BDDC Domain Decomposition Algorithms

Olof B. Widlund
Courant Institute, New York University

75th Anniversary of
Mathematics of Computation
November 3, 2018

Problems considered

- BDDC stands for Balancing Domain Decomposition by Constraints and this family of algorithms was introduced by Clark Dohrmann in 2003 following the introduction of the FETI–DP (Dual Primal Finite Element Tearing and Interconnecting) algorithms by Charbel Farhat et al in 2000. Dohrmann remains a main provider of ideas and analysis.

Problems considered

- BDDC stands for Balancing Domain Decomposition by Constraints and this family of algorithms was introduced by Clark Dohrmann in 2003 following the introduction of the FETI–DP (Dual Primal Finite Element Tearing and Interconnecting) algorithms by Charbel Farhat et al in 2000. Dohrmann remains a main provider of ideas and analysis.
- In this talk, I will introduce the basics of BDDC, give examples of successful applications and talk about some recent work.

Problems considered

- BDDC stands for Balancing Domain Decomposition by Constraints and this family of algorithms was introduced by Clark Dohrmann in 2003 following the introduction of the FETI–DP (Dual Primal Finite Element Tearing and Interconnecting) algorithms by Charbel Farhat et al in 2000. Dohrmann remains a main provider of ideas and analysis.
- In this talk, I will introduce the basics of BDDC, give examples of successful applications and talk about some recent work.
- My last few PhD students were all involved in the development of the BDDC family. I have also worked with Dohrmann and with Beirão da Veiga, Pavarino, Scacchi, Zampini, Oh, and Calvo. Recently, the focus has been on small coarse problems for BDDC, adaptive choices of the coarse problems, and isogeometric analysis problems for elasticity including the almost incompressible case..

Problems considered

- BDDC domain decomposition algorithms for finite element approximations for a variety of elliptic problems with very many degrees of freedom.

Problems considered

- BDDC domain decomposition algorithms for finite element approximations for a variety of elliptic problems with very many degrees of freedom.
- Among applications, elasticity, problems formulated in $H(\mathbf{curl})$, $H(\mathrm{div})$, and Reissner-Mindlin plates.

Problems considered

- BDDC domain decomposition algorithms for finite element approximations for a variety of elliptic problems with very many degrees of freedom.
- Among applications, elasticity, problems formulated in $H(\mathbf{curl})$, $H(\mathrm{div})$, and Reissner-Mindlin plates.
- Mostly lowest order finite element methods for selfadjoint elliptic problems but we have also helped develop solvers for isogeometric analysis also for higher order methods.

Problems considered

- BDDC domain decomposition algorithms for finite element approximations for a variety of elliptic problems with very many degrees of freedom.
- Among applications, elasticity, problems formulated in $H(\mathbf{curl})$, $H(\mathrm{div})$, and Reissner-Mindlin plates.
- Mostly lowest order finite element methods for selfadjoint elliptic problems but we have also helped develop solvers for isogeometric analysis also for higher order methods.
- All this aims at developing preconditioners for the stiffness matrices. These approximate inverses are then combined with conjugate gradients or other Krylov space methods.

Problems considered

- BDDC domain decomposition algorithms for finite element approximations for a variety of elliptic problems with very many degrees of freedom.
- Among applications, elasticity, problems formulated in $H(\mathbf{curl})$, $H(\text{div})$, and Reissner-Mindlin plates.
- Mostly lowest order finite element methods for selfadjoint elliptic problems but we have also helped develop solvers for isogeometric analysis also for higher order methods.
- All this aims at developing preconditioners for the stiffness matrices. These approximate inverses are then combined with conjugate gradients or other Krylov space methods.
- Primarily interested in hard problems with very many subdomains and having convergence rates independent of that number and with rates that decrease slowly with the size of the subdomain problems. Many bounds independent of jumps in coefficients between subdomains.

BDDC, finite element meshes, and equivalence classes

- BDDC algorithms work on decompositions of the domain Ω of the elliptic problem into non-overlapping subdomains Ω_i, each often with tens of thousands of degrees of freedom. In between the subdomains the interface Γ. The local interface of Ω_i: $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. Γ does not cut any elements.

BDDC, finite element meshes, and equivalence classes

- BDDC algorithms work on decompositions of the domain Ω of the elliptic problem into non-overlapping subdomains Ω_i, each often with tens of thousands of degrees of freedom. In between the subdomains the interface Γ. The local interface of Ω_i: $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. Γ does not cut any elements.
- Many of the finite element nodes are interior to individual subdomains while others belong to several subdomain interfaces.

BDDC, finite element meshes, and equivalence classes

- BDDC algorithms work on decompositions of the domain Ω of the elliptic problem into non-overlapping subdomains Ω_i, each often with tens of thousands of degrees of freedom. In between the subdomains the interface Γ. The local interface of Ω_i: $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. Γ does not cut any elements.
- Many of the finite element nodes are interior to individual subdomains while others belong to several subdomain interfaces.
- The nodes on Γ are partitioned into equivalence classes of sets of indices of the local interfaces Γ_i to which they belong. For 3D and nodal finite elements, we have classes of face nodes, associated with two local interfaces, and classes of edge nodes and subdomain vertex nodes.

BDDC, finite element meshes, and equivalence classes

- BDDC algorithms work on decompositions of the domain Ω of the elliptic problem into non-overlapping subdomains Ω_i, each often with tens of thousands of degrees of freedom. In between the subdomains the interface Γ. The local interface of Ω_i: $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. Γ does not cut any elements.
- Many of the finite element nodes are interior to individual subdomains while others belong to several subdomain interfaces.
- The nodes on Γ are partitioned into equivalence classes of sets of indices of the local interfaces Γ_i to which they belong. For 3D and nodal finite elements, we have classes of face nodes, associated with two local interfaces, and classes of edge nodes and subdomain vertex nodes.
- For $H(\mathbf{curl})$ and Nédélec (edge) elements, only equivalence classes of element edges on subdomain faces and on subdomain edges. For $H(\text{div})$ and Raviart-Thomas elements, only degrees of freedom for element faces.

Partial assembly

- These equivalence classes play a central role in the design, analysis, and programming of domain decomposition methods.

Partial assembly

- These equivalence classes play a central role in the design, analysis, and programming of domain decomposition methods.
- The BDDC and FETI–DP families are related algorithmically and have a common theoretical foundation. They are based on using partially subassembled stiffness matrices assembled from the subdomain stiffness matrices $A^{(i)}$. We will first look at a simple 2D nodal finite element problem.

