

THE UNIVERSITY OF TEXAS AT AUSTIN

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

Space-Time DPG

Motivation

Extends the capabilities of a DPG solver

- Preserves stability and robustness of DPG method
- Unified treatment of space and time
- Local space-time adaptivity (local time stepping)
  - Small solution features require small time step
  - Global time step not limited to smallest element
- Natural framework for moving meshes

More computationally difficult

- Solve requires $d + 1$ dimensions
- Mesh structure more difficult
- Need to differentiate between spatial and temporal boundaries
- Larger global solves than finite difference time stepping

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

Camellia: DPG for the Masses

THE UNIVERSITY OF TEXAS AT AUSTIN

Overview

Design Goal

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

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)
- Arbitrarily irregular meshes
- Modular refinement strategy interface

Experimental support for:

- Space-time computations
- Iterative solvers

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

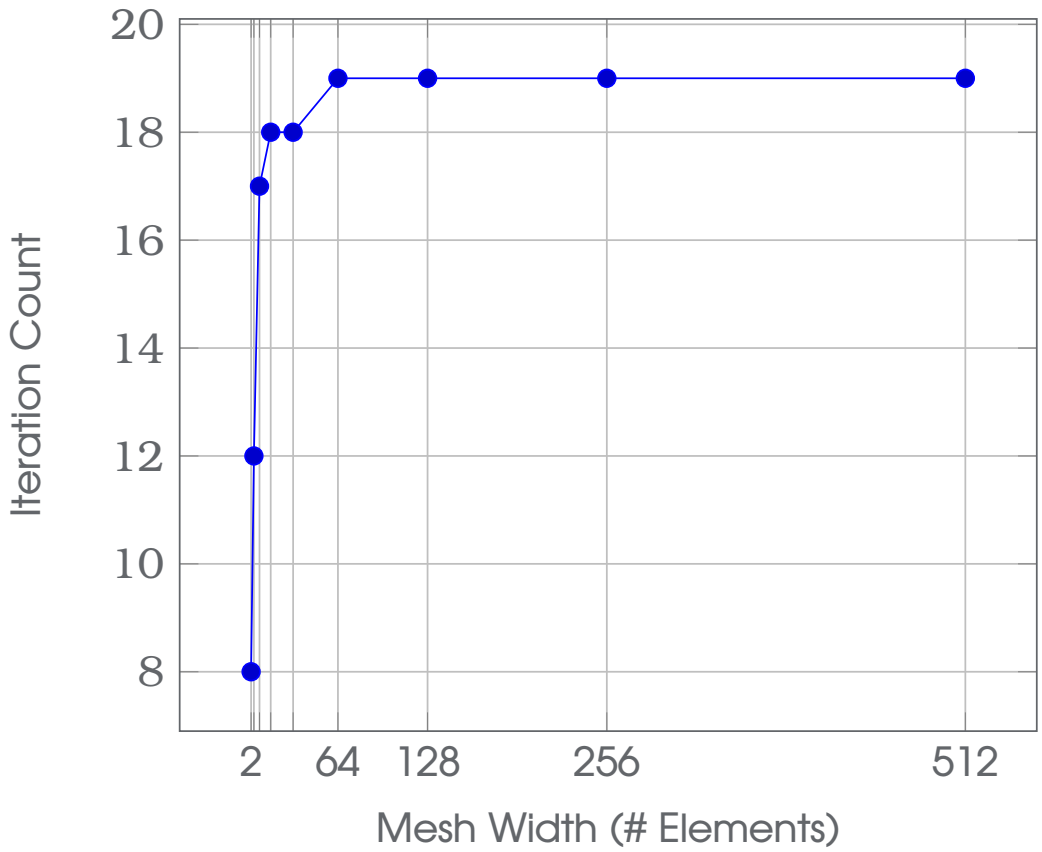
Test Norm, Solving, and Adaptivity

THE UNIVERSITY OF TEXAS AT AUSTIN

```
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();
}
```

