An Algorithm for Solving Non-Symmetric Saddle-Point Systems

Jaroslav Haslinger, Charles University, Prague

Tomáš Kozubek, VŠB–TU Ostrava

Radek Kučera, VŠB–TU Ostrava

HARRACHOV, August 2007

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

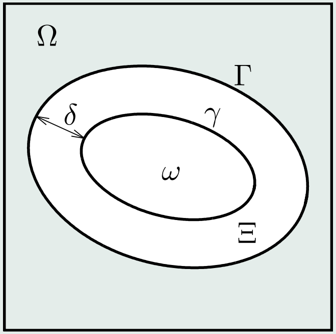
MODEL PROBLEM 1: Dirichlet problem

$$-\Delta u = f \qquad \text{on } \omega \tag{1}$$

$$u = g \qquad \text{in } \gamma \equiv \partial\omega \tag{2}$$

Fictitious domain method (FDM):

PDE (1) is solved on the fictitious domain Ω, $\overline{\omega} \subset \Omega$, with a simple geometry. The corresponding stiffness matrix **A** is structured. The original boundary conditions (2) on γ are enforced by Lagrange multipliers or control variables.



Classical FDM with $\Gamma = \gamma$

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_\gamma^\top \\ \mathbf{B}_\gamma & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda}_\gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}$$

Smooth FDM with $\Gamma \neq \gamma$

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_\Gamma^\top \\ \mathbf{B}_\gamma & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}$$

MODEL PROBLEM 2: Signorini problem

$$-\Delta u = f \qquad \text{on } \omega \tag{3}$$

$$u - g \geq 0, \quad \frac{\partial u}{\partial n_\gamma} \geq 0, \quad (u-g)\frac{\partial u}{\partial n_\gamma} = 0 \qquad \text{in } \gamma \equiv \partial\omega \tag{4}$$

FDM formulation uses the non-differentiable $\max$-function to express BC (4):

$$\left.\begin{aligned} &\mathbf{A}\mathbf{u} + \mathbf{B}_\Gamma \boldsymbol{\lambda}_\Gamma = \mathbf{f} \\ &\mathbf{C}_{\gamma,i}\mathbf{u} = \max\{0, \mathbf{C}_{\gamma,i}\mathbf{u} - \rho(\mathbf{B}_{\gamma,i}\mathbf{u} - \mathbf{g}_i)\}, \quad i = 1, \ldots, m \end{aligned}\right\} \tag{5}$$

where $\mathbf{B}_{\gamma,i}$, $\mathbf{B}_{\Gamma,i}$ and $\mathbf{C}_{\gamma,i}$ are rows of Dirichlet and Neumann trace matrices, respectively.

The equations (5) can be solved by the semi-smooth Newton method, in which

$$\textbf{Jacobian} = \begin{pmatrix} \mathbf{A} & \mathbf{B}_\Gamma^\top \\ \partial G(\mathbf{u}) & \mathbf{0} \end{pmatrix}$$

is determined by the generalized derivative $\partial G(\mathbf{u})$.

MODEL PROBLEM 2: Newton method = Active set algorithm

(0) Set $k := 1$, $\rho > 0$, $\varepsilon_u > 0$, $\mathbf{u}^{(0)} \in \mathbb{R}^n$, $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^m$.

(1) Define the inactive and active sets by:

$$\begin{aligned}\mathcal{I}^k &:= \{i : \mathbf{C}_{\gamma,i}\mathbf{u}^{k-1} - \rho(\mathbf{B}_{\gamma,i}\mathbf{u}^{k-1} - \mathbf{g}_i) \le 0\} \\ \mathcal{A}^k &:= \{i : \mathbf{C}_{\gamma,i}\mathbf{u}^{k-1} - \rho(\mathbf{B}_{\gamma,i}\mathbf{u}^{k-1} - \mathbf{g}_i) > 0\}\end{aligned}$$

(2) Solve:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_\Gamma^\top \\ \mathbf{B}_{\gamma,\mathcal{A}^k} & \mathbf{0} \\ \mathbf{C}_{\gamma,\mathcal{I}^k} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}^k \\ \boldsymbol{\lambda}_\Gamma^k \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g}_{\mathcal{A}^k} \\ \mathbf{0} \end{pmatrix}$$

(3) If $\|\mathbf{u}^k - \mathbf{u}^{k-1}\| / \|\mathbf{u}^k\| \le \varepsilon_u$, return $\mathbf{u} := \mathbf{u}^k$.