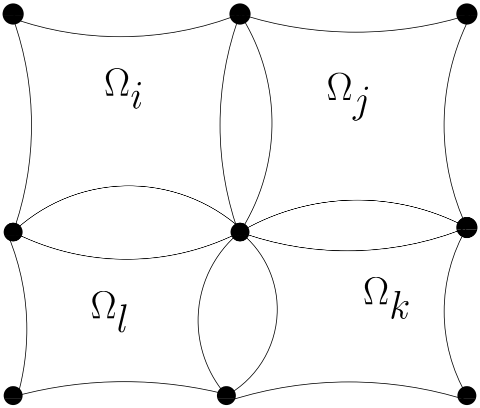
Partial assembly

- These equivalence classes play a central role in the design, analysis, and programming of domain decomposition methods.
- The BDDC and FETI–DP families are related algorithmically and have a common theoretical foundation. They are based on using partially subassembled stiffness matrices assembled from the subdomain stiffness matrices $A^{(i)}$. We will first look at a simple 2D nodal finite element problem.
- The nodes of $\Omega_i \cup \Gamma_i$ are divided into those in the interior (I) and those on the interface (Γ). The interface set is further divided into a primal set (Π) and a dual set (Δ).

Partial assembly

- These equivalence classes play a central role in the design, analysis, and programming of domain decomposition methods.
- The BDDC and FETI–DP families are related algorithmically and have a common theoretical foundation. They are based on using partially subassembled stiffness matrices assembled from the subdomain stiffness matrices $A^{(i)}$. We will first look at a simple 2D nodal finite element problem.
- The nodes of $\Omega_i \cup \Gamma_i$ are divided into those in the interior (I) and those on the interface (Γ). The interface set is further divided into a primal set (Π) and a dual set (Δ).
- When developing theory, we can now handle quite irregular subdomains, in particular, in 2D. In 3D, we obtain bounds in terms of the Lipschitz parameters of the subdomains, often obtained by a mesh partitioner.

Torn 2D scalar elliptic problem



Subdomain matrices

- Represent the subdomain stiffness matrix $A^{(i)}$ as

$$\begin{pmatrix} A^{(i)}_{II} & A^{(i)}_{I\Delta} & A^{(i)}_{I\Pi} \\ A^{(i)}_{\Delta I} & A^{(i)}_{\Delta\Delta} & A^{(i)}_{\Delta\Pi} \\ A^{(i)}_{\Pi I} & A^{(i)}_{\Pi\Delta} & A^{(i)}_{\Pi\Pi} \end{pmatrix}.$$

Subdomain matrices

- Represent the subdomain stiffness matrix $A^{(i)}$ as

$$\begin{pmatrix} A^{(i)}_{II} & A^{(i)}_{I\Delta} & A^{(i)}_{I\Pi} \\ A^{(i)}_{\Delta I} & A^{(i)}_{\Delta\Delta} & A^{(i)}_{\Delta\Pi} \\ A^{(i)}_{\Pi I} & A^{(i)}_{\Pi\Delta} & A^{(i)}_{\Pi\Pi} \end{pmatrix}.$$

- This matrix represents the energy contributed by Ω_i.

Subdomain matrices

- Represent the subdomain stiffness matrix $A^{(i)}$ as

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} & A_{I\Pi}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} & A_{\Delta\Pi}^{(i)} \\ A_{\Pi I}^{(i)} & A_{\Pi\Delta}^{(i)} & A_{\Pi\Pi}^{(i)} \end{pmatrix}.$$

- This matrix represents the energy contributed by Ω_i.
- We enforce continuity of the primal variables, as in the given finite element model, but allow multiple values of the dual variables when working with the partially subassembled model.

Partially subassembled matrix

- Maintain continuity of the primal variables at the vertices. Partially subassemble and mark with tilde:

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Delta}^{(1)} & & & & A_{\Pi I}^{(1)T} \\ A_{\Delta I}^{(1)} & A_{\Delta\Delta}^{(1)} & & & & A_{\Pi\Delta}^{(1)T} \\ & & \ddots & & & \vdots \\ & & & A_{II}^{(N)} & A_{I\Delta}^{(N)} & A_{\Pi I}^{(N)T} \\ & & & A_{\Delta I}^{(N)} & A_{\Delta\Delta}^{(N)} & A_{\Pi\Delta}^{(N)T} \\ A_{\Pi I}^{(1)} & A_{\Pi\Delta}^{(1)} & \cdots & A_{\Pi I}^{(N)} & A_{\Pi\Delta}^{(N)} & \tilde{A}_{\Pi\Pi} \end{bmatrix}$$

Partially subassembled matrix

- Maintain continuity of the primal variables at the vertices. Partially subassemble and mark with tilde:

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Delta}^{(1)} & & & & A_{\Pi I}^{(1)T} \\ A_{\Delta I}^{(1)} & A_{\Delta\Delta}^{(1)} & & & & A_{\Pi\Delta}^{(1)T} \\ & & \ddots & & & \vdots \\ & & & A_{II}^{(N)} & A_{I\Delta}^{(N)} & A_{\Pi I}^{(N)T} \\ & & & A_{\Delta I}^{(N)} & A_{\Delta\Delta}^{(N)} & A_{\Pi\Delta}^{(N)T} \\ A_{\Pi I}^{(1)} & A_{\Pi\Delta}^{(1)} & \cdots & A_{\Pi I}^{(N)} & A_{\Pi\Delta}^{(N)} & \tilde{A}_{\Pi\Pi} \end{bmatrix}$$

- BDDC: After solving, enforce the continuity constraints by using an averaging operator E_D.

Partially subassembled matrix

- Maintain continuity of the primal variables at the vertices. Partially subassemble and mark with tilde:

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Delta}^{(1)} & & & & A_{\Pi I}^{(1)T} \\ A_{\Delta I}^{(1)} & A_{\Delta\Delta}^{(1)} & & & & A_{\Pi\Delta}^{(1)T} \\ & & \ddots & & & \vdots \\ & & & A_{II}^{(N)} & A_{I\Delta}^{(N)} & A_{\Pi I}^{(N)T} \\ & & & A_{\Delta I}^{(N)} & A_{\Delta\Delta}^{(N)} & A_{\Pi\Delta}^{(N)T} \\ A_{\Pi I}^{(1)} & A_{\Pi\Delta}^{(1)} & \cdots & A_{\Pi I}^{(N)} & A_{\Pi\Delta}^{(N)} & \tilde{A}_{\Pi\Pi} \end{bmatrix}$$

- BDDC: After solving, enforce the continuity constraints by using an averaging operator E_D.
- FETI–DP: Uses Lagrange multipliers. A saddle point problem is then reduced to an equation for the Lagrange multipliers.