Towards a Robust Iterative Solver

Poisson 1D, p-multigrid Preconditioners, $k = 16$

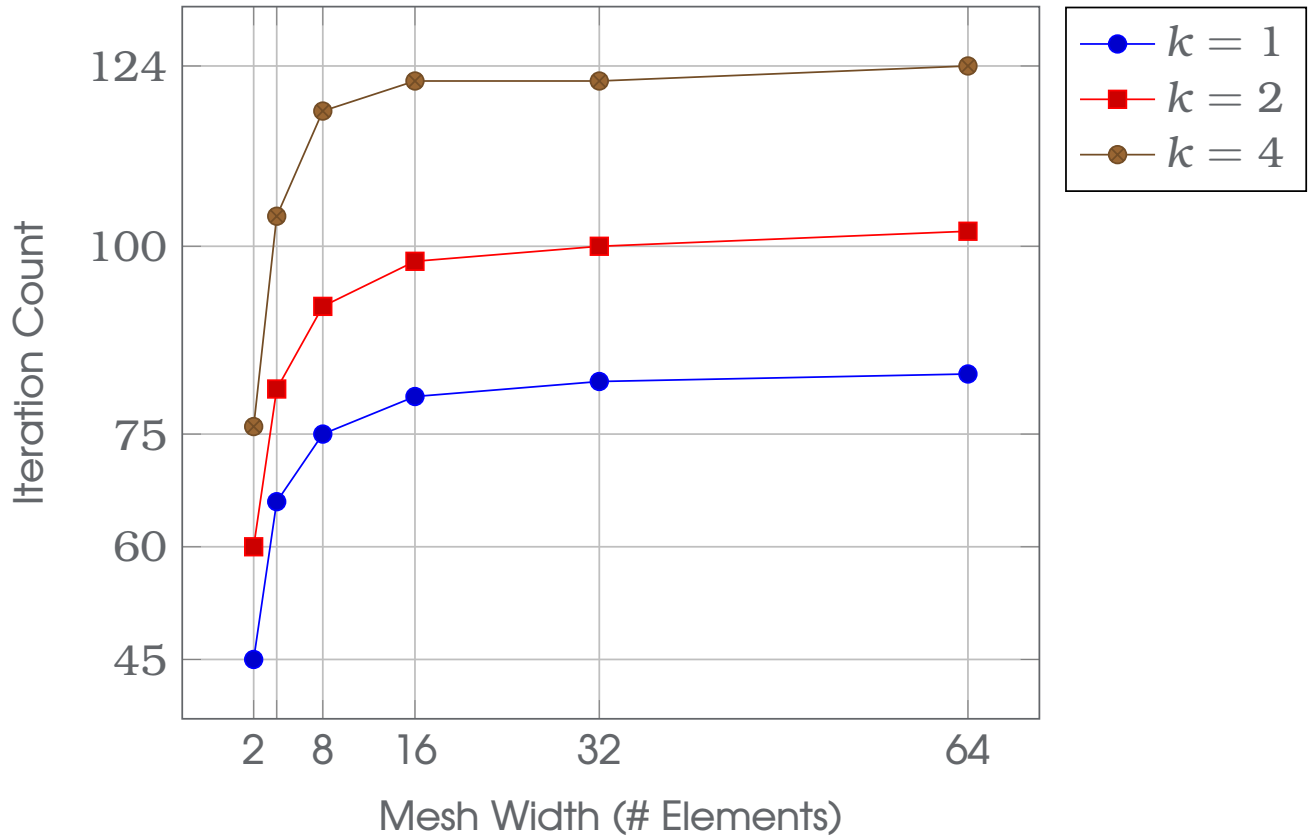


Poisson 1D: number of CG iterations to reduce error by a factor of 10^{10} using p-multigrid preconditioners with the indicated type of Schwarz smoother for $k = 16$. The results for $k = 1, 2, 4$, and 8 are essentially identical.

