IsogEometric Tearing and Interconnecting

Christoph Hofer and Ludwig Mitter

Johannes Kepler University, Linz

26.01.2017

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_\Pi$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

Motivation

- In 2D, using vertex primal variables works quite well.
- In 3D, condition number grows with $H/h(1+\log H/h)^2$.

2D				3D			
#dofs	H/h	κ	It.	#dofs	H/h	κ	It.
3350	11	11.4	23	3100	3	29.9	28
9614	19	14.5	25	7228	6	75.8	38
31742	35	18.1	27	23680	12	161	45
114398	67	22.2	28	106168	25	370	64
433310	131	26.6	30				

- Using continuous edge/face averages gives $(1+\log(H/h))^2$.
- Implementation gets a bit more tricky.
- Present method for arbitrary linear primal variables.
- Pechstein, C. (2012). *Finite and boundary element tearing and interconnecting solvers for multiscale problems* (Vol. 90). Springer Science & Business Media.

Motivation

- In 2D, using vertex primal variables works quite well.
- In 3D, condition number grows with $H/h(1 + \log H/h)^2$.

2D				3D			
#dofs	H/h	κ	It.	#dofs	H/h	κ	It.
3350	11	11.4	23	3100	3	29.9	28
9614	19	14.5	25	7228	6	75.8	38
31742	35	18.1	27	23680	12	161	45
114398	67	22.2	28	106168	25	370	64
433310	131	26.6	30				

- Using continuous edge/face averages gives $(1 + \log(H/h))^2$.
- Implementation gets a bit more tricky.
- Present method for arbitrary linear primal variables.
- Pechstein, C. (2012). *Finite and boundary element tearing and interconnecting solvers for multiscale problems* (Vol. 90). Springer Science & Business Media.

Problem formulation - cG setting

Find $u_h \in V_{D,h}$:

$$a(u_h, v_h) = \langle F, v_h \rangle \quad \forall v_h \in V_{D,h},$$

where $V_{D,h}$ is a conforming discrete subspaces of V_D, e.g.

$$a(u,v) = \int_\Omega \alpha \nabla u \nabla v \, dx, \quad \langle F, v \rangle = \int_\Omega f v \, dx + \int_{\Gamma_N} g_N v \, ds$$

$$V_D = \{u \in H^1 : \gamma_0 u = 0 \text{ on } \Gamma_D\},$$

$$V_{D,h} = \prod_k \text{span}\{N_{i,p}^{(k)}\} \cap H^1(\Omega).$$

The variational equation is equivalent to $Ku = f$.

Overview

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

IETI-DP

Given $K^{(k)}$ and $f^{(k)}$, we can reformulate

$$Ku = f \quad \leftrightarrow \quad \begin{bmatrix} K_e & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} f_e \\ 0 \end{bmatrix},$$

where $K_e = \text{diag}(K^{(k)})$ and $f_e = [f^{(k)}]$.

Since K_e is not invertible, we need additional primal variables incorporated in $K_e \rightsquigarrow \widetilde{K}, \widetilde{B}, \widetilde{f}$:

- continuous vertex values
- continuous edge/face averages

IETI-DP

Given $K^{(k)}$ and $f^{(k)}$, we can reformulate

$$Ku = f \quad \leftrightarrow \quad \begin{bmatrix} K_e & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} f_e \\ 0 \end{bmatrix},$$

where $K_e = \mathrm{diag}(K^{(k)})$ and $f_e = [f^{(k)}]$.