More on BDDC and FETI–DP

- The partially subassembled stiffness matrix of this alternative finite element model is used to define preconditioners; the resulting linear system is much cheaper to solve than the fully assembled system. Primal variables provide a global component of these preconditioners. Makes matrices invertible.

More on BDDC and FETI–DP

- The partially subassembled stiffness matrix of this alternative finite element model is used to define preconditioners; the resulting linear system is much cheaper to solve than the fully assembled system. Primal variables provide a global component of these preconditioners. Makes matrices invertible.
- In a FETI–DP algorithm, the continuity at the edge nodes enforced by using Lagrange multipliers and the rate of convergence enhanced by also solving a Dirichlet problem on each subdomain in each iteration. The conjugate gradient algorithm is used to find accurate enough values of Lagrange multipliers. There are subtle scaling issues.

More on BDDC and FETI–DP

- The partially subassembled stiffness matrix of this alternative finite element model is used to define preconditioners; the resulting linear system is much cheaper to solve than the fully assembled system. Primal variables provide a global component of these preconditioners. Makes matrices invertible.
- In a FETI–DP algorithm, the continuity at the edge nodes enforced by using Lagrange multipliers and the rate of convergence enhanced by also solving a Dirichlet problem on each subdomain in each iteration. The conjugate gradient algorithm is used to find accurate enough values of Lagrange multipliers. There are subtle scaling issues.
- In a BDDC algorithm, continuity is instead restored in each iteration by computing a weighted average across the interface. Can lead to non-zero residuals at nodes next to Γ. If so, use a subdomain Dirichlet solve to eliminate them, in each iteration.

Alternative sets of primal constraints

- For scalar second order elliptic equations in 2D, this approach yields condition number of $C(1 + \log(H/h))^2$. Results can be made independent of jumps in the coefficients, if the interface averages chosen carefully.

Alternative sets of primal constraints

- For scalar second order elliptic equations in 2D, this approach yields condition number of $C(1+\log(H/h))^2$. Results can be made independent of jumps in the coefficients, if the interface averages chosen carefully.
- Good numerical results in 2D but for competitive algorithms in 3D, certain averages (and first order moments) of the displacement over individual edges (and faces) should also take common values across interface Γ. Same matrix structure as before after a change of variables.

Alternative sets of primal constraints

- For scalar second order elliptic equations in 2D, this approach yields condition number of $C(1 + \log(H/h))^2$. Results can be made independent of jumps in the coefficients, if the interface averages chosen carefully.
- Good numerical results in 2D but for competitive algorithms in 3D, certain averages (and first order moments) of the displacement over individual edges (and faces) should also take common values across interface Γ. Same matrix structure as before after a change of variables.
- Reliable recipes exist for selecting small sets of primal constraints for elasticity in 3D, which primarily use edge averages and first order moments as primal constraints. High quality PETSc-based codes have been developed by Stefano Zampini and been successfully tested on very large systems. Public domain software. Also great work by Klawonn's group.

Product spaces

- The BDDC and FETI–DP algorithms can be described in terms of three product spaces of finite element functions/vectors defined by their interface nodal values:

$$\widehat{W}_\Gamma \subset \widetilde{W}_\Gamma \subset W_\Gamma.$$

  W_Γ: no constraints; $\widehat{W}_\Gamma$: continuity at every point on Γ; $\widetilde{W}_\Gamma$: common values of the primal variables.

Product spaces

- The BDDC and FETI–DP algorithms can be described in terms of three product spaces of finite element functions/vectors defined by their interface nodal values:

$$\widehat{W}_\Gamma \subset \widetilde{W}_\Gamma \subset W_\Gamma.$$

  W_Γ: no constraints; $\widehat{W}_\Gamma$: continuity at every point on Γ; $\widetilde{W}_\Gamma$: common values of the primal variables.
- Can change variables, explicitly introducing primal variables and complementary sets of dual displacement variables. Simplifies presentation and might make methods more robust.

Product spaces

- The BDDC and FETI–DP algorithms can be described in terms of three product spaces of finite element functions/vectors defined by their interface nodal values:

$$\widehat{W}_\Gamma \subset \widetilde{W}_\Gamma \subset W_\Gamma .$$

  W_Γ: no constraints; $\widehat{W}_\Gamma$: continuity at every point on Γ; $\widetilde{W}_\Gamma$: common values of the primal variables.
- Can change variables, explicitly introducing primal variables and complementary sets of dual displacement variables. Simplifies presentation and might make methods more robust.
- After eliminating the interior variables, write the subdomain Schur complements as

$$S^{(i)} = \begin{pmatrix} S_{\Delta\Delta}^{(i)} & S_{\Delta\Pi}^{(i)} \\ S_{\Pi\Delta}^{(i)} & S_{\Pi\Pi}^{(i)} \end{pmatrix} .$$

Product spaces

- The BDDC and FETI–DP algorithms can be described in terms of three product spaces of finite element functions/vectors defined by their interface nodal values:

$$\widehat{W}_\Gamma \subset \widetilde{W}_\Gamma \subset W_\Gamma.$$

  W_Γ: no constraints; $\widehat{W}_\Gamma$: continuity at every point on Γ; $\widetilde{W}_\Gamma$: common values of the primal variables.
- Can change variables, explicitly introducing primal variables and complementary sets of dual displacement variables. Simplifies presentation and might make methods more robust.
- After eliminating the interior variables, write the subdomain Schur complements as

$$S^{(i)} = \begin{pmatrix} S^{(i)}_{\Delta\Delta} & S^{(i)}_{\Delta\Pi} \\ S^{(i)}_{\Pi\Delta} & S^{(i)}_{\Pi\Pi} \end{pmatrix}.$$

- Partially subassemble the $S^{(i)}$, obtaining $\tilde{S}$ and a global problem.

More details on BDDC

- Work with $\widetilde{W}_\Gamma$ and a set of primal constraints. At the end of each iterative step, the approximate solution will be made continuous at all nodal points of the interface; continuity is restored by applying a weighted average operator E_D, which maps $\widetilde{W}_\Gamma$ into $\widehat{W}_\Gamma$.

