SAND2015-5148C

BDDC Algorithms for Discontinuous Petrov Galerkin Methods

Clark Dohrmann

**23rd International Conference
on Domain Decomposition Methods
Jeju Island, Korea
July 6-10, 2015**

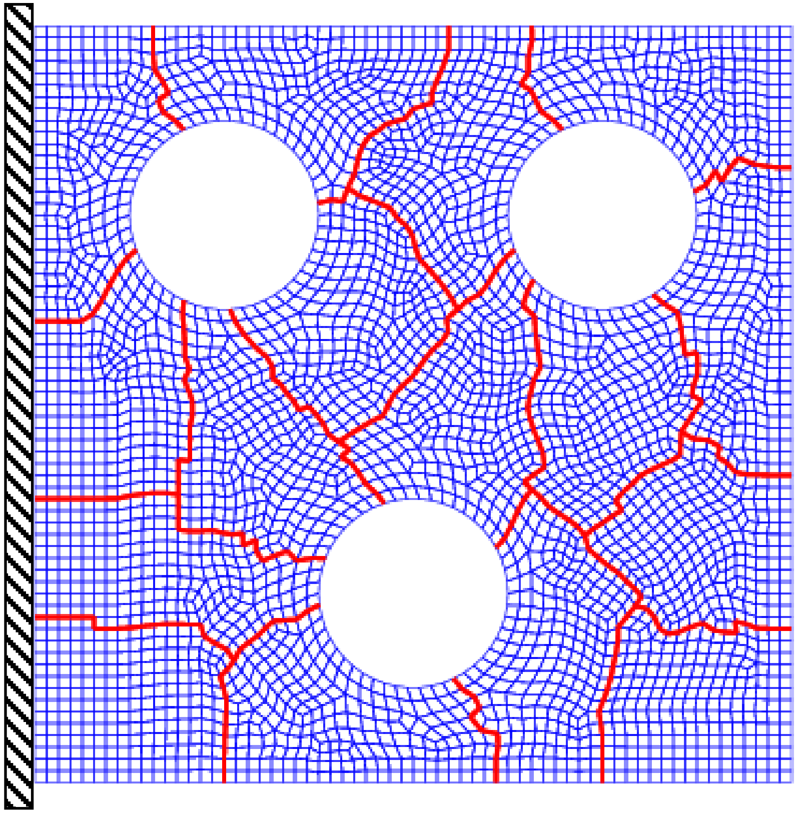
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline

- **Background:**
  - Balancing Domain Decomposition by Constraints (BDDC)
  - Discontinuous Petrov Galerkin (DPG) methods
  - Why the interest?
- **Model Problem:**
  - Initial disappointing results
  - A remedy and some connections
- **Modified BDDC Preconditioner:**
  - What adaptive constraint algorithms tell us
- **Applications**
- **Closing Remarks**

Iterative Substructuring

interior I shown in blue
interface Γ shown in red



$$\begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} x_I \\ u \end{bmatrix} = \begin{bmatrix} b_I \\ b_\Gamma \end{bmatrix}$$

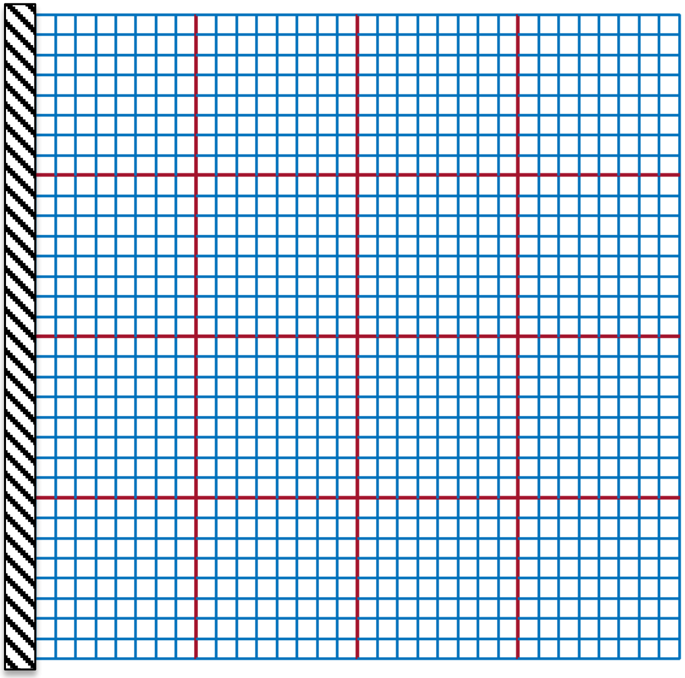
- restrict problem to interface*
- interface unknowns:
  - Lagrange multipliers, FETI-DP
  - original unknowns, BDDC
- precondition interface problem:
  - local “subdomain” solves
  - global “coarse” solve
- No need to form Schur complement

$$\boxed{\bar{S}u = g}$$

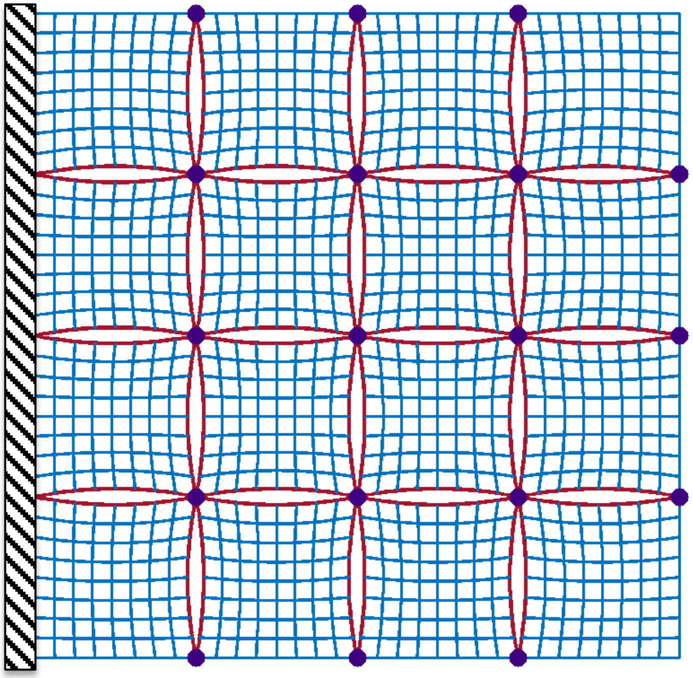
$$\bar{S} = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{I\Gamma}$$

