PECOS

Predictive Engineering and Computational Sciences

Camellia

A Discontinuous Petrov-Galerkin Toolbox Using Trilinos

Jesse Chan, Nate Roberts

The University of Texas at Austin

March 7, 2012

Acknowledgments

Collaborators:

- Leszek Demkowicz (UT/PECOS)
- Denis Ridzal (Sandia/CSRI)
- Pavel Bochev (Sandia/CSRI)

Support:

- Sandia (summer internships)
- PECOS (GRA funding)
- ICES (CSEM Fellowship)

From Strong-Form PDE to DPG Form

PDE

$$\triangle \phi = f$$

First-Order System

$$\nabla \cdot \psi = f$$
$$\psi - \nabla \phi = 0$$

Weak Form

$$(\psi \cdot n, v)_{\partial\Omega} - (\psi, \nabla v)_\Omega = (f, v)_\Omega$$
$$(\psi, q)_\Omega + (\phi, q \cdot n)_{\partial\Omega} - (\phi, \nabla \cdot q)_\Omega = 0$$

DPG Form

$$(\widehat{\psi}_n, v)_{\partial K} - (\psi, \nabla v)_K = (f, v)_K$$
$$(\psi, q)_K + (\widehat{\phi}, q_n)_{\partial K} - (\phi, \nabla \cdot q)_K = 0$$

$$\text{“}b(u, v) = l(v)\text{”}$$

Solving with DPG

Continuous Test Space

DPG Form

$$b(u_h, v) = l(v)$$

Optimal Test Functions

For each $u \in U_h$, find
$v_u \in V : (v_u, w)_V = b(u, w) \forall w \in V$

Discrete Test Space

DPG Form

$$b(u_h, v_h) = l(v_h)$$

Optimal Test Functions

For each $u \in U_h$, find
$v_u \in V_{p+\triangle p} : (v_u, w)_V = b(u, w)$
$\forall w \in V_{p+\triangle p}$

Stiffness Matrix

$$K_{ij} = b(e_i, v_{e_j}) = (v_{e_i}, v_{e_j})_V = (v_{e_j}, v_{e_i})_V = b(e_j, v_{e_i}) = K_{ji}$$

Optimality

$$||u - u_h||_E \leq ||u - w_h||_E \quad \forall w_h \in U_h$$

$$\left(||u||_E = \sup_{||v||_V = 1} b(u, v) = ||v_u||_V \right)$$

Goals for Camellia

Goals for Camellia (Achieved)

- Define $b(u, v)$ in the *continuous* space (separation of concerns)
- Arbitrary, hp-adaptive 2D meshes (quads and triangles)
- “Reasonable” speed and scalability
- Provide prebuilt versions of inner products commonly used in DPG research (mathematician’s and quasi-optimal)

Goals for Camellia (Aspirational)

- Support for nonlinear PDEs (Navier-Stokes in particular)
- Better scalability (distributed solution storage, distributed mesh)
- Longer term: 3D meshes

Camellia: Verification Approach

Unit Tests

- Aspire to test-driven development.
- Test individual features using e.g. simple PDEs, and building up from there.
- 7095 lines of code for unit tests (compared to 10,888 for core code).

Convergence Studies

For Poisson and two Stokes formulations (VVP and VSP), run

- h-convergence studies (triangles and quads),
- "perverse" p-refinement pattern on 16×16 mesh, and
- "hybrid" mesh of quads and triangles.
- 2708 lines of code for convergence studies.

Convergence Study: Stokes VSP

Stokes VSP, u_1 (triangular mesh)						
Mesh Size	$k=1$	rate	$k=2$	rate	$k=3$	rate
1×1	9.4e-1	-	2.6e-1	-	5.0e-2	-
2×2	3.1e-1	1.62	4.3e-2	2.63	4.0e-3	3.66
4×4	8.0e-2	1.94	5.8e-3	2.88	2.6e-4	3.95
8×8	2.0e-2	1.99	7.4e-4	2.97	1.6e-5	3.99
16×16	5.1e-3	2.00	9.3e-4	2.99	1.0e-6	4.00
32×32	1.3e-3	2.00	1.2e-5	3.00	6.4e-8	4.00

Table: L^2 error and h-convergence rates for $u_1, k=1,2,3$. We observe optimal convergence.