More details on BDDC

- Work with $\widetilde{W}_\Gamma$ and a set of primal constraints. At the end of each iterative step, the approximate solution will be made continuous at all nodal points of the interface; continuity is restored by applying a weighted average operator E_D, which maps $\widetilde{W}_\Gamma$ into $\widehat{W}_\Gamma$.
- In each iteration, first compute the residual of the fully assembled Schur complement. Then apply E_D^T to obtain right-hand side of the partially subassembled linear system. Solve this system and then apply E_D.

More details on BDDC

- Work with $\widetilde{W}_\Gamma$ and a set of primal constraints. At the end of each iterative step, the approximate solution will be made continuous at all nodal points of the interface; continuity is restored by applying a weighted average operator E_D, which maps $\widetilde{W}_\Gamma$ into $\widehat{W}_\Gamma$.
- In each iteration, first compute the residual of the fully assembled Schur complement. Then apply E_D^T to obtain right-hand side of the partially subassembled linear system. Solve this system and then apply E_D.
- This last step changes the values on Γ, unless the iteration has converged, and can results in non-zero residuals at nodes next to Γ.

More details on BDDC

- Work with $\widetilde{W}_\Gamma$ and a set of primal constraints. At the end of each iterative step, the approximate solution will be made continuous at all nodal points of the interface; continuity is restored by applying a weighted average operator E_D, which maps $\widetilde{W}_\Gamma$ into $\widehat{W}_\Gamma$.
- In each iteration, first compute the residual of the fully assembled Schur complement. Then apply E_D^T to obtain right-hand side of the partially subassembled linear system. Solve this system and then apply E_D.
- This last step changes the values on Γ, unless the iteration has converged, and can results in non-zero residuals at nodes next to Γ.
- In final step of each iteration, eliminate these residuals by solving a Dirichlet problem on each of the subdomains. Accelerate with preconditioned conjugate gradients.

E_D and some other operators

- For any application, only an estimate of $\|E_D\|_{\widetilde{S}}$ is needed.

E_D and some other operators

- For any application, only an estimate of $\|E_D\|_{\widetilde{S}}$ is needed.
- Let $\widetilde{R}_\Gamma$ be the operator that finally subassembles $\widetilde{S}_\Gamma$:

$$\widehat{S}_\Gamma = \widetilde{R}_\Gamma \widetilde{S}_\Gamma \widetilde{R}_\Gamma^T .$$

E_D and some other operators

- For any application, only an estimate of $\|E_D\|_{\widetilde{S}}$ is needed.
- Let $\widetilde{R}_\Gamma$ be the operator that finally subassembles $\widetilde{S}_\Gamma$:

$$\widehat{S}_\Gamma = \widetilde{R}_\Gamma \widetilde{S}_\Gamma \widetilde{R}_\Gamma^T .$$

- Obtain $\widetilde{R}_{D\Gamma}$ by scaling $\widetilde{R}_\Gamma$ by a block diagonal scaling.

E_D and some other operators

- For any application, only an estimate of $\|E_D\|_{\widetilde{S}}$ is needed.
- Let $\widetilde{R}_\Gamma$ be the operator that finally subassembles $\widetilde{S}_\Gamma$:

$$\widehat{S}_\Gamma = \widetilde{R}_\Gamma \widetilde{S}_\Gamma \widetilde{R}_\Gamma^T .$$

- Obtain $\widetilde{R}_{D\Gamma}$ by scaling $\widetilde{R}_\Gamma$ by a block diagonal scaling.
- $E_D := \widetilde{R}_\Gamma \widetilde{R}_{D\Gamma}^T$ and $M_{BDDC}^{-1} := \widetilde{R}_{D\Gamma}^T \widetilde{S}^{-1} \widetilde{R}_{D\Gamma}$.

E_D and some other operators

- For any application, only an estimate of $\|E_D\|_{\widetilde{S}}$ is needed.
- Let $\widetilde{R}_\Gamma$ be the operator that finally subassembles $\widetilde{S}_\Gamma$:

$$\widehat{S}_\Gamma = \widetilde{R}_\Gamma \widetilde{S}_\Gamma \widetilde{R}_\Gamma^T.$$

- Obtain $\widetilde{R}_{D\Gamma}$ by scaling $\widetilde{R}_\Gamma$ by a block diagonal scaling.
- $E_D := \widetilde{R}_\Gamma \widetilde{R}_{D\Gamma}^T$ and $M_{BDDC}^{-1} := \widetilde{R}_{D\Gamma}^T \widetilde{S}^{-1} \widetilde{R}_{D\Gamma}$.
- E_D is a projection and $\widetilde{R}_{D\Gamma}^T \widetilde{R}_\Gamma = I$.

Condition number bounds

Lemma

(Lower bound)

$$u^T M_{BDDC} u \leq u^T \widetilde{S}_\Gamma u, \quad \forall u \in \widetilde{W}_\Gamma. \tag{1}$$

Proof.

Let $w = M_{BDDC} u$. Since $\widetilde{R}_\Gamma^T \widetilde{R}_{D,\Gamma} = I$, we have

$$\begin{aligned}
u^T M_{BDDC} u \leq u^T w &= u^T \widetilde{R}_\Gamma^T \widetilde{R}_{D,\Gamma} w = u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w \\
&\leq (\widetilde{R}_\Gamma u, \widetilde{R}_\Gamma u)_{\widetilde{S}_\Gamma}^{1/2} (\widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w, \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w)_{\widetilde{S}_\Gamma}^{1/2} \\
&= (u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma u)^{1/2} (w^T \widetilde{R}_{D,\Gamma}^T \widetilde{S}_\Gamma^{-1} \widetilde{S}_\Gamma \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w)^{1/2} \\
&= (u^T \widetilde{S}_\Gamma u)^{1/2} (u^T M_{BDDC} u)^{1/2}.
\end{aligned}$$

Square and cancel common factor. ∎

Condition number bounds

Lemma

(Upper bound) If $|E_D v|^2_{\widetilde{S}_\Gamma} \le C_E |v|^2_{\widetilde{S}_\Gamma} \quad \forall v \in \widetilde{W}_\Gamma$, *then*

$$u^T \widetilde{S}_\Gamma u \le C_E u^T M_{BDDC} u \quad \forall u \in \widetilde{W}_\Gamma. \tag{2}$$

Proof.