Since K_e is not invertible, we need additional primal variables incorporated in $K_e \rightsquigarrow \widetilde{K}, \widetilde{B}, \widetilde{f}$:

- continuous vertex values
- continuous edge/face averages

IETI-DP

Find $(u, \boldsymbol{\lambda})$

$$\begin{bmatrix} \widetilde{K} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widetilde{f}_e \\ 0 \end{bmatrix}.$$

$\widetilde{K}$ is SPD, hence, we can define:

$$F := \widetilde{B}\widetilde{K}^{-1}\widetilde{B}^T \quad d := \widetilde{B}\widetilde{K}^{-1}\widetilde{f}$$

The saddle point system is equivalent to solving:

$$\text{find } \boldsymbol{\lambda} \in U : \quad F\boldsymbol{\lambda} = d.$$

Using the preconditioner M_{sD}^{-1}, we obtain:

$$\kappa(M_{sD}^{-1} F_{|\mathsf{ker}(\widetilde{B}^T)}) \leq C \max_{1\leq k\leq N} \left(1 + \log\left(\frac{H_k}{h_k}\right)\right)^2,$$

A bit more on primal variables

$$W^{(k)} := V_h^{(k)}, \quad W := \prod_k W^{(k)}, \quad \widehat{W} := V_h.$$

Intermediate space $\widetilde{W}$: $\widehat{W} \subset \widetilde{W} \subset W$, $\widetilde{K} := K_{|\widetilde{W}}$ is SPD.

Let $\Psi \subset V_h^*$ be a set of linearly independent *primal variables*,

$$\widetilde{W} := \{w \in W : \forall \psi \in \Psi : \psi(w_i) = \psi(w_j)\}$$

$$W_\Delta := \prod_{k=0}^{n} W_\Delta^{(k)} \quad \text{where } W_\Delta^{(k)} := \{w \in W^{(k)} : \forall \psi \in \Psi : \psi(w_k) = 0\}$$

$$\widetilde{W} = W_\Pi \oplus W_\Delta, \quad W_\Pi \subset \widehat{W} \quad \textit{(there are many choices for } W_\Pi\textit{)}$$

If $\widetilde{W} \cap \ker(K) = \{0\}$, then $\widetilde{K}$ is invertible.

A bit more on primal variables

$$W^{(k)} := V_h^{(k)}, \quad W := \prod_k W^{(k)}, \quad \widehat{W} := V_h.$$

Intermediate space $\widetilde{W}$: $\widehat{W} \subset \widetilde{W} \subset W$, $\widetilde{K} := K_{|\widetilde{W}}$ is SPD.
Let $\Psi \subset V_h^*$ be a set of linearly independent *primal variables*,

$$\widetilde{W} := \{w \in W : \forall \psi \in \Psi : \psi(w_i) = \psi(w_j)\}$$

$$W_\Delta := \prod_{k=0}^{n} W_\Delta^{(k)} \quad \text{where } W_\Delta^{(k)} := \{w \in W^{(k)} : \forall \psi \in \Psi : \psi(w_k) = 0\}$$

$$\widetilde{W} = W_\Pi \oplus W_\Delta, \quad W_\Pi \subset \widehat{W} \quad \textit{(there are many choices for } W_\Pi\textit{)}$$

If $\widetilde{W} \cap \ker(K) = \{0\}$, then $\widetilde{K}$ is invertible.

Typical examples of Ψ and ψ

Choices for ψ:

- Vertex evaluation: $\psi^{\mathcal{V}}(v) = v(\mathcal{V})$
- Edge averages: $\psi^{\mathcal{E}}(v) = \frac{1}{|\mathcal{E}|}\int_{\mathcal{E}} v\,ds$
- Face averages: $\psi^{\mathcal{F}}(v) = \frac{1}{|\mathcal{F}|}\int_{\mathcal{F}} v\,ds$

Choices for Ψ:

- Algorithm A: $\Psi = \{\psi^{\mathcal{V}}\}$
- Algorithm B: $\Psi = \{\psi^{\mathcal{V}}\} \cup \{\psi^{\mathcal{E}}\} \cup \{\psi^{\mathcal{F}}\}$
- Algorithm C: $\Psi = \{\psi^{\mathcal{V}}\} \cup \{\psi^{\mathcal{E}}\}$

Since $\widetilde{W} \subset W$, there is a natural embedding $\widetilde{I} : \widetilde{W} \to W$.
We can define:

- $\widetilde{B} := B\widetilde{I} : \quad \widetilde{W} \to U^*$,
- $\widetilde{B}^T = \widetilde{I}^T B^T : \quad U \to \widetilde{W}^*$,
- $\widetilde{f} := \widetilde{I}^T f \quad \in \widetilde{W}^*$

As before, we can write our saddle point problem as:
Find $(u, \boldsymbol{\lambda}) \in \widetilde{W} \times U$:

$$\begin{bmatrix} \widetilde{K} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widetilde{f} \\ 0 \end{bmatrix},$$

Since $\widetilde{W} \subset W$, there is a natural embedding $\widetilde{I} : \widetilde{W} \to W$.
We can define:

- $\widetilde{B} := B\widetilde{I} : \quad \widetilde{W} \to U^*,$
- $\widetilde{B}^T = \widetilde{I}^T B^T : \quad U \to \widetilde{W}^*,$
- $\widetilde{f} := \widetilde{I}^T f \quad \in \widetilde{W}^*$

As before, we can write our saddle point problem as:
Find $(u, \boldsymbol{\lambda}) \in \widetilde{W} \times U$:

$$\begin{bmatrix} \widetilde{K} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} u \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widetilde{f} \\ 0 \end{bmatrix},$$

Overview

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_\Pi$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

CG Algorithm

The equation $F\boldsymbol{\lambda} = d$, is solved via the PCG algorithm:

$\boldsymbol{\lambda}_0$ given

$r_0 = d - F\boldsymbol{\lambda}_0, \quad k = 0, \quad \beta_{-1} = 0$

repeat

$$s_k = M_{sD}^{-1} r_k$$
$$\beta_{k-1} = \frac{(r_k, s_k)}{(r_{k-1}, s_{k-1})}$$
$$p_k = s_k + \beta_{k-1} p_{k-1}$$
$$\alpha_k = \frac{(r_k, s_k)}{(F p_k, p_k)}$$
$$\boldsymbol{\lambda}_{k+1} = r_k + \alpha_k p_k$$
$$r_{k+1} = r_k - \alpha_k F p_k$$
$$k = k + 1$$

until stopping criterion fulfilled

Required realizations

In order to use the CG-algorithm, we need

- Application of $F := \widetilde{B}\widetilde{K}^{-1}\widetilde{B}^T$
- Application of $M_{sD}^{-1} := B_D S_e B_D^T$

Representation of $\widetilde{W}$:

- $\widetilde{K} : \widetilde{W} \to \widetilde{W}^*, \quad \widetilde{K}^{-1} : \widetilde{W}^* \to \widetilde{W}$
- $\widetilde{W} = W_\Pi \oplus \prod W_\Delta^{(k)}$
- representation of $w \in \widetilde{W}$ as $\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_k\}$
- representation of $f \in \widetilde{W}^*$ as $\{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}_k\}$

Required realizations

In order to use the CG-algorithm, we need

- Application of $F := \widetilde{B}\widetilde{K}^{-1}\widetilde{B}^T$
- Application of $M_{sD}^{-1} := B_D S_e B_D^T$

Representation of $\widetilde{W}$:

- $\widetilde{K} : \widetilde{W} \to \widetilde{W}^*, \quad \widetilde{K}^{-1} : \widetilde{W}^* \to \widetilde{W}$
- $\widetilde{W} = W_\Pi \oplus \prod W_\Delta^{(k)}$
- representation of $w \in \widetilde{W}$ as $\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_k\}$
- representation of $f \in \widetilde{W}^*$ as $\{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}_k\}$

Required realizations

$$\widetilde{W} = W_\Pi \oplus \prod W_\Delta^{(k)}$$

- Construction of the primal space W_Π and its basis.
- We choose the so called *energy minimizing primal subspaces*.
- The basis should be at least *local* and *nodal*.
- 2 possibilities to realize the dual space W_Δ:
  - Transformation of basis: construction of basis, such that the primal variables vanishes.
  - Realization with local constraints: constraints are added to the matrix to enforce vanishing of primal variables.