(4) Set $k := k + 1$, and go to step (1).

Remark: The mixed Dirichlet-Neumann problem is solved in each Newton step, that is described by the non-symmetric saddle-point system.

FORMULATION: Non-symmetric sadle-point system

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{B}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}$$

General assumptions

$\mathbf{A}$ . . . non-symmetric $(n \times n)$–matrix

. . . singular with $p = \dim Ker\, \mathbf{A}$

$\mathbf{B}_1, \ \mathbf{B}_2$ . . . full rank $(m \times n)$–matrices

. . . $\mathbf{B}_1 \neq \mathbf{B}_2$

Special FDM assumptions

- n is large ($n = 4198401$)
- $m \ll n$ ($m = 360$)
- $p \ll m$ ($p = 1$)
- $\mathbf{A}$ is structured so that actions of $\mathbf{A}^\dagger$ or ($\mathbf{A}^{-1}$) are ”cheap”
- $\mathbf{B}_1, \mathbf{B}_2$ are highly sparse so that their actions are ”cheap”

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

ALGORITHMS based on the Schur complement reduction

Case 1: **A** non-singular, symmetric

Case 2: **A** non-singular, non-symmetric

Case 3: **A** singular, symmetric

Case 4: **A** singular, non-symmetric

Case 1: **A** non-singular, symmetric

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \quad \begin{array}{l} \Longrightarrow \quad \mathbf{u} = \mathbf{A}^{-1}(\mathbf{f} - \mathbf{B}^\top \boldsymbol{\lambda}) \\ \Longrightarrow \quad \underbrace{\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\top}_{\textit{negative Schur complement } \mathbf{S}} \boldsymbol{\lambda} = \mathbf{B}\mathbf{A}^{-1}\mathbf{f} - \mathbf{g} \end{array}$$

Algorithm

1° Assemble $\mathbf{d} := \mathbf{B}\mathbf{A}^{-1}\mathbf{f} - \mathbf{g}$.

2° Solve iteratively $\mathbf{S}\boldsymbol{\lambda} = \mathbf{d}$ with $\mathbf{S} := \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\top$.

3° Assemble $\mathbf{u} := \mathbf{A}^{-1}(\mathbf{f} - \mathbf{B}^\top \boldsymbol{\lambda})$.

If **A** is positive defined, then CGM can be used.

Matrix-vector products $\mathbf{S}\boldsymbol{\mu}$ are performed by:

$$\mathbf{S}\boldsymbol{\mu} := \left(\mathbf{B}\left(\mathbf{A}^{-1}\left(\mathbf{B}^\top \boldsymbol{\mu}\right)\right)\right)$$

Case 2: **A** non-singular, non-symmetric

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{B}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \quad \begin{array}{ll} \Longrightarrow & \mathbf{u} = \mathbf{A}^{-1}(\mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda}) \\ \Longrightarrow & \underbrace{\mathbf{B}_2 \mathbf{A}^{-1} \mathbf{B}_1^\top}_{\textit{negative Schur complement } \mathbf{S}} \boldsymbol{\lambda} = \mathbf{B}_2 \mathbf{A}^{-1} \mathbf{f} - \mathbf{g} \end{array}$$

Algorithm is analogous.