$$u^T \widetilde{S}_\Gamma u = u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma u = u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma M_{BDDC}^{-1} M_{BDDC} u$$

$$
\begin{aligned}
&= u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma \widetilde{R}_{D,\Gamma}^T \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w \\
&\le (\widetilde{R}_\Gamma u, \widetilde{R}_\Gamma u)^{1/2}_{\widetilde{S}_\Gamma} (E_D \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w, E_D \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w)^{1/2}_{\widetilde{S}_\Gamma} \\
&\le (u^T \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma u)^{1/2} C_E^{1/2} (\widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w, \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} w)^{1/2}_{\widetilde{S}_\Gamma} \\
&= C_E^{1/2} (u^T \widetilde{S}_\Gamma u)^{1/2} (u^T M_{BDDC} u)^{1/2},
\end{aligned}
$$

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.
- Both the $H(\mathbf{curl})$ and $H(\mathrm{div})$ problems have two material parameters; complicates the design of the average operator.

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.
- Both the $H(\mathbf{curl})$ and $H(\text{div})$ problems have two material parameters; complicates the design of the average operator.
- Alternative found, also very robust for 3D $H(\text{div})$ problems: Duk-Soon Oh, OBW, Dohrmann, and Zampini; Math. Comp. 87(310), 2018.

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.
- Both the $H(\mathbf{curl})$ and $H(\mathrm{div})$ problems have two material parameters; complicates the design of the average operator.
- Alternative found, also very robust for 3D $H(\mathrm{div})$ problems: Duk-Soon Oh, OBW, Dohrmann, and Zampini; Math. Comp. 87(310), 2018.
- Papers on isogeometric elements, with Beirão da Veiga, Pavarino, Scacchi, and Zampini: M^3AS 23(6), SISC 36(3) and 39(1), M^3AS 28(7). The deluxe variant is superior.

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.
- Both the $H(\mathbf{curl})$ and $H(\mathrm{div})$ problems have two material parameters; complicates the design of the average operator.
- Alternative found, also very robust for 3D $H(\mathrm{div})$ problems: Duk-Soon Oh, OBW, Dohrmann, and Zampini; Math. Comp. 87(310), 2018.
- Papers on isogeometric elements, with Beirão da Veiga, Pavarino, Scacchi, and Zampini: M^3AS 23(6), SISC 36(3) and 39(1), M^3AS 28(7). The deluxe variant is superior.
- My former student Jong Ho Lee has published a paper on Reissner-Mindlin plates: SINUM 53(1).

BDDC deluxe

- When designing a BDDC algorithm, we have to choose an effective set of primal constraints and also a recipe for the averaging across interface.
- Traditional averaging recipes found not to work uniformly well for 3D problems in $H(\mathbf{curl})$: DD20 paper, CPAM 69 with CRD.
- Both the $H(\mathbf{curl})$ and $H(\mathrm{div})$ problems have two material parameters; complicates the design of the average operator.
- Alternative found, also very robust for 3D $H(\mathrm{div})$ problems: Duk-Soon Oh, OBW, Dohrmann, and Zampini; Math. Comp. 87(310), 2018.
- Papers on isogeometric elements, with Beirão da Veiga, Pavarino, Scacchi, and Zampini: M^3AS 23(6), SISC 36(3) and 39(1), M^3AS 28(7). The deluxe variant is superior.
- My former student Jong Ho Lee has published a paper on Reissner-Mindlin plates: SINUM 53(1).
- Also work on DG by Dryja and Sarkis and by Chung and Kim.

Stiffness scaling in frequency space

- New average operator E_D across a face $F \subset \Gamma$, common to two subdomains Ω_i and Ω_j, defined in terms of two Schur complements:

$$S_{FF}^{(k)} := A_{FF}^{(k)} - A_{FI}^{(k)} {A_{II}^{(k)}}^{-1} A_{IF}^{(k)}, \;\; k = i, j.$$

Stiffness scaling in frequency space

- New average operator E_D across a face $F \subset \Gamma$, common to two subdomains Ω_i and Ω_j, defined in terms of two Schur complements:

$$S_{FF}^{(k)} := A_{FF}^{(k)} - A_{FI}^{(k)} {A_{II}^{(k)}}^{-1} A_{IF}^{(k)}, \;\; k = i, j.$$

- The deluxe averaging operator is then defined by

$$\bar{w}_F := (E_D w)_F := (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} (S_{FF}^{(i)} w^{(i)} + S_{FF}^{(j)} w^{(j)}).$$

Stiffness scaling in frequency space

- New average operator E_D across a face $F \subset \Gamma$, common to two subdomains Ω_i and Ω_j, defined in terms of two Schur complements:

$$S_{FF}^{(k)} := A_{FF}^{(k)} - A_{FI}^{(k)} A_{II}^{(k)^{-1}} A_{IF}^{(k)}, \;\; k = i, j.$$

- The deluxe averaging operator is then defined by

$$\bar{w}_F := (E_D w)_F := (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} (S_{FF}^{(i)} w^{(i)} + S_{FF}^{(j)} w^{(j)}).$$

- This action can be implemented by solving a Dirichlet problem on $\Omega_i \cup \Gamma_{ij} \cup \Omega_j$; Γ_{ij} interface between two subdomains. Adds to the costs. But MUMS provides the Schur complements.

Stiffness scaling in frequency space

- New average operator E_D across a face $F \subset \Gamma$, common to two subdomains Ω_i and Ω_j, defined in terms of two Schur complements:

$$S_{FF}^{(k)} := A_{FF}^{(k)} - A_{FI}^{(k)} {A_{II}^{(k)}}^{-1} A_{IF}^{(k)}, \;\; k = i, j.$$

- The deluxe averaging operator is then defined by

$$\bar{w}_F := (E_D w)_F := (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} (S_{FF}^{(i)} w^{(i)} + S_{FF}^{(j)} w^{(j)}).$$

- This action can be implemented by solving a Dirichlet problem on $\Omega_i \cup \Gamma_{ij} \cup \Omega_j$; Γ_{ij} interface between two subdomains. Adds to the costs. But MUMS provides the Schur complements.
- Similar formulas for subdomain edges and other equivalence classes of interface variables. The operator E_D is assembled from these components.