Required realizations

$$\widetilde{W} = W_{\Pi} \oplus \prod W_{\Delta}^{(k)}$$

- Construction of the primal space W_{Π} and its basis.
- We choose the so called *energy minimizing primal subspaces*.
- The basis should be at least *local* and *nodal*.
- 2 possibilities to realize the dual space W_{Δ}:
  - Transformation of basis: construction of basis, such that the primal variables vanishes.
  - Realization with local constraints: constraints are added to the matrix to enforce vanishing of primal variables.

Required realizations

In any case, a block LDL^T factorization yields:

$$\widetilde{K} = \begin{bmatrix} K_{\Pi\Pi} & K_{\Pi\Delta} \\ K_{\Delta\Pi} & K_{\Delta\Delta} \end{bmatrix} \quad (\star)$$

$$\widetilde{K}^{-1} = \begin{bmatrix} I & 0 \\ -K_{\Delta\Delta}^{-1} K_{\Delta\Pi} & I \end{bmatrix} \begin{bmatrix} S_{\Pi}^{-1} & 0 \\ 0 & K_{\Delta\Delta}^{-1} \end{bmatrix} \begin{bmatrix} I & -K_{\Pi\Delta} K_{\Delta\Delta}^{-1} \\ 0 & I \end{bmatrix},$$

where $S_{\Pi} = K_{\Pi\Pi} - K_{\Pi\Delta} K_{\Delta\Delta}^{-1} K_{\Delta\Pi}$.

- In order to apply $\widetilde{K}^{-1}$, one needs a realization of the individual subcomponents.
- If only continuous vertex values are use, one obtains $(\star)$ just by reordering. (as in the previous talk)

Required realizations

In any case, a block LDL^T factorization yields:

$$\widetilde{K} = \begin{bmatrix} K_{\Pi\Pi} & K_{\Pi\Delta} \\ K_{\Delta\Pi} & K_{\Delta\Delta} \end{bmatrix} \quad (\star)$$

$$\widetilde{K}^{-1} = \begin{bmatrix} I & 0 \\ -K_{\Delta\Delta}^{-1}K_{\Delta\Pi} & I \end{bmatrix} \begin{bmatrix} S_{\Pi}^{-1} & 0 \\ 0 & K_{\Delta\Delta}^{-1} \end{bmatrix} \begin{bmatrix} I & -K_{\Pi\Delta}K_{\Delta\Delta}^{-1} \\ 0 & I \end{bmatrix},$$

where $S_{\Pi} = K_{\Pi\Pi} - K_{\Pi\Delta}K_{\Delta\Delta}^{-1}K_{\Delta\Pi}$.

- In order to apply $\widetilde{K}^{-1}$, one needs a realization of the individual subcomponents.
- If only continuous vertex values are use, one obtains $(\star)$ just by reordering. (as in the previous talk)

Overview

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

A nice subspace W_Π and its basis

- *energy minimizing primal subspaces*: $W_\Pi := W_\Delta^{\perp_K}$
- $\rightsquigarrow W_\Pi$ is K-orthogonal to W_Δ, i.e.

$$\langle K w_\Pi, w_\Delta \rangle = 0 \quad \forall w_\Pi \in W_\Pi, w_\Delta \in W_\Delta.$$

$$\widetilde{K} = \begin{bmatrix} K_{\Pi\Pi} & 0 \\ 0 & K_{\Delta\Delta} \end{bmatrix} \Longrightarrow \widetilde{K}^{-1} = \begin{bmatrix} K_{\Pi\Pi}^{-1} & 0 \\ 0 & K_{\Delta\Delta}^{-1} \end{bmatrix}$$

A nice subspace W_Π and its basis

- *energy minimizing primal subspaces*: $W_\Pi := W_\Delta^{\perp_K}$
- $\rightsquigarrow$ W_Π is K-orthogonal to W_Δ, i.e.