*see P63

BDDC Preconditioner (pictures)



fully assembled model



partially assembled model*

Basic components: decomposition, constraints, weights

FETI-DP connection

Note: solving partially assembled problem equivalent to solving independent local subdomain problems and adding coarse solution

*see LW06

BDDC Preconditioner (equations)

Recall we want to solve $\bar{S}u = g$

Bookkeeping: $u_i = R_i u$, R: row concatenation of $R_1, \ldots, R_N$

Weights: $D = \mathrm{diag}(D_1, \ldots, D_n)$, $\quad R^T D R = I$ (partition of unity)

Schur Complement: $S = \mathrm{diag}(S_1, \ldots, S_N)$, $\quad \bar{S} = R^T S R$

Constraints: enforced in subdomain corrections $M_1^{-1}, \ldots, M_N^{-1}$.

Subdomain Correction: $T_{sub} = \mathrm{diag}(M_1^{-1}, \ldots, M_N^{-1})$

Coarse Correction: $T_0 = \Psi(\Psi^T S \Psi)^{-1}\Psi^T$, $\quad \Psi$: E-M coarse basis

BDDC Preconditioner: $\boxed{M_{BDDC}^{-1} = R^T D^T (T_{sub} + T_0) D R}$

Alternative*: $M_{BDDC} = \tilde{R}_D^T \tilde{S}^{-1} \tilde{R}_D$ (nice connection to picture)

Partially assembled Schur complement

*see LW06

BDDC: A Few More Comments

- **Constraints (Coarse Space):**
  - Simple sometimes (2D corners only, 3D maybe edges/faces too)
  - Not always simple
    - Material property jumps, almost incompressible materials, …
    - Active research area (several talks this week)
    - Can work with localized version of BDDC estimate*

$$\kappa_{BDDC} \leq \sup_{u \in \widetilde{W}} \frac{||P_D u||_S}{||u||_S}$$

- **Weights:**
  - Cardinality, stiffness, “rho”, deluxe, e-deluxe, …
  - Story not over yet
- **Applications:**
  - Elasticity, plates/shells, porous media, Stokes, H(curl), H(div), …
  - Today: Discontinuous Petrov Galerkin Methods

*see, e.g., MDT05

What is DPG?

- **Discontinuous Petrov Galerkin Method:**
  - Introduced by L. Demkowicz and J. Gopalakrishnan*
  - Why discontinuous in name?
    - Test functions for elements completely independent from each other
    - “Field” trial functions in L^2, but eliminated at element level
    - “Numerical trace” and “numerical flux” trial functions are continuous across element boundaries
  - Why Petrov Galerkin in name?
    - Test functions are different than trial functions
    - Test functions obtained by solving independent optimization problem local to each element
    - Test functions from higher dimensional space than trial functions
  - Anything else?
    - works with first-order form of equations
    - derivatives commonly moved from trial to test functions in all equations via integration by parts (“ultra-weak formulation”)
    - element matrices are symmetric and positive semidefinite

*see, e.g., DG11 and references therein

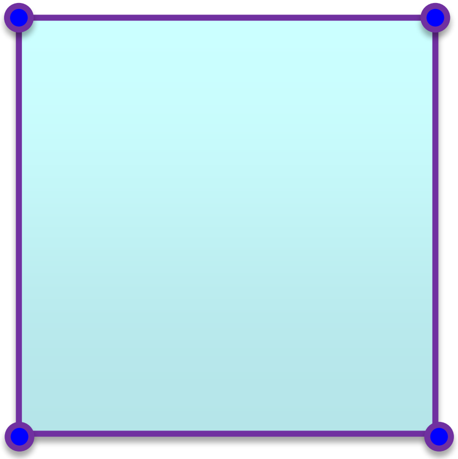
Why the Interest in DPG?

- **Discretization:**
  - Continuous trial functions on element boundaries
    - Potential for more flexibility in meshing complex geometries (e.g. polyhedral elements)
  - Favorable error estimates
    - Potential for higher accuracy solutions. For example, displacements and stresses elasticity with comparable convergence rates
  - Attractive results for wave propagation problems*
    - Good pre-asymptotic stability
    - Reported negligible phase/pollution errors (at least for higher-order)
  - Active research area for many applications
  - Potentially well-suited for next generation computing platforms
- **Solvers:**
  - SPSD element matrices (rich DD theory available)
  - Non-symmetric or indefinite systems possibly more accessible
  - New area, potential of adaptive coarse spaces algorithms
  - Q: who could possibly resist?