BDDC deluxe

- As we have shown, the core of any estimate for a BDDC algorithm is the norm of the average operator E_D. Known for FETI–DP since 2002,

$$\kappa(M^{-1}A) \leq \|E_D\|_{\widetilde{S}}.$$

BDDC deluxe

- As we have shown, the core of any estimate for a BDDC algorithm is the norm of the average operator E_D. Known for FETI–DP since 2002,

$$\kappa(M^{-1}A) \leq \|E_D\|_{\widetilde{S}}.$$

- We will show that essentially the analysis can be reduced to bounds for individual subdomains.

BDDC deluxe

- As we have shown, the core of any estimate for a BDDC algorithm is the norm of the average operator E_D. Known for FETI–DP since 2002,

$$\kappa(M^{-1}A) \leq \|E_D\|_{\widetilde{S}}.$$

- We will show that essentially the analysis can be reduced to bounds for individual subdomains.
- Arbitrary jumps in two coefficients can often be accommodated.

BDDC deluxe

- As we have shown, the core of any estimate for a BDDC algorithm is the norm of the average operator E_D. Known for FETI–DP since 2002,

$$\kappa(M^{-1}A) \leq \|E_D\|_{\widetilde{S}}.$$

- We will show that essentially the analysis can be reduced to bounds for individual subdomains.
- Arbitrary jumps in two coefficients can often be accommodated.
- Analysis of traditional BDDC requires an extension theorem; deluxe version does not.

BDDC deluxe

- As we have shown, the core of any estimate for a BDDC algorithm is the norm of the average operator E_D. Known for FETI–DP since 2002,

$$\kappa(M^{-1}A) \leq \|E_D\|_{\widetilde{S}}.$$

- We will show that essentially the analysis can be reduced to bounds for individual subdomains.
- Arbitrary jumps in two coefficients can often be accommodated.
- Analysis of traditional BDDC requires an extension theorem; deluxe version does not.
- In addition to this good choice of averaging, a good primal space has to be chosen.

Analysis of BDDC deluxe

- Instead of estimating $(R_F^T \bar{w}_F)^T S^{(i)} R_F^T \bar{w}_F$, estimate the norm of $R_F^T(w^{(i)} - \bar{w}_F)$. By simple algebra, we find that

$$w_F^{(i)} - \bar{w}_F = (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} S_{FF}^{(j)} (w_F^{(i)} - w_F^{(j)}).$$

- Instead of estimating $(R_F^T \bar{w}_F)^T S^{(i)} R_F^T \bar{w}_F$, estimate the norm of $R_F^T(w^{(i)} - \bar{w}_F)$. By simple algebra, we find that

$$w_F^{(i)} - \bar{w}_F = (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} S_{FF}^{(j)} (w_F^{(i)} - w_F^{(j)}).$$

- By more algebra, noting that $R_F S^{(i)} R_F^T = S_{FF}^{(i)}$:

$$(R_F^T (w_F^{(i)} - \bar{w}_F))^T S^{(i)} (R_F^T (w_F^{(i)} - \bar{w}_F)) =$$

$$(w_F^{(i)} - w_F^{(j)})^T S_{FF}^{(j)} (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} S_{FF}^{(i)} (S_{FF}^{(i)} + S_{FF}^{(j)})^{-1} S_{FF}^{(j)} (w_F^{(i)} - w_F^{(ji)})).$$

  Add similar expression for the subdomain Ω_j. More algebra gives the bound:

$$(w_F^{(i)} - w_F^{(j)})^T S_{FF}^{(i)} : S_{FF}^{(j)} (w_F^{(i)} - w_F^{(ji)}))$$

  where

$$A : B := (A^{-1} + B^{-1})^{-1}$$

  is known as a parallel sum.

Proof continued

Lemma

Let A and B be two symmetric positive semi-definite matrices of the same order. Then, $z^T A : Bz = \inf_{z=x+y}(x^T Ax + y^T By)$.

Proof continued

Lemma

Let A and B be two symmetric positive semi-definite matrices of the same order. Then, $z^T A : Bz = \inf_{z=x+y}(x^T Ax + y^T By)$.

- Now use $x := (w^{(i)} - w_\Pi)_F$ and $y := -(w^{(j)} - w_\Pi)_F$ to obtain bounds for single subdomains.

Proof continued

Lemma

Let A and B be two symmetric positive semi-definite matrices of the same order. Then, $z^T A : Bz = \inf_{z=x+y}(x^T Ax + y^T By)$.

- Now use $x := (w^{(i)} - w_\Pi)_F$ and $y := -(w^{(j)} - w_\Pi)_F$ to obtain bounds for single subdomains.
- Remains to estimate $(w^{(i)} - w_\Pi)_F^T S_{FF}^{(i)} (w^{(i)} - w_\Pi)_F$ by ${w^{(i)}}^T S^{(i)} w^{(i)}$ choosing a good shift with an element of the primal space.

Proof continued

Lemma

Let A and B be two symmetric positive semi-definite matrices of the same order. Then, $z^T A : Bz = \inf_{z=x+y}(x^T Ax + y^T By)$.

- Now use $x := (w^{(i)} - w_\Pi)_F$ and $y := -(w^{(j)} - w_\Pi)_F$ to obtain bounds for single subdomains.
- Remains to estimate $(w^{(i)} - w_\Pi)_F^T S_{FF}^{(i)} (w^{(i)} - w_\Pi)_F$ by ${w^{(i)}}^T S^{(i)} w^{(i)}$ choosing a good shift with an element of the primal space.
- Routine for H^1: completely standard estimates in the domain decomposition literature: Face lemma; compare energy of zero extension of face values by zero with that of the minimal energy extension. Factor of $C(1 + \log(H/h))^2$ results.

Proof continued

Lemma

Let A and B be two symmetric positive semi-definite matrices of the same order. Then, $z^T A : Bz = \inf_{z=x+y}(x^T Ax + y^T By)$.

- Now use $x := (w^{(i)} - w_\Pi)_F$ and $y := -(w^{(j)} - w_\Pi)_F$ to obtain bounds for single subdomains.
- Remains to estimate $(w^{(i)} - w_\Pi)_F^T S_{FF}^{(i)}(w^{(i)} - w_\Pi)_F$ by ${w^{(i)}}^T S^{(i)} w^{(i)}$ choosing a good shift with an element of the primal space.
- Routine for H^1: completely standard estimates in the domain decomposition literature: Face lemma; compare energy of zero extension of face values by zero with that of the minimal energy extension. Factor of $C(1 + \log(H/h))^2$ results.
- Also estimate contributions from subdomain edges, etc. For many edges, more than two Schur complements enters in in the averages. Bound but with small integer factors.