- an iterative method for non-symmetric matrices is required (GMRES, BiCG, BiCGSTAB, ...)

$$\mathcal{A} := \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{B}_2 & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{B}_2 \mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{0} & -\mathbf{S} \end{pmatrix}$$

Theorem 1 Let **A** be non-singular. Then $\mathcal{A}$ is invertible iff **S** is invertible.

Case 3: $\mathbf{A}$ singular, symmetric

- a generalized inverse $\mathbf{A}^\dagger$ satisfying $\mathbf{A} = \mathbf{A}\mathbf{A}^\dagger\mathbf{A}$
- an $(n \times p)$–matrix $\mathbf{N}$ whose columns span *Ker* $\mathbf{A}$

$$\mathbf{A}\mathbf{u} + \mathbf{B}^\top\boldsymbol{\lambda} = \mathbf{f} \quad \Longleftrightarrow \quad \mathbf{f} - \mathbf{B}^\top\boldsymbol{\lambda} \in \mathit{Im}\,\mathbf{A} \perp \mathit{Ker}\,\mathbf{A}$$

$$\Updownarrow \qquad\qquad\qquad \Updownarrow$$

$$\mathbf{u} = \mathbf{A}^\dagger(\mathbf{f} - \mathbf{B}^\top\boldsymbol{\lambda}) + \mathbf{N}\boldsymbol{\alpha} \qquad\qquad \mathbf{N}^\top(\mathbf{f} - \mathbf{B}^\top\boldsymbol{\lambda}) = \mathbf{0}$$

$$\&$$

The reduced system:

$$\mathbf{B}\mathbf{u} = \mathbf{g} \qquad\qquad \begin{pmatrix} \mathbf{B}\mathbf{A}^\dagger\mathbf{B}^\top & -\mathbf{B}\mathbf{N} \\ -\mathbf{N}^\top\mathbf{B}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{A}^\dagger\mathbf{f} - \mathbf{g} \\ -\mathbf{N}^\top\mathbf{f} \end{pmatrix}$$

$$\Downarrow$$

$$\mathbf{B}\mathbf{A}^\dagger\mathbf{B}^\top\boldsymbol{\lambda} - \mathbf{B}\mathbf{N}\boldsymbol{\alpha} = \mathbf{B}\mathbf{A}^\dagger\mathbf{f} - \mathbf{g}$$

If $\mathbf{A}$ is positive semidefinite, then it corresponds to the algebra in FETI DDM.

Case 4: $\mathbf{A}$ singular, non-symmetric

- a generalized inverse $\mathbf{A}^\dagger$
- columns of $(n \times p)$–matrices $\mathbf{N}$, $\mathbf{M}$ span *Ker* $\mathbf{A}$, *Ker* $\mathbf{A}^\top$, respectively

$$\mathbf{A}\mathbf{u} + \mathbf{B}_1^\top \boldsymbol{\lambda} = \mathbf{f} \quad \Longleftrightarrow \quad \mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda} \in \mathit{Im}\, \mathbf{A} \perp \mathit{Ker}\, \mathbf{A}^\top$$

$$\Updownarrow \qquad\qquad \Updownarrow$$

$$\mathbf{u} = \mathbf{A}^\dagger(\mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda}) + \mathbf{N}\boldsymbol{\alpha} \qquad \mathbf{M}^\top(\mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda}) = \mathbf{0}$$

$$\&$$

The reduced system:

$$\begin{pmatrix} \mathbf{B}_2\mathbf{A}^\dagger\mathbf{B}_1^\top & -\mathbf{B}_2\mathbf{N} \\ -\mathbf{M}^\top\mathbf{B}_1^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_2\mathbf{A}^\dagger\mathbf{f} - \mathbf{g} \\ -\mathbf{M}^\top\mathbf{f} \end{pmatrix}$$

$$\mathbf{B}_2\mathbf{u} = \mathbf{g}$$

$$\Downarrow$$

$$\mathbf{B}_2\mathbf{A}^\dagger\mathbf{B}_1^\top \boldsymbol{\lambda} - \mathbf{B}_2\mathbf{N}\boldsymbol{\alpha} = \mathbf{B}_2\mathbf{A}^\dagger\mathbf{f} - \mathbf{g}$$