Variable Polynomial Orders Study

For this test, we used a 16×16 mesh and the following polynomial order pattern (repeated 4 times):

4	4	4	4	1	1	1	1	2	2	2	2	3	3	3	3
3	3	3	3	4	4	4	4	1	1	1	1	2	2	2	2
2	2	2	2	3	3	3	3	4	4	4	4	1	1	1	1
1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4

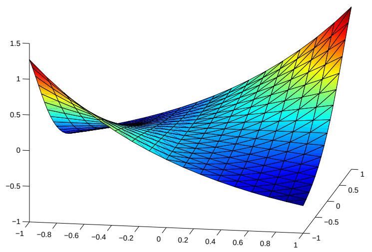
Variable Polynomial Orders Study, Poisson Results

Triangles					
ϕ		ψ_1		ψ_2	
$k=1$	mixed k	$k=1$	mixed k	$k=1$	mixed k
2.0e-3	9.1e-4	3.4e-3	1.7e-3	2.4e-3	1.1e-3

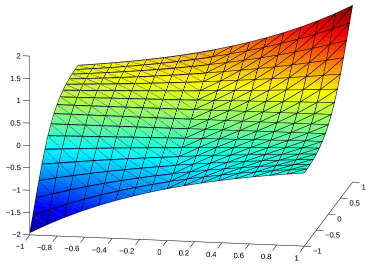
Quads					
ϕ		ψ_1		ψ_2	
$k=1$	mixed k	$k=1$	mixed k	$k=1$	mixed k
1.0e-3	3.7e-4	2.3e-3	6.6e-4	2.9e-3	1.2e-3

Poisson Manufactured Solution, "Hybrid" Cubic Mesh

ϕ

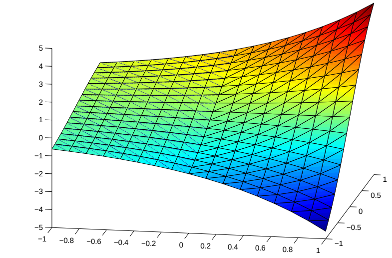


ψ_1

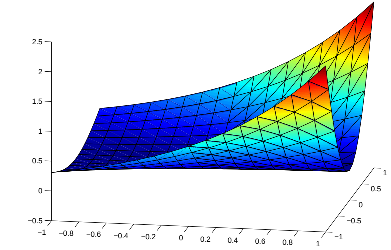


Stokes Manufactured Solution, “Hybrid” Cubic Mesh

Pressure

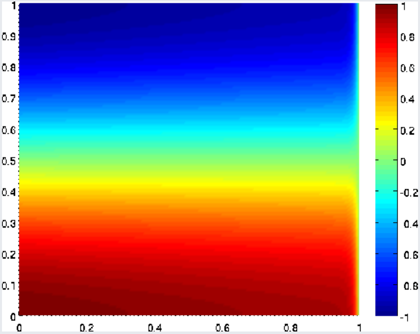


Velocity (y component)

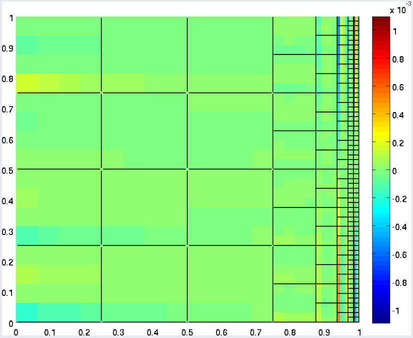


New proof of robustness in ϵ

Solution variable u



L^2 error

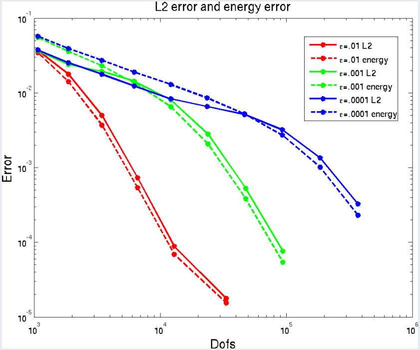


Solution/pointwise error in u for inflow boundary condition on $\beta_n u - \sigma_n$.