$$\langle K w_\Pi, w_\Delta \rangle = 0 \quad \forall w_\Pi \in W_\Pi, w_\Delta \in W_\Delta.$$

$$\widetilde{K} = \begin{bmatrix} K_{\Pi\Pi} & 0 \\ 0 & K_{\Delta\Delta} \end{bmatrix} \Longrightarrow \widetilde{K}^{-1} = \begin{bmatrix} K_{\Pi\Pi}^{-1} & 0 \\ 0 & K_{\Delta\Delta}^{-1} \end{bmatrix}$$

Choosing $\widetilde{W}_\Pi$ and constructing the basis

Nodal basis $\widetilde{\phi}$: $\psi_i(\widetilde{\phi}_j) = \delta_{i,j}$.
For each patch k we define:

$$C^{(k)} : W^{(k)} \to \mathbb{R}^{n_{\Pi,k}}$$
$$(C^{(k)}v)_l = \psi_{i(k,l)}(v) \quad \forall v \in W^{(k)}, \forall l$$

The basis functions $\{\widetilde{\phi}_j^{(k)}\}_{j=1}^{n_{\Pi,k}}$ are the solution of:

$$\begin{bmatrix} K^{(k)} & {C^{(k)}}^T \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\phi}^{(k)} \\ \widetilde{\mu}^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

For each patch the LU factorization is computed and stored.

Choosing $\widetilde{W}_\Pi$ and constructing the basis

Nodal basis $\widetilde{\phi}$: $\psi_i(\widetilde{\phi}_j) = \delta_{i,j}$.
For each patch k we define:

$$C^{(k)} : W^{(k)} \to \mathbb{R}^{n_{\Pi,k}}$$
$$(C^{(k)}v)_l = \psi_{i(k,l)}(v) \quad \forall v \in W^{(k)}, \forall l$$

The basis functions $\{\widetilde{\phi}_j^{(k)}\}_{j=1}^{n_{\Pi,k}}$ are the solution of:

$$\begin{bmatrix} K^{(k)} & {C^{(k)}}^T \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\phi}^{(k)} \\ \widetilde{\mu}^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

For each patch the LU factorization is computed and stored.

Overview

- IETI-DP
- Implementation of primal variables
    1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
    2. Application of $\widetilde{K}^{-1}$
    3. Application of the preconditioner
- Numerical examples
- Conclusion

Application of $\widetilde{K}$

Given $f := \{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}\} \in \widetilde{W}^*$,
Find $w := \{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}\} \in \widetilde{W}: \quad w = \widetilde{K}^{-1} f$

$$\widetilde{K}^{-1} = \begin{bmatrix} K_{\Pi\Pi}^{-1} & 0 \\ 0 & K_{\Delta\Delta}^{-1} \end{bmatrix}$$

The application of $\widetilde{K}^{-1}$ reduces to

$$\boldsymbol{w}_\Pi = K_{\Pi\Pi}^{-1} \boldsymbol{f}_\Pi \qquad\qquad w_\Delta^{(k)} = {K_{\Delta\Delta}^{(k)}}^{-1} f_\Delta^{(k)} \quad \forall k = 0, \ldots, n$$

Application of ${K_{\Delta\Delta}^{(k)}}^{-1}$

The application of ${K_{\Delta\Delta}^{(k)}}^{-1}$ corresponds to

$$K^{(k)} w_k = f_{\Delta}^{(k)}$$

with the constraint $C^{(k)} w_k = 0$.

This is equivalent to:

$$\begin{bmatrix} K^{(k)} & {C^{(k)}}^T \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} w_k \\ \cdot \end{bmatrix} = \begin{bmatrix} f_{\Delta}^{(k)} \\ 0 \end{bmatrix}$$

Application of ${K_{\Delta\Delta}^{(k)}}^{-1}$

The application of ${K_{\Delta\Delta}^{(k)}}^{-1}$ corresponds to

$$K^{(k)} w_k = f_\Delta^{(k)}$$

with the constraint $C^{(k)} w_k = 0$.
This is equivalent to:

$$\begin{bmatrix} K^{(k)} & {C^{(k)}}^T \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} w_k \\ \cdot \end{bmatrix} = \begin{bmatrix} f_\Delta^{(k)} \\ 0 \end{bmatrix}$$