Adaptive choice of primal spaces

- The subdomain bounds can be expressed in terms of Schur complements of Schur complements

$$\breve{S}_{FF}^{(i)} := S_{FF}^{(i)} - S_{F'F}^{(i)T} S_{F'F'}^{(i)-1} S_{F'F}^{(i)}$$

where $F' := \Gamma_i \setminus F$.

Adaptive choice of primal spaces

- The subdomain bounds can be expressed in terms of Schur complements of Schur complements

$$\breve{S}_{FF}^{(i)} := S_{FF}^{(i)} - S_{F'F}^{(i)T} S_{F'F'}^{(i)-1} S_{F'F}^{(i)}$$

where $F' := \Gamma_i \setminus F$.

- The relevant bound is given by the generalized eigenvalue problem

$$S_{FF}^{(i)} : S_{FF}^{(j)} \psi = \nu \breve{S}_{FF}^{(i)} : \breve{S}_{FF}^{(j)} \psi.$$

Adaptive choice of primal spaces

- The subdomain bounds can be expressed in terms of Schur complements of Schur complements

$$\breve{S}_{FF}^{(i)} := S_{FF}^{(i)} - S_{F'F}^{(i)T} S_{F'F'}^{(i)-1} S_{F'F}^{(i)}$$

where $F' := \Gamma_i \setminus F$.

- The relevant bound is given by the generalized eigenvalue problem

$$S_{FF}^{(i)} : S_{FF}^{(j)} \psi = \nu \breve{S}_{FF}^{(i)} : \breve{S}_{FF}^{(j)} \psi.$$

- We can now improve the bound by incorporating the eigenvectors of the largest eigenvalues into the primal space.

Adaptive choice of primal spaces

- The subdomain bounds can be expressed in terms of Schur complements of Schur complements

$$\breve{S}_{FF}^{(i)} := S_{FF}^{(i)} - S_{F'F}^{(i)T} S_{F'F'}^{(i)-1} S_{F'F}^{(i)}$$

where $F' := \Gamma_i \setminus F$.
- The relevant bound is given by the generalized eigenvalue problem

$$S_{FF}^{(i)} : S_{FF}^{(j)} \psi = \nu \breve{S}_{FF}^{(i)} : \breve{S}_{FF}^{(j)} \psi.$$

- We can now improve the bound by incorporating the eigenvectors of the largest eigenvalues into the primal space.
- Similar devices, so far not equally appealing, have been developed for equivalence classes with more than two elements. Pechstein and Dohrmann ETNA 46 pp. 273-336.

Adaptive choice of primal spaces

- The subdomain bounds can be expressed in terms of Schur complements of Schur complements

$$\breve{S}_{FF}^{(i)} := S_{FF}^{(i)} - S_{F'F}^{(i)T} S_{F'F'}^{(i)-1} S_{F'F}^{(i)}$$

where $F' := \Gamma_i \setminus F$.

- The relevant bound is given by the generalized eigenvalue problem

$$S_{FF}^{(i)} : S_{FF}^{(j)} \psi = \nu \breve{S}_{FF}^{(i)} : \breve{S}_{FF}^{(j)} \psi.$$

- We can now improve the bound by incorporating the eigenvectors of the largest eigenvalues into the primal space.
- Similar devices, so far not equally appealing, have been developed for equivalence classes with more than two elements. Pechstein and Dohrmann ETNA 46 pp. 273-336.
- For $H(\text{div})$ there are no such classes; Oh et al, Math. Comp. 87(310). Works well for nasty coefficients.

How to deal with primal spaces that are too large?

- The primal space, which generates a global problem, can create a bottle-neck when solving very large problems on computer systems with very many processors.

How to deal with primal spaces that are too large?

- The primal space, which generates a global problem, can create a bottle-neck when solving very large problems on computer systems with very many processors.
- This problem has been and remains a focus of many studies of BDDC and FETI–DP.

How to deal with primal spaces that are too large?

- The primal space, which generates a global problem, can create a bottle-neck when solving very large problems on computer systems with very many processors.
- This problem has been and remains a focus of many studies of BDDC and FETI–DP.
- One way of dealing with this is to introduce a third or even more levels. Very interesting proof that it works well is due to Xuemin Tu. PETSc code by Stefano Zampini has been quite successful and has been combined with adaptive choices of primal spaces, e.g., for $H(\text{div})$ and $H(\mathbf{curl})$ problems.

How to deal with primal spaces that are too large?

- The primal space, which generates a global problem, can create a bottle-neck when solving very large problems on computer systems with very many processors.
- This problem has been and remains a focus of many studies of BDDC and FETI–DP.
- One way of dealing with this is to introduce a third or even more levels. Very interesting proof that it works well is due to Xuemin Tu. PETSc code by Stefano Zampini has been quite successful and has been combined with adaptive choices of primal spaces, e.g., for $H(\text{div})$ and $H(\mathbf{curl})$ problems.
- Choosing a standard primal space based on vertex variables only leads to poor performance for 3D. However, in a paper, in progress by Dohrmann, Pierson, and OBW, it will be shown that in many cases, the primal problem can be solved approximately and very effectively using a coarse space of that same small dimension.

Outline of recent work

- A two-level BDDC preconditioner in additive form:

$$M^{-1} = M_{local}^{-1} + \Phi_D K_c^{-1} \Phi_D^T,$$

where K_c is the coarse matrix and Φ_D is a weighted interpolation matrix.

- A two-level BDDC preconditioner in additive form:

$$M^{-1} = M_{local}^{-1} + \Phi_D K_c^{-1} \Phi_D^T,$$

where K_c is the coarse matrix and Φ_D is a weighted interpolation matrix.

- Let M_c^{-1} be a preconditioner for K_c and for $0 < \beta_1 \leq \beta_2$

$$\beta_1 u_c^T K_c^{-1} u_c \leq u_c^T M_c^{-1} u_c \leq \beta_2 u_c^T K_c^{-1} u_c \quad \forall u_c.$$

We easily find that $\kappa_a \leq \frac{\max(1,\beta_2)}{\min(1,\beta_1)} \kappa$.

Outline of recent work

- A two-level BDDC preconditioner in additive form:

$$M^{-1} = M_{local}^{-1} + \Phi_D K_c^{-1} \Phi_D^T,$$

where K_c is the coarse matrix and Φ_D is a weighted interpolation matrix.