Towards a Robust Iterative Solver

THE UNIVERSITY OF TEXAS AT AUSTIN

Stokes 2D, p-multigrid Preconditioners, Schwarz algebraic overlap 0

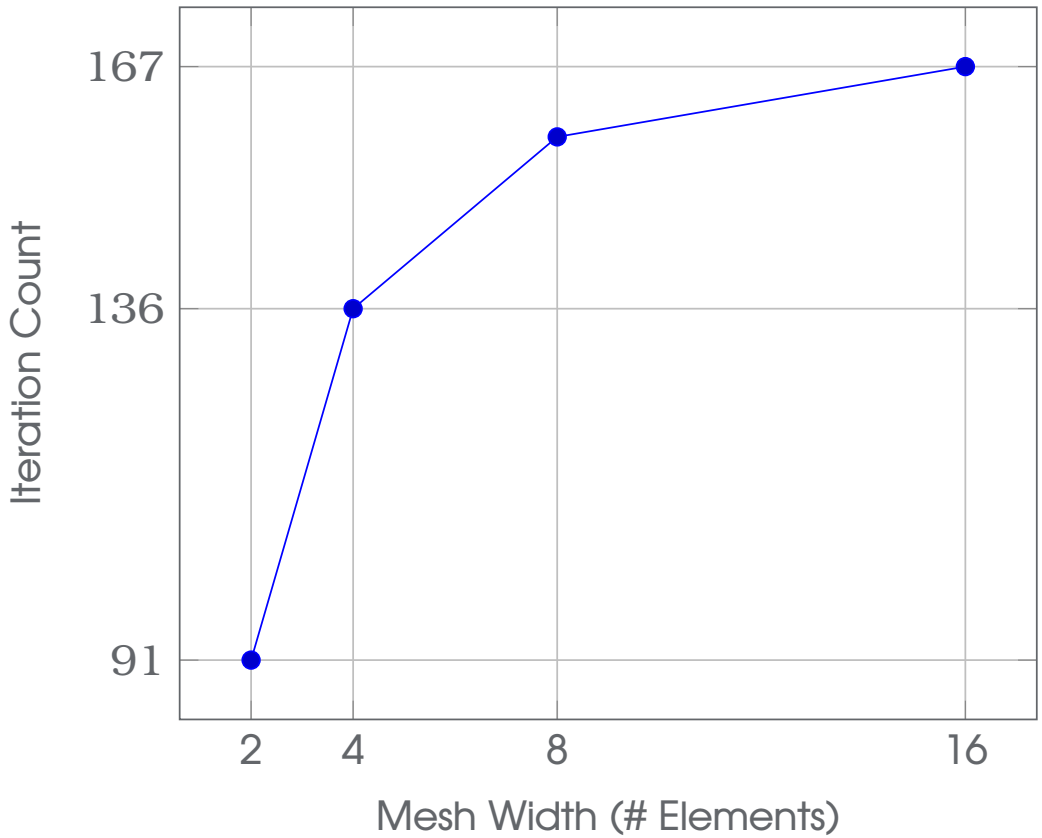


Stokes 2D: number of CG iterations to reduce error by a factor of 10^{10} using p-multigrid, algebraic Schwarz smoother with 0 overlap.

Towards a Robust Iterative Solver

THE UNIVERSITY OF TEXAS AT AUSTIN

Stokes 3D, p-multigrid Preconditioners, $k = 2$, Schwarz with Incomplete Cholesky



Stokes 3D: number of CG iterations to reduce error by a factor of 10^{10} using statically condensed system matrix with p-multigrid, algebraic Schwarz smoother, using Incomplete Cholesky factorization (fill ratio 5) for the Schwarz blocks.

Table of Contents

THE UNIVERSITY OF TEXAS AT AUSTIN

Other DPG Research

THE UNIVERSITY OF TEXAS AT AUSTIN

- Entropy scaling for physically meaningful test norms
- DPG for non-Hilbert L_p spaces

Thank You!

THE UNIVERSITY OF TEXAS AT AUSTIN