Application of $K_{\Pi\Pi}^{-1}$

$K_{\Pi\Pi}$ can be assembled from the patch local matrices $K_{\Pi\Pi}^{(k)}$.
Due to our special construction of $\widetilde{\phi}^{(k)}$, we have

$$\begin{aligned}\left(K_{\Pi\Pi}^{(k)}\right)_{i,j} &= \left\langle K^{(k)}\widetilde{\phi}_i^{(k)}, \widetilde{\phi}_j^{(k)}\right\rangle = -\left\langle {C^{(k)}}^T \widetilde{\boldsymbol{\mu}}_i^{(k)}, \widetilde{\phi}_j^{(k)}\right\rangle \\ &= -\left\langle \widetilde{\boldsymbol{\mu}}_i^{(k)}, C^{(k)}\widetilde{\phi}_j^{(k)}\right\rangle = -\left\langle \widetilde{\boldsymbol{\mu}}_i^{(k)}, \boldsymbol{e}_j^{(k)}\right\rangle \\ &= -\widetilde{\boldsymbol{\mu}}_{i,j}^{(k)}\end{aligned}$$

Once $K_{\Pi\Pi}$ is assembled, one can calculate its LU factorization.

Summary for application of $F = \widetilde{B} K^{-1} \widetilde{B}^T$

Given $\boldsymbol{\lambda} \in U$:

1. Application of B^T : $\{f^{(k)}\}_{k=0}^n = B^T \boldsymbol{\lambda}$
2. Application of $\widetilde{I}^T$: $\{\boldsymbol{f}_\Pi, \{f_\Delta^{(k)}\}_{k=0}^n\} = \widetilde{I}^T \left(\{f^{(k)}\}_{k=0}^n\right)$
3. Application of $\widetilde{K}^{-1}$:
   - $\boldsymbol{w}_\Pi = K_{\Pi\Pi}^{-1} \boldsymbol{f}_\Pi$
   - $w_\Delta^{(k)} = {K_{\Delta\Delta}^{(k)}}^{-1} f_\Delta^{(k)} \quad \forall k = 0, \dots, n$
4. Application of $\widetilde{I}$: $\{w^{(k)}\}_{k=0}^n = \widetilde{I} \left(\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_{k=0}^n\}\right)$
5. Application of B : $\nu = B \left(\{w^{(k)}\}_{k=0}^n\right)$

It remains to investigate $\widetilde{I}$ and $\widetilde{I}^T$.

Application of $\widetilde{I}$ and $\widetilde{I}^T$

- embedding operator: $\widetilde{I} : \widetilde{W} \to W$

$$\{\boldsymbol{w}_\Pi, \{w_\Delta^{(k)}\}_k\} \mapsto \boldsymbol{\Phi} \boldsymbol{A}^T \boldsymbol{w}_\Pi + w_\Delta$$

- partial assembling operator: $\widetilde{I}^T : W^* \to \widetilde{W}^*$

$$f \mapsto \{\boldsymbol{A}\boldsymbol{\Phi}^T f, \{(I - C^T \boldsymbol{\Phi}^T) f\}_k\}$$

$\boldsymbol{\Phi}$. . . basis of W_Π (block version)
C. . . matrix representation of primal variables W_Π (block version)
$\boldsymbol{A}$. . . *assembling operator*
$\boldsymbol{A}^T$. . . *distribution operator*

Overview

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

Application of M_{sD}^{-1}

The application of the preconditioner $M_{sD}^{-1} = B_D S B_D^T$ is basically the application of S:

$$S = \text{diag}(S^{(k)})$$
$$S^{(k)} = K_{BB}^{(k)} - K_{BI}^{(k)} (K_{II}^{(k)})^{-1} K_{IB}^{(k)}$$

The calculation of $v^{(k)} = S^{(k)} w^{(k)}$ consists of 2 steps:

1. Solve: $K_{II}^{(k)} x^{(k)} = -K_{IB}^{(k)} w^{(k)}$ (*Dirichlet problem*)
2. $v^{(k)} = K_{BB}^{(k)} w^{(k)} + K_{BI}^{(k)} x^{(k)}$

Again, a LU factorization of $K_{II}^{(k)}$ can be computed and stored.