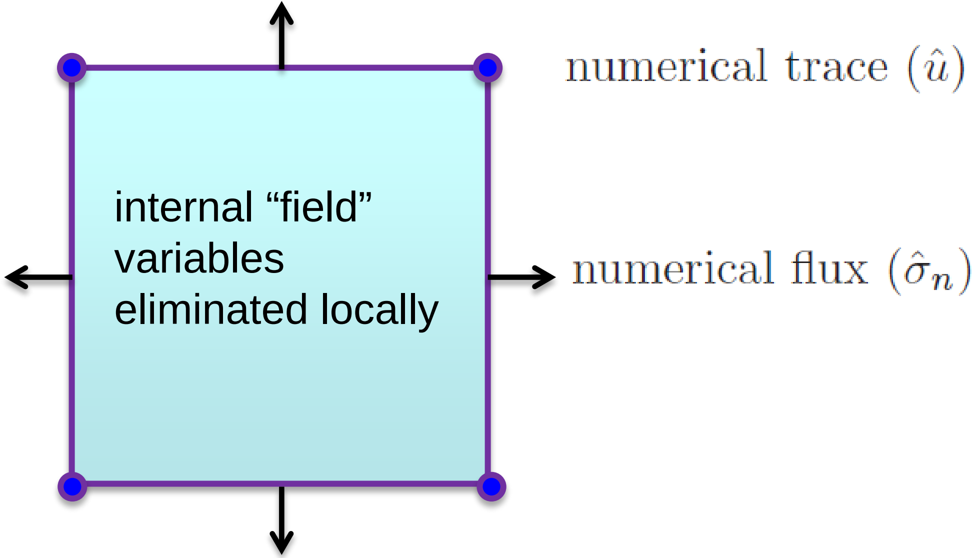
*see DGMZ12

A Quick Look Ahead

Model Problem: Poisson equation



standard element



numerical trace $(\hat{u})$

internal “field” variables eliminated locally

numerical flux $(\hat{\sigma}_n)$

DPG element

Observations:

- More unknowns for DPG ($\approx$ 3x in 2D and $\approx$ 4x in 3D)
- Code more complex (6x even when excluding additional functions*)
- Remaining shape functions only needed on boundary (more flexibility)
- Not for the faint of heart
- Is it worth it? Well, it depends

*codes like Camellia (R14) can reduce complexity, but more worked needed here

DPG Formulation*

- **Starting point:**

$$\begin{aligned} -\vec{\nabla}\cdot(\alpha\vec{\nabla}u) &= f && \text{on } \Omega,\\ u &= 0 && \text{on } \partial\Omega. \end{aligned}$$

- **First order form:**

$$\begin{aligned} \alpha^{-1}\vec{\sigma} + \vec{\nabla}u &= 0, && \vec{\sigma} = -\alpha\vec{\nabla}u\\ \vec{\nabla}\cdot\vec{\sigma} &= f, \end{aligned}$$

- **Integration by parts:**

$$\begin{aligned} (\alpha^{-1}\vec{\sigma},\vec{\tau})_K - (u,\vec{\nabla}\cdot\vec{\tau})_K + \langle u,\vec{\tau}\cdot\vec{n}\rangle_{\partial K} &= 0 && \forall\vec{\tau}\in H(\mathrm{div},K),\\ -(\vec{\sigma},\vec{\nabla}v)_K + \langle v,\vec{\sigma}\cdot\vec{n}\rangle_{\partial K} &= (f,v)_K && \forall v\in H^1(K). \end{aligned}$$

*see DG11. Goal here: provide quick startup for implementation if interested

DPG Formulation (continued)

- **Integration by parts:**

$$(\alpha^{-1}\vec{\sigma},\vec{\tau})_K - (u,\vec{\nabla}\cdot\vec{\tau})_K + \langle u,\vec{\tau}\cdot\vec{n}\rangle_{\partial K} = 0 \qquad \forall\vec{\tau}\in H(\mathrm{div},K),$$

$$-(\vec{\sigma},\vec{\nabla}v)_K + \langle v,\vec{\sigma}\cdot\vec{n}\rangle_{\partial K} = (f,v)_K \qquad \forall v\in H^1(K).$$

- **Ultra-weak formulation:**

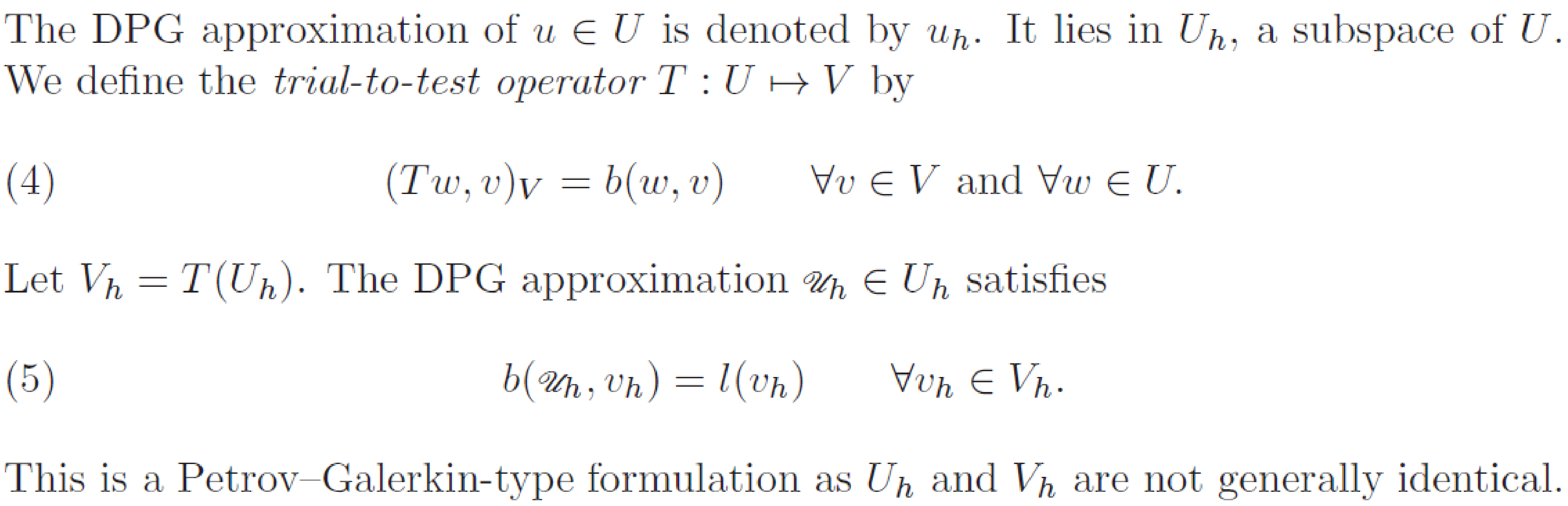
We now replace the terms $\langle u,\vec{\tau}\cdot\vec{n}\rangle_{\partial K}$ and $\langle v,\vec{\sigma}\cdot\vec{n}\rangle_{\partial K}$ by $\langle \hat{u},\vec{\tau}\cdot\vec{n}\rangle_{1/2,\partial K}$ and $\langle v,\hat{\sigma}_n\rangle_{1/2,\partial K}$, respectively, where $\hat{u}$ and $\hat{\sigma}_n$ are new unknowns, and $\langle\cdot,\ell\rangle_{1/2,\partial K}$ denotes the action of a functional ℓ in $H^{-1/2}(\partial K)$. This motivates the following ultra-weak formulation:

$$(12)\qquad \begin{cases} \text{Find } (\vec{\sigma},u,\hat{u},\hat{\sigma}_n)\in U \text{ such that} \\ \qquad b((\vec{\sigma},u,\hat{u},\hat{\sigma}_n),\ (\vec{\tau},v)) = l(\vec{\tau},v) \qquad \forall(\vec{\tau},v)\in V, \end{cases}$$

- **Shorthand notation:**

$$\begin{cases} \text{Find } \mathscr{U}\in U \text{ such that} \\ b(\mathscr{U},v) = l(v) \quad \forall v\in V. \end{cases}$$

DPG Formulation (almost there)



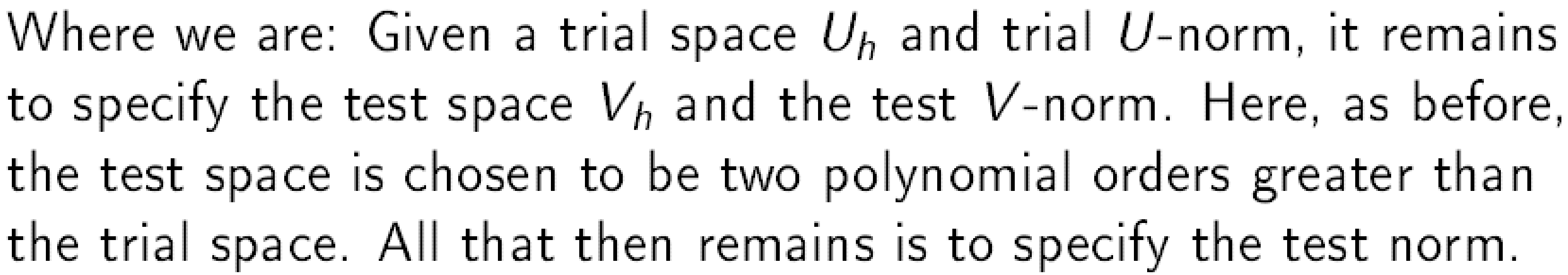
The DPG approximation of $u \in U$ is denoted by u_h. It lies in U_h, a subspace of U. We define the *trial-to-test operator* $T : U \mapsto V$ by

$$(Tw, v)_V = b(w, v) \qquad \forall v \in V \text{ and } \forall w \in U. \tag{4}$$

Let $V_h = T(U_h)$. The DPG approximation $\mathscr{u}_h \in U_h$ satisfies

$$b(\mathscr{u}_h, v_h) = l(v_h) \qquad \forall v_h \in V_h. \tag{5}$$

This is a Petrov–Galerkin-type formulation as U_h and V_h are not generally identical.



Where we are: Given a trial space U_h and trial U-norm, it remains to specify the test space V_h and the test V-norm. Here, as before, the test space is chosen to be two polynomial orders greater than the trial space. All that then remains is to specify the test norm.

DPG Formulation (we're there!)

If we want the DPG solution to coincide with the best approximation from U_h in the U-norm, then choose

$$\|v\|_V = \|v\|_{opt,V} = \sup_{u \in U_h} \frac{b(u,v)}{\|u\|_U}$$

Unfortunately, the optimal test norm is difficult to compute with, and a "broken" or "localizable" test space norm is used in its place. It is shown in DG11 that the simpler test norm

$$\|(\vec{\tau}, v)\|_V^2 = \|\vec{\tau}\|^2_{H(div,\Omega_h)} + \|v\|^2_{H^1(\Omega_h)}$$

is equivalent to the optimal test norm for the Poisson equation. We use this test norm here.

DPG in a Nutshell

- Choose trial space
- Choose basis for test space (often enrichment of trial space)
- Calculate test functions associated with each trial function using trial-to-test operator
- Trial-Test norm pairs available in literature for variety of applications
- Standard Petrov Galerkin FEM at this point