Theorem 2 The saddle-point matrix $\mathcal{A} := \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{B}_2 & \mathbf{0} \end{pmatrix}$ is invertible iff

$$\left. \begin{array}{l} \mathbf{B}_1 \text{ has full row-rank} \\ \mathit{Ker}\,\mathbf{A} \cap \mathit{Ker}\,\mathbf{B}_2 = \{\mathbf{0}\} \\ \mathbf{A}\,\mathit{Ker}\,\mathbf{B}_2 \cap \mathit{Im}\,\mathbf{B}_1^\top = \{\mathbf{0}\} \end{array} \right\} \quad \text{(NSC)}$$

Remark: The third equality is equivalent to the MinMax condition that is well-known in the continuous setting:

$$\exists C > 0 : \quad \min_{\mathbf{u} \in \mathit{Ker}\,\mathbf{B}_2, \mathbf{u} \neq 0} \; \max_{\mathbf{v} \in \mathit{Ker}\,\mathbf{B}_1, \mathbf{v} \neq 0} \frac{\mathbf{v}^\top \mathbf{A} \mathbf{u}}{\|\mathbf{v}\| \|\mathbf{u}\|} \geq C$$

The generalized Schur complement: the matrix of the reduced system

$$\mathcal{S} := \begin{pmatrix} -\mathbf{B}_2\mathbf{A}^{\dagger}\mathbf{B}_1^{\top} & \mathbf{B}_2\mathbf{N} \\ \mathbf{M}^{\top}\mathbf{B}_1^{\top} & \mathbf{0} \end{pmatrix}$$

Theorem 3 The following three statements are equivalent:

- The necessary and sufficient condition (NSC) holds.
- $\mathcal{A}$ is invertible.
- $\mathcal{S}$ is invertible.

Remark: The generalized Schur complement $\mathcal{S}$ is not defined uniquely.

First step of the algorithm = Schur complement reduction:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^\top \\ \mathbf{B}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \Longleftrightarrow \left\{ \begin{array}{l} \begin{pmatrix} \mathbf{F} & \mathbf{G}_1^\top \\ \mathbf{G}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix} \\ \\ \mathbf{u} = \mathbf{A}^\dagger(\mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda}) + \mathbf{N}\boldsymbol{\alpha} \end{array} \right.$$

How to solve the reduced system again with the saddle-point structure?

- matrix-vector products via $\mathbf{F}\boldsymbol{\mu} := \left(\mathbf{B}_2 \left(\mathbf{A}^\dagger \left(\mathbf{B}_1^\top \boldsymbol{\mu}\right)\right)\right)$
- $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{d}$, $\mathbf{e}$ may be assembled

1) Again the Schur complement reduction (the second elimination)

$\mathbf{E}\boldsymbol{\alpha} = \mathbf{r}$ with $\mathbf{E} := \mathbf{G}_2\mathbf{F}^{-1}\mathbf{G}_1^\top$, then $\boldsymbol{\lambda} := \mathbf{F}^{-1}(\mathbf{d} - \mathbf{G}_1^\top\boldsymbol{\alpha})$ and $\mathbf{u}$

(R.K., Appl. Math. 50(2005))

2) Null-space method

(Farhat, Mandel, Roux: FETI DDM, 1994)

Second step of the algorithm = Null-space method:

$$\begin{pmatrix} \mathbf{F} & \mathbf{G}_1^\top \\ \mathbf{G}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}$$

Two orthogonal projectors $\mathbf{P}_1$ and $\mathbf{P}_2$ onto $Ker\,\mathbf{G}_1$ and $Ker\,\mathbf{G}_2$:

$$\mathbf{P}_k : \mathbb{R}^m \mapsto Ker\,\mathbf{G}_k, \qquad \mathbf{P}_k := \mathbf{I} - \mathbf{G}_k^\top (\mathbf{G}_k \mathbf{G}_k^\top)^{-1} \mathbf{G}_k, \qquad k = 1, 2$$