Application of M_{sD}^{-1}

The application of the preconditioner $M_{sD}^{-1} = B_D S B_D^T$ is basically the application of S:

$$S = \mathrm{diag}(S^{(k)})$$
$$S^{(k)} = K_{BB}^{(k)} - K_{BI}^{(k)} (K_{II}^{(k)})^{-1} K_{IB}^{(k)}$$

The calculation of $v^{(k)} = S^{(k)} w^{(k)}$ consists of 2 steps:

1. Solve: $K_{II}^{(k)} x^{(k)} = -K_{IB}^{(k)} w^{(k)}$ (*Dirichlet problem*)
2. $v^{(k)} = K_{BB}^{(k)} w^{(k)} + K_{BI}^{(k)} x^{(k)}$

Again, a LU factorization of $K_{II}^{(k)}$ can be computed and stored.

Overview

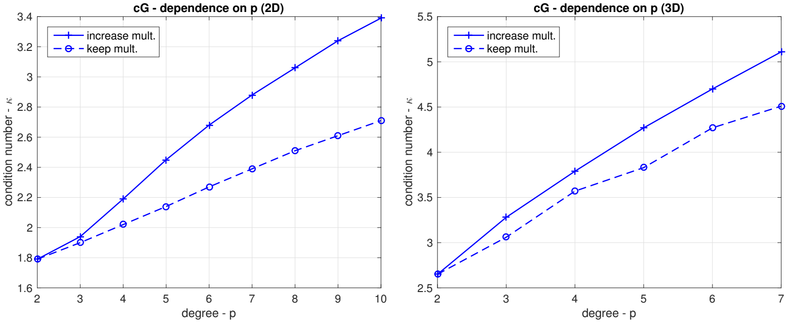
- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

Example with $\alpha \equiv 1$, $p = 4$

2D				3D			
$\mathcal{V}$				$\mathcal{V}$			
#dofs	H/h	κ	It.	#dofs	H/h	κ	It.
3350	11	11.4	23	3100	3	29.9	28
9614	19	14.5	25	7228	6	75.8	38
31742	35	18.1	27	23680	12	161	45
114398	67	22.2	28	106168	25	370	64
433310	131	26.6	30				
$\mathcal{V}+\mathcal{E}$				$\mathcal{V}+\mathcal{E}+\mathcal{F}$			
3350	11	2.02	13	3100	3	3.1	16
9614	19	2.39	14	7228	6	4.0	18
31742	35	2.85	16	23680	12	5.0	21
114398	67	3.37	17	106168	25	6.4	23
433310	131	3.95	18				

p-dependence: 2D + 3D & different multiplicity

- keeping multiplicity & increasing smoothness (- - - -)
- increasing multiplicity & keeping smoothness (——)



Overview

- IETI-DP
- Implementation of primal variables
  1. Choosing $\widetilde{W}_{\Pi}$ and constructing the basis
  2. Application of $\widetilde{K}^{-1}$
  3. Application of the preconditioner
- Numerical examples
- Conclusion

Conclusion and Extensions

- Also other primal variables can be realized in an efficient way.
- Provides application of IETI-DP to 3D problems.
- With suitable scaling $\rightsquigarrow$ robustness wrt. jumping coefficients.
- Method can be combined with dG-formulation.
- Parallelization wrt. patches (distributed memory setting).
- Instead of LU-factorization, one can use Multigrid (inexact IETI).
- Extension to nonlinear problems
  - Apply IETI to linearized equation
  - Apply IETI to non-linear equation and use Newton on each subdomain.