DPG Element Stiffness Matrix

Recall $V_h = T(U_h)$ with trial-to-test operator T given by

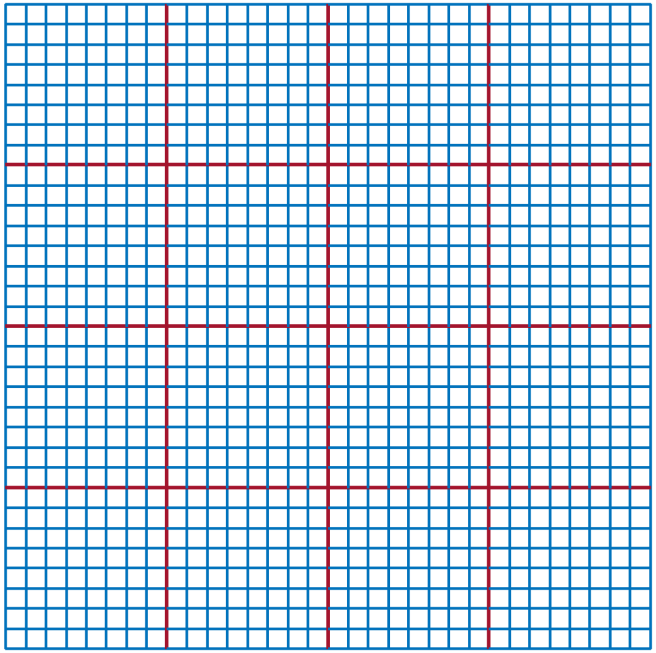
$$\begin{aligned} (Tw, v)_V &= b(w, v) \quad \forall v \in V_h \quad \textit{and} \quad w \in U_h \\ v^H G T w &= v^H B w \\ TW &= G^{-1} B w \\ (Tw)^H B u &= w^H \underbrace{B^H G^{-1} B}_{\substack{\text{stiffness} \\ \text{matrix}}} u \end{aligned}$$

Least squares connection: minimize $(Bu - f)^H G^{-1} (Bu - f) \Rightarrow$

$$\begin{aligned} B^H G^{-1}(Bu - f) &= 0 \\ \underbrace{B^H G^{-1} B}_{\substack{\text{stiffness} \\ \text{matrix}}} u &= \underbrace{B^H G^{-1} f}_{\substack{\text{element} \\ \text{load}}} \end{aligned}$$

Note: matrix includes “field” variables initially. Use static condensation to remove.

Let’s give it a try



essential BCs on entire boundary
4 x 4 subdomains
corners only coarse space*
H/h = 8
unit length
Iterations = 135
condition number > 1.5 x 10^4

What just happened?

Flashback (humor): “Whatever you do, do not apply essential and natural BCs to the same part of boundary”. “Okay dad, can we go play golf now?”
What next?

- Optimized Schwarz, overlapping Schwarz?
- **Take a step back**

* Adding average flux component to coarse space didn’t help much

A step back (first piece of puzzle)

Same problem, but no subdomains

$$A = \begin{bmatrix} A_{pp} & A_{pf} \\ A_{fp} & A_{ff} \end{bmatrix} \quad A_D = \begin{bmatrix} A_{pp} & 0 \\ 0 & A_{ff} \end{bmatrix}$$

Consider the eigenproblem $A_D x = \lambda A x$ for different meshes

1/h	λ_{min}	λ_{max}
5	0.8724	1.3427
10	0.8662	1.3653
15	0.8651	1.3695
20	0.8647	1.3710
25	0.8646	1.3717

A step back (second piece of puzzle)

$$K_{elem} = \begin{bmatrix} K_{pp} & K_{pf} \\ K_{fp} & K_{ff} \end{bmatrix}$$

$$K_{pp} = \begin{matrix} 0.4170 & -0.1667 & -0.0836 & -0.1667 \\ -0.1667 & 0.4170 & -0.1667 & -0.0836 \\ -0.0836 & -0.1667 & 0.4170 & -0.1667 \\ -0.1667 & -0.0836 & -0.1667 & 0.4170 \end{matrix}$$

$$a(u, v) = \int_K \nabla u \cdot \nabla v \, dx$$

$$K_{ff} = \begin{matrix} 1.0021 & 1.0000 & 0.9996 & 1.0000 \\ 1.0000 & 1.0021 & 1.0000 & 0.9996 \\ 0.9996 & 1.0000 & 1.0021 & 1.0000 \\ 1.0000 & 0.9996 & 1.0000 & 1.0021 \end{matrix}$$