Property: $$Ker\,\mathbf{P}_k = Im\,\mathbf{G}_k^\top \quad \Longleftrightarrow \quad \mathbf{P}_k \mathbf{G}_k^\top = \mathbf{0}$$

- $\mathbf{P}_1$ splits the saddle-point structure: $\mathbf{P}_1 \mathbf{F} \boldsymbol{\lambda} + \mathbf{P}_1 \mathbf{G}_1^\top \boldsymbol{\alpha} = \mathbf{P}_1 \mathbf{d}$

$$\mathbf{P}_1 \mathbf{F} \boldsymbol{\lambda} = \mathbf{P}_1 \mathbf{d}, \quad \mathbf{G}_2 \boldsymbol{\lambda} = \mathbf{e}, \quad \boldsymbol{\alpha} := (\mathbf{G}_1 \mathbf{G}_1^\top)^{-1} (\mathbf{G}_1 \mathbf{d} - \mathbf{G}_1 \mathbf{F} \boldsymbol{\lambda})$$

- $\mathbf{P}_2$ decomposes $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{Im} + \boldsymbol{\lambda}_{Ker}, \quad \boldsymbol{\lambda}_{Im} \in Im\,\mathbf{G}_2^\top, \quad \boldsymbol{\lambda}_{Ker} \in Ker\,\mathbf{G}_2$

  At first: $\mathbf{G}_2 \boldsymbol{\lambda} = \mathbf{G}_2 \boldsymbol{\lambda}_{Im} = \mathbf{e} \quad \Longrightarrow \quad \boldsymbol{\lambda}_{Im} := \mathbf{G}_2^\top (\mathbf{G}_2 \mathbf{G}_2^\top)^{-1} \mathbf{e}$

  At second: $\mathbf{P}_1 \mathbf{F} \boldsymbol{\lambda}_{Ker} = \mathbf{P}_1 (\mathbf{d} - \mathbf{F} \boldsymbol{\lambda}_{Im})$ on $Ker\,\mathbf{G}_2$

Theorem 4 Let $\mathcal{A}$ be invertibel. The linear operator $\mathbf{P}_1\mathbf{F}\colon\ Ker\,\mathbf{G}_2 \mapsto\ Ker\,\mathbf{G}_1$ is invertible.

Proof.

As both null-spaces $Ker\,\mathbf{G}_1$ and $Ker\,\mathbf{G}_2$ have the same dimension $m - p$, it is enough to prove that $\mathbf{P}_1\mathbf{F}$ is injective.

Let $\boldsymbol{\mu} \in\ Ker\,\mathbf{G}_2$ be such that $\mathbf{P}_1\mathbf{F}\boldsymbol{\mu} = \mathbf{0}$. Then $\mathbf{F}\boldsymbol{\mu} \in\ Ker\,\mathbf{P}_1 =\ Im\,\mathbf{G}_1^\top$ and, therefore, there is $\boldsymbol{\beta} \in \mathbb{R}^p$ so that

$$\mathbf{F}\boldsymbol{\mu} = \mathbf{G}_1^\top\boldsymbol{\beta} \quad \text{and} \quad \mathbf{G}_2\boldsymbol{\mu} = \mathbf{0}.$$

We obtain

$$\begin{pmatrix} \mathbf{F} & \mathbf{G}_1^\top \\ \mathbf{G}_2 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ -\boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$

where the matrix is the (negative) Schur complement $-\mathcal{S}$ that is invertible iff $\mathcal{A}$ is invertibel. Therefore $\boldsymbol{\mu} = \mathbf{0}$.