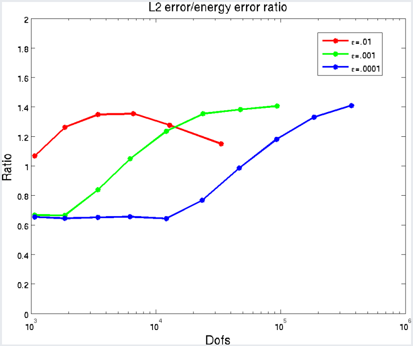
Method is provably robust (quality does not degenerate with $\epsilon \longrightarrow 0$)

New proof of robustness in ϵ

Solution variable u



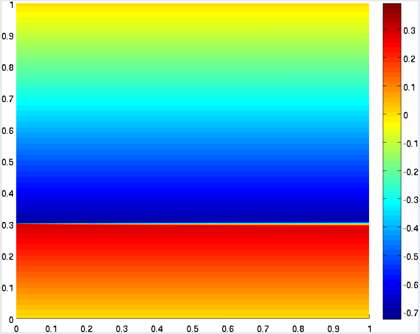
L^2 error



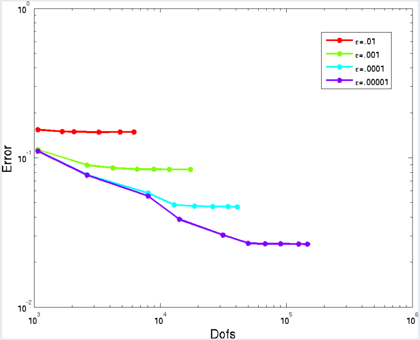
L^2 and energy error comparison, and the ratio of L^2 to energy error for inflow boundary condition on $\beta_n u - \sigma_n$. Solution is nearly identical to the L^2 projection for wide range of ϵ.

Regularization using small diffusion

Solution variable u



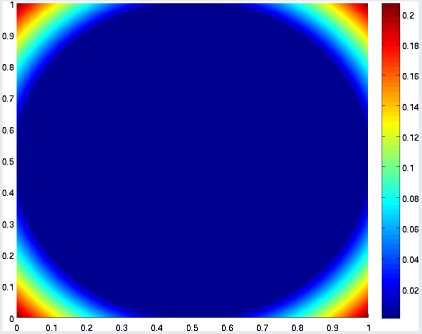
L^2 pure convection error



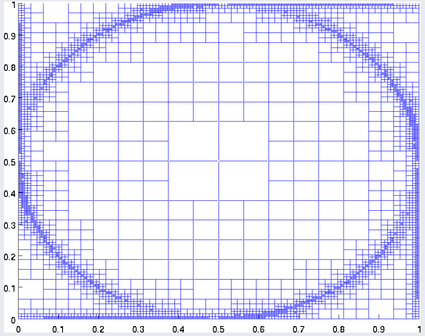
Convection of a discontinuous hat, regularized by small diffusion $\epsilon = 1e-5$. Unaligned with the mesh (worst case scenario).

Regularization of ill-posed convection problems

Solution variable u



Mesh after 5 refinement steps

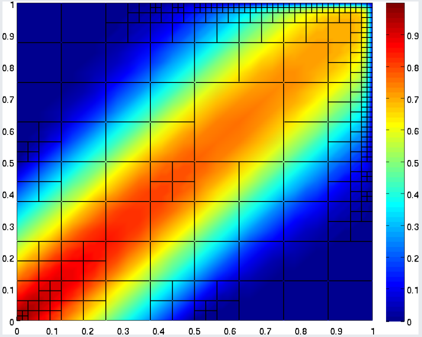


Vorticular convection problem with $\beta = (-y, x)$, regularized with $\epsilon = 1e - 5$. Ill-posed in the convection setting.

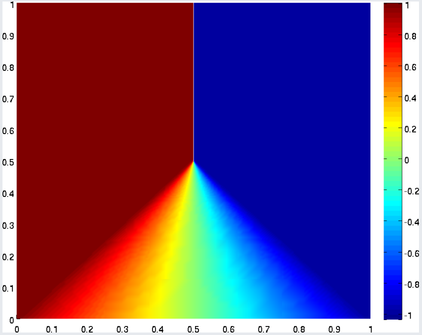
Current and Future Work

- Nonlinear PDE support
- Better scalability (distributed mesh and solution storage)
- Application to Navier-Stokes

Convection-Diffusion, $\epsilon = 10^{-2}$



Burgers', $\epsilon = 10^{-4}$



For more info on Camellia:

$\{$nroberts, jchan985$\}$@ices.utexas.edu

https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/Roberts_et_al_SAND2011-6678.pdf