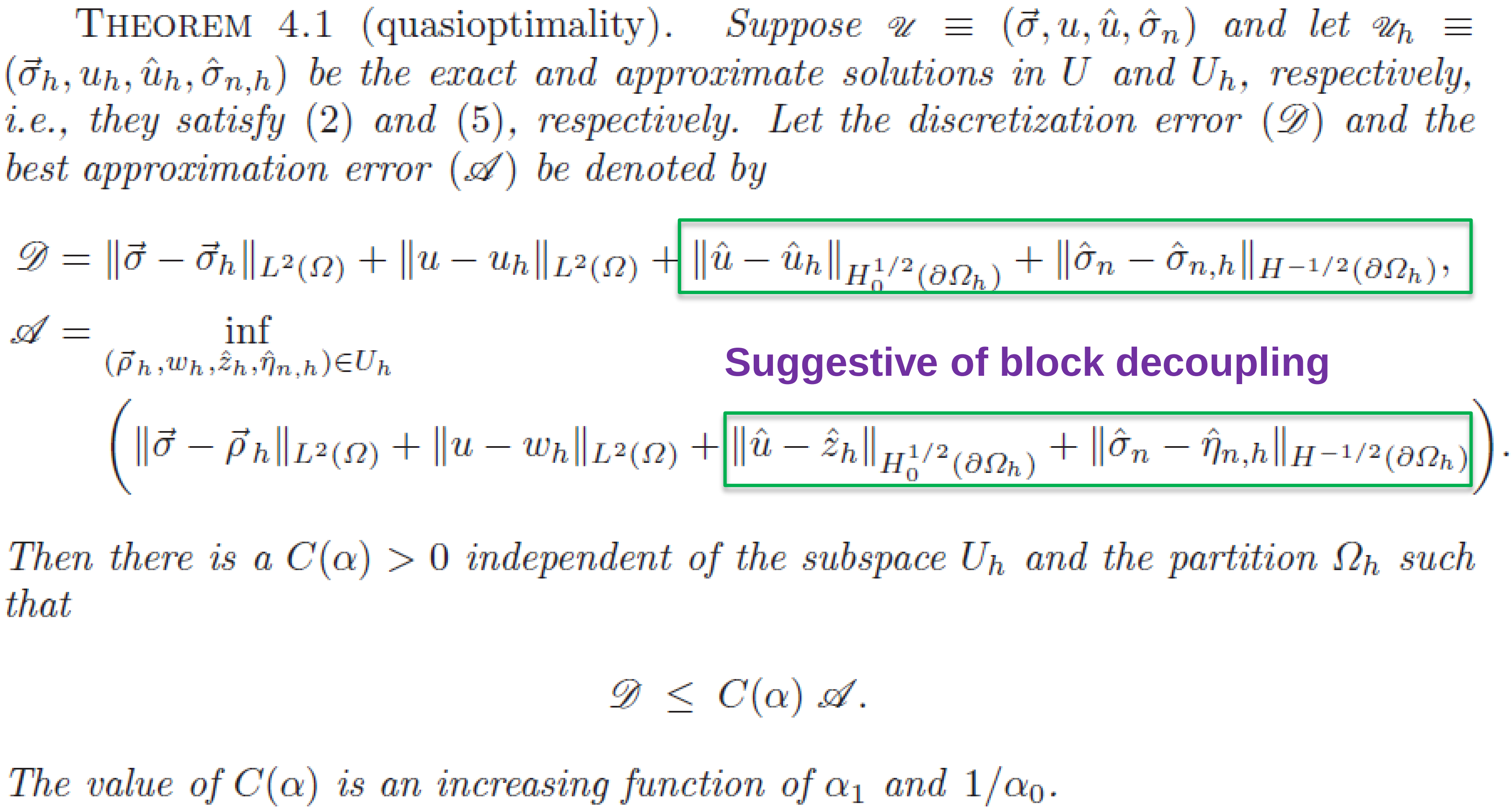
$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_K (\alpha(\nabla \cdot \boldsymbol{u})(\nabla \cdot \boldsymbol{v}) + \beta \boldsymbol{u} \cdot \boldsymbol{v}) \, dx$$

Note: nice BDDC preconditioner for R-T vector fields thanks to Oh & Widlund (OWD13)

Some Connections

- Related BDDC preconditions for porous media:
  - Mixed formulation rather than least squares
    - T05, T07
    - Tuesday talk in MS5-2
- Least squares connection:
  - Preconditioner strategies based on block diagonal matrices, see, e.g., BG09
- In hindsight, DPG error estimate insightful:
  - See following slide

DPG Error Estimate



THEOREM 4.1 (quasioptimality). *Suppose* $\mathscr{U} \equiv (\vec{\sigma}, u, \hat{u}, \hat{\sigma}_n)$ *and let* $\mathscr{U}_h \equiv (\vec{\sigma}_h, u_h, \hat{u}_h, \hat{\sigma}_{n,h})$ *be the exact and approximate solutions in* U *and* U_h*, respectively, i.e., they satisfy* (2) *and* (5)*, respectively. Let the discretization error* ($\mathscr{D}$) *and the best approximation error* ($\mathscr{A}$) *be denoted by*

$$\mathscr{D} = \|\vec{\sigma} - \vec{\sigma}_h\|_{L^2(\Omega)} + \|u - u_h\|_{L^2(\Omega)} + \boxed{\|\hat{u} - \hat{u}_h\|_{H_0^{1/2}(\partial\Omega_h)} + \|\hat{\sigma}_n - \hat{\sigma}_{n,h}\|_{H^{-1/2}(\partial\Omega_h)},}$$

$$\mathscr{A} = \inf_{(\vec{\rho}_h, w_h, \hat{z}_h, \hat{\eta}_{n,h}) \in U_h}$$

Suggestive of block decoupling

$$\left(\|\vec{\sigma} - \vec{\rho}_h\|_{L^2(\Omega)} + \|u - w_h\|_{L^2(\Omega)} + \boxed{\|\hat{u} - \hat{z}_h\|_{H_0^{1/2}(\partial\Omega_h)} + \|\hat{\sigma}_n - \hat{\eta}_{n,h}\|_{H^{-1/2}(\partial\Omega_h)}} \right).$$

Then there is a $C(\alpha) > 0$ *independent of the subspace* U_h *and the partition* Ω_h *such that*

$$\mathscr{D} \leq C(\alpha)\, \mathscr{A}.$$

The value of $C(\alpha)$ *is an increasing function of* α_1 *and* $1/\alpha_0$.

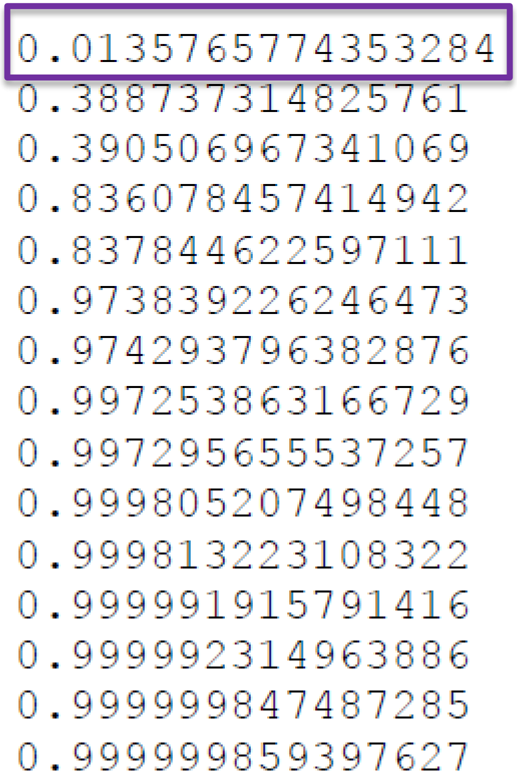
*see again DG11

Modified BDDC Preconditioner

- Takes advantage of matrix properties:
  - We already have good BDDC preconditioners for
    - Poisson equation
    - H(div) problems
  - Can we combine the two at once?
    - Yes, by basing preconditioner on A_D instead of A
  - Sounds pretty easy, but still need to solve original problem
    - No problem, just need to apply Schur complement of original matrix during conjugate gradient iterations
  - Will we still have an interface preconditioner?
    - Yes, interior unknowns are still eliminated based on A
- Summary:
  - Preconditioner based on block diagonal matrix A_D
  - Matrix A still used for initial elimination of interior unknowns and for application of Schur complement during iterations