- Let M_c^{-1} be a preconditioner for K_c and for $0 < \beta_1 \leq \beta_2$

$$\beta_1 u_c^T K_c^{-1} u_c \leq u_c^T M_c^{-1} u_c \leq \beta_2 u_c^T K_c^{-1} u_c \quad \forall u_c.$$

We easily find that $\kappa_a \leq \frac{\max(1,\beta_2)}{\min(1,\beta_1)} \kappa$.

- Choose $M_c^{-1} = \Psi K_{cr}^{-1} \Psi^T + \text{diag}(K_c)^{-1}$ where $K_{cr} := \Psi^T K_c \Psi$. (Point Jacobi + coarse correction.)

Outline of recent work

- A two-level BDDC preconditioner in additive form:

$$M^{-1} = M_{local}^{-1} + \Phi_D K_c^{-1} \Phi_D^T,$$

  where K_c is the coarse matrix and Φ_D is a weighted interpolation matrix.

- Let M_c^{-1} be a preconditioner for K_c and for $0 < \beta_1 \leq \beta_2$

$$\beta_1 u_c^T K_c^{-1} u_c \leq u_c^T M_c^{-1} u_c \leq \beta_2 u_c^T K_c^{-1} u_c \quad \forall u_c.$$

  We easily find that $\kappa_a \leq \frac{\max(1,\beta_2)}{\min(1,\beta_1)} \kappa$.

- Choose $M_c^{-1} = \Psi K_{cr}^{-1} \Psi^T + \text{diag}(K_c)^{-1}$ where $K_{cr} := \Psi^T K_c \Psi$. (Point Jacobi + coarse correction.)
- The hard part is to estimate β_1; for an important case, the result by Tu and ultimately Brenner again comes into play.

Almost incompressible elasticity

- How to develop good domain decomposition algorithms for mixed methods for almost incompressible elasticity and incompressible Stokes problems when the pressure variable p is continuous? Open problem for quite some time.

Almost incompressible elasticity

- How to develop good domain decomposition algorithms for mixed methods for almost incompressible elasticity and incompressible Stokes problems when the pressure variable p is continuous? Open problem for quite some time.
- If the pressure is discontinuous, the elasticity problem can be reduced to a positive definite problem by eliminating the pressure on the element level.

Almost incompressible elasticity

- How to develop good domain decomposition algorithms for mixed methods for almost incompressible elasticity and incompressible Stokes problems when the pressure variable p is continuous? Open problem for quite some time.
- If the pressure is discontinuous, the elasticity problem can be reduced to a positive definite problem by eliminating the pressure on the element level.
- Discrete saddle point system:

$$\begin{bmatrix} \mu A & B^T \\ B & -\frac{1}{\lambda} C \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} f_h \\ 0 \end{bmatrix},$$

where A, B, and C are the matrices associated with the bilinear forms of the mixed formulation of almost incompressible elasticity. μ and λ the Lamé parameters.

Block FETI-DP preconditioner a la Tu and Li, 2015

- Split displacement into interior, dual, and primal components:

$$\mathbf{u}_h = (\mathbf{u}_I, \mathbf{u}_\Delta, \mathbf{u}_\Pi),$$

and pressure into interiors and, interface components:

$$p_h = (p_I, p_\Gamma),$$

and introduce $\lambda_\Delta =$ Lagrange multipliers used to enforce the continuity of the dual displacements in the limit.

Block FETI-DP preconditioner a la Tu and Li, 2015

- Split displacement into interior, dual, and primal components:

$$\mathbf{u}_h = (\mathbf{u}_I, \mathbf{u}_\Delta, \mathbf{u}_\Pi),$$

  and pressure into interiors and, interface components:

$$p_h = (p_I, p_\Gamma),$$

  and introduce λ_Δ = Lagrange multipliers used to enforce the continuity of the dual displacements in the limit.

- Permute the saddle point system to the form:

$$\begin{bmatrix} \mu A_{II} & B_{II}^T & \mu A_{I\Delta} & \mu A_{I\Pi} & B_{\Gamma I}^T & 0 \\ B_{II} & -\frac{1}{\lambda} C_{II} & B_{\Delta I}^T & B_{\Pi I}^T & -\frac{1}{\lambda} C_{I\Gamma} & 0 \\ \mu A_{\Delta I} & B_{\Delta I} & \mu A_{\Delta\Delta} & \mu A_{\Delta\Pi} & B_{\Gamma\Delta}^T & B_\Delta^T \\ \mu A_{\Pi I} & B_{\Pi I} & \mu A_{\Pi\Delta} & \mu A_{\Pi\Pi} & B_{\Gamma\Pi}^T & 0 \\ B_{\Gamma I} & -\frac{1}{\lambda} C_{\Gamma I} & B_{\Gamma\Delta} & B_{\Gamma\Pi} & -\frac{1}{\lambda} C_{\Gamma\Gamma} & 0 \\ 0 & 0 & B_\Delta & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Delta \\ \mathbf{u}_\Pi \\ p_\Gamma \\ \lambda_\Delta \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \\ f_\Delta \\ f_\Pi \\ 0 \\ 0 \end{bmatrix}$$

Block FETI-DP preconditioners

- After a block Gaussian elimination step, a Schur complement $-G$, which is negative definite and a reduced system.

FETI-DP type system

$$G \begin{bmatrix} p_\Gamma \\ \lambda_\Delta \end{bmatrix} = g. \tag{3}$$

IGA block FETI-DP preconditioner for the reduced system (3)

$$M_D^{-1} = \begin{bmatrix} M_{p_\Gamma}^{-1} & 0 \\ 0 & M_{\lambda_\Delta}^{-1} \end{bmatrix},$$

Block FETI-DP preconditioners

- After a block Gaussian elimination step, a Schur complement $-G$, which is negative definite and a reduced system.

FETI-DP type system

$$G \left[\begin{array}{c} p_\Gamma \\ \lambda_\Delta \end{array} \right] = g. \quad (3)$$

IGA block FETI-DP preconditioner for the reduced system (3)

$$M_D^{-1} = \left[\begin{array}{cc} M_{p_\Gamma}^{-1} & 0 \\ 0 & M_{\lambda_\Delta}^{-1} \end{array} \right],$$

$M_{p_\Gamma}^{-1} = \mu$ times the inverse of Schur complement of C or use BDDC. $M_{\lambda_\Delta}^{-1}$ essentially the same as for compressible elasticity.