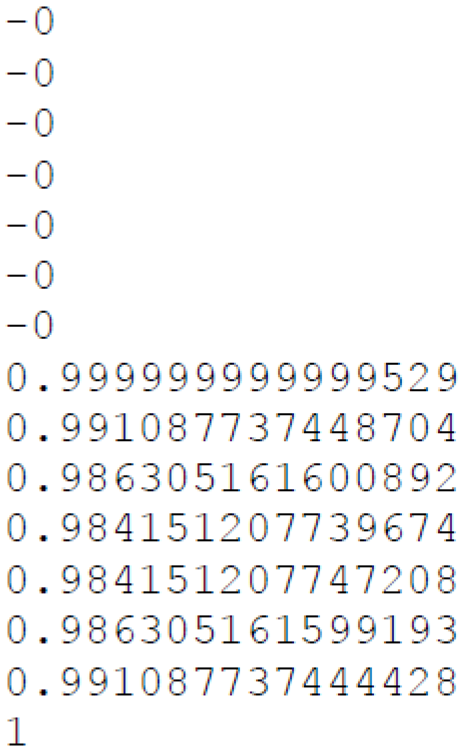
What Adaptive Algorithms Tell Us

4 x 4 subdomains, H/h = 8, corners only coarse space



```
0.0135765774353284
0.388737314825761
0.390506967341069
0.836078457414942
0.837844622597111
0.973839226246473
0.974293796382876
0.997253863166729
0.997295655537257
0.999805207498448
0.999813223108322
0.999991915791416
0.999992314963886
0.999999847487285
0.999999859397627
```

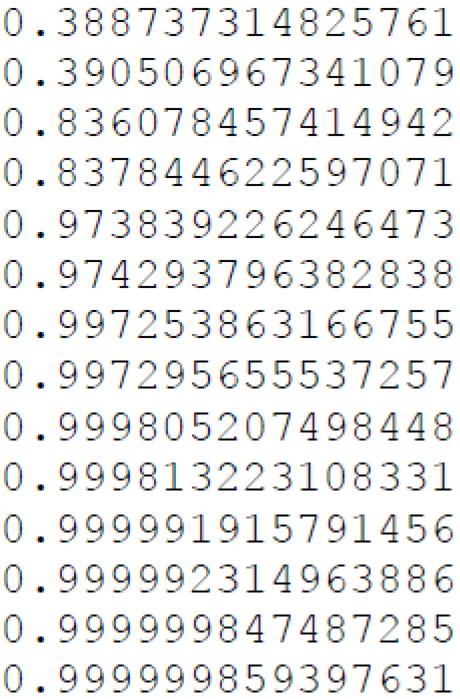
Eigenvalues associated with edge (small is bad)



```
-0
-0
-0
-0
-0
-0
-0
0.999999999999529
0.991087737448704
0.986305161600892
0.984151207739674
0.984151207747208
0.986305161599193
0.991087737444428
1
```

flux dofs

Edge constraint for smallest eigenvalue



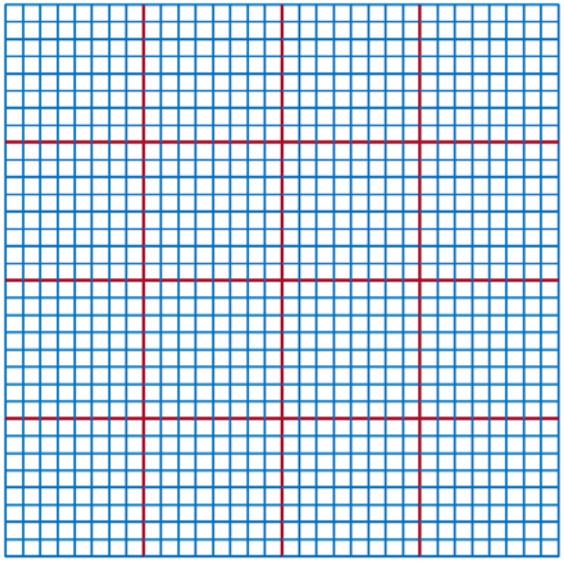
```
0.388737314825761
0.390506967341079
0.836078457414942
0.837844622597071
0.973839226246473
0.974293796382838
0.997253863166755
0.997295655537257
0.999805207498448
0.999813223108331
0.999991915791456
0.999992314963886
0.999999847487285
0.999999859397631
```

New edge eigenvalues after constraint

Including average flux constraint for edges should really help reduce condition #

Poisson Equation (constant properties)

N x N subdomains, corners + average flux coarse space

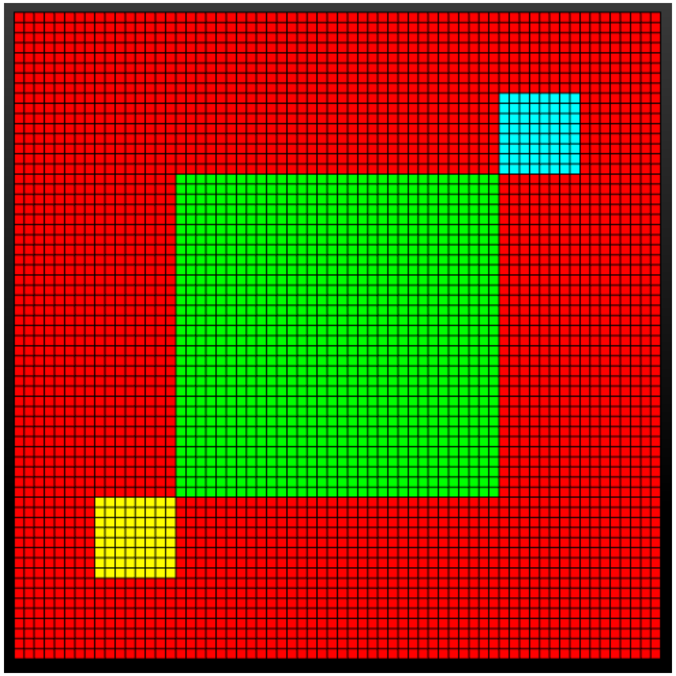


N = 4, H/h = 8

N	H/h = 4	H/h = 8	H/h = 12	H/h = 16
4	2.10	2.84	3.34	3.72
6	2.18	2.98	3.52	3.93
8	2.20	3.02	3.57	3.99
10	2.20	3.03	3.58	4.01

Poisson Equation (variable properties)

8 x 8 subdomains, corners + average flux coarse space



α = 1 in red region, $\alpha = \sigma$ in other regions, H/h = 8

Much larger sensitivity to material property jumps than what we've seen for the second order Poisson equation. Based on adaptive algorithms, difficulties appear associated with flux variables. Scaling of trial and/or test norms may be able to address problem.

Elasticity

- Error estimates*:
  - Suggestive of decoupling between displacements and fluxes as for the Poisson equation
  - Analogous trial/test norms and error estimates to Poisson
  - More accurate stresses for smooth solutions
  - Locking-free convergence properties in the case of almost incompressible elasticity (AIE) problems
- Preconditioner:
  - Need to confirm, but expect only flux constraints on faces in 3D needed for AIE
    - Simplification over “standard” approach where edge constraints must also be considered

*see BDGQ12

Closing Remarks

- **Standard BDDC preconditioner applied to DPG model problem performed poorly**
- **Modified BDDC preconditioner performed well**
  - Based on block diagonal approximation of operator matrix
  - Average flux constraint needed for edges in 2D and faces in 3D
  - Still an interface preconditioner, but existing theory now only applies to block matrix
- **Many questions remain (lots of opportunities here!)**
  - How to deal with material property jumps, Helmholtz problem performance, theory development, coarse spaces, …
- **High performance computing future of DPG**
  - Effective domain decomposition or multigrid solvers a must
  - Software tools needed to make DPG much more accessible

Extra Slides

References

- BDGQ12: J. Bramwell, L. Demkowiz, J. Gopalakrishnan and W. Qiu, "A locking-free hp DPG method for linear elasticity with symmetric stresses", Numer. Math. Vol. 122, pp. 671-707, 2012.
- BG09: P. Bochev and M. Gunzburger, Least-Squares Finite Element Methods, Springer, Applied Mathematical Sciences 166, 2009.
- D07: C. Dohrmann, "An approximate BDDC preconditioner", Numer. Linear Algebra Appl., Vol 14, pp. 149-168, 2007.
- DG11: L. Demkowicz and J.Gopalakrishnan, "Analysis of the DPG method for the Poisson equation", SIAM J. Numer. Anal., Vol. 49, pp. 1788-1809. 2011.
- DGMZ: L. Demkowicz, J. Gopalakrishnan, I. Muga and J. Zitelli, "Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation", Comput. Methods Appl. Mech. Engrg., pp. 126-138, 2012.
- LW06: J. Li and O. Widlund, "FETI-DP, BDDC, and block Cholesky methods", Int. J. Numer. Meth. Engng, Vol. 66, pp. 250-271, 2006.

References

- LW07: J. Li and O. Widlund, “On the use of inexact subdomain solvers for BDDC algorithms”, Comput. Methods Appl. Mech. Engrg., Vol. 196, pp. 1415-1428, 2007.
- MDT05: J. Mandel, C. Dohrmann and R. Tezaur, “An algebraic theory for primal and dual substructuring methods by constraints”, Applied Numerical Mathematics, Vol. 54, pp. 167-193, 2005.
- O11: D. Oh, O. Widlund and C. Dohrmann, “A BDDC algorithm for Raviart-Thomas vector fields”, Technical Report TR-951, Courant Institute of Mathematical Sciences, 2013.
- P63: Przemieniecki, J., “Matrix Structural Analysis of Substructures”, AIAA Journal, Vol. 1, pp. 138-147, 1963.
- R14: N. Roberts, “Camellia: A software framework for discontinuous Petrov-Galerkin methods”, Vol. 68, pp. 1581-1604, 2014.
- J. Schoberl, J. Melenk, C. Pechstein and S. Zaglmayr, “Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements”, Vol. 28, pp. 1-24, 2008.
- T05: X. Tu, “A BDDC algorithm for a mixed formulation of flows in porous media”, Electron. Trans. Numer. Anal., Vol. 20, pp. 164-179, 2005.

References

- T07: X. Tu, “A BDDC algorithm for flow in porous media with a hybrid finite element discretization”, Electron. Trans. Numer. Anal., Vol 26, pp. 146-160, 2007.

Higher Order DPG

- **Greater accuracy but much more demanding on sparse direct solvers**
- **Replace direct solvers with more memory efficient preconditioners (see, e.g., SMPZ2007)**
- **Replace standard BDDC algorithm with approximate one (see, e.g., LW07 and D07)**