Algorithm PSCM

Step 1.a: Assemble $\mathbf{G}_1 := -\mathbf{N}^\top \mathbf{B}_2^\top,\ \mathbf{G}_2 := -\mathbf{M}^\top \mathbf{B}_1^\top.$

Step 1.b: Assemble $\mathbf{d} := \mathbf{B}_2 \mathbf{A}^\dagger \mathbf{f} - \mathbf{g},\ \mathbf{e} := -\mathbf{M}^\top \mathbf{f}.$

Step 1.c: Assemble $\mathbf{H}_1 := \left(\mathbf{G}_1 \mathbf{G}_1^\top\right)^{-1},\ \mathbf{H}_2 := \left(\mathbf{G}_2 \mathbf{G}_2^\top\right)^{-1}.$

Step 1.d: Assemble $\boldsymbol{\lambda}_{Im} := \mathbf{G}_2^\top \mathbf{H}_2 \mathbf{e},\ \tilde{\mathbf{d}} := \mathbf{P}_1 (\mathbf{d} - \mathbf{F} \boldsymbol{\lambda}_{Im}).$

Step 1.e: Solve $\mathbf{P}_1 \mathbf{F} \boldsymbol{\lambda}_{Ker} = \tilde{\mathbf{d}}$ on $Ker\,\mathbf{G}_2.$

Step 1.f: Assemble $\boldsymbol{\lambda} := \boldsymbol{\lambda}_{Im} + \boldsymbol{\lambda}_{Ker}.$

Step 2: Assemble $\boldsymbol{\alpha} := \mathbf{H}_1 \mathbf{G}_1 (\mathbf{d} - \mathbf{F} \boldsymbol{\lambda}).$

Step 3: Assemble $\mathbf{u} := \mathbf{A}^\dagger (\mathbf{f} - \mathbf{B}_1^\top \boldsymbol{\lambda}) + \mathbf{N} \boldsymbol{\alpha}.$

- an iterative projected Krylov subspace method for non-symmetric operators can be used in Step 1.e

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

$$\text{Find} \quad \boldsymbol{\lambda} \in \mathbb{R}^m \quad \text{so that} \quad \mathbf{F}\boldsymbol{\lambda} = \mathbf{d}, \quad \text{where} \quad \mathbf{d} \in \mathbb{R}^m.$$

Algorithm BiCGSTAB $[\epsilon, \boldsymbol{\lambda}^0, \mathbf{F}, \mathbf{d}\,] \rightarrow \boldsymbol{\lambda}$

Initialize: $\mathbf{r}^0 := \mathbf{d} - \mathbf{F}\boldsymbol{\lambda}^0$, $\mathbf{p}^0 := \mathbf{r}^0$, $\tilde{\mathbf{r}}^0$ arbitrary, $k := 0$

While $\|\mathbf{r}^k\| > \epsilon$

1° $\quad \tilde{\mathbf{p}}^k := \mathbf{F}\mathbf{p}^k$

2° $\quad \alpha_k := (\mathbf{r}^k)^\top \tilde{\mathbf{r}}^0 / (\tilde{\mathbf{p}}^k)^\top \tilde{\mathbf{r}}^0$

3° $\quad \mathbf{s}^k := \mathbf{r}^k - \alpha_k \tilde{\mathbf{p}}^k$

4° $\quad \tilde{\mathbf{s}}^k := \mathbf{F}\mathbf{s}^k$

5° $\quad \omega_k := (\tilde{\mathbf{s}}^k)^\top \mathbf{s}^k / (\tilde{\mathbf{s}}^k)^\top \tilde{\mathbf{s}}^k$

6° $\quad \boldsymbol{\lambda}^{k+1} := \boldsymbol{\lambda}^k + \alpha_k \mathbf{p}^k + \omega_k \mathbf{s}^k$

7° $\quad \mathbf{r}^{k+1} := \mathbf{s}^k - \omega_k \tilde{\mathbf{s}}^k$

8° $\quad \beta_{k+1} := (\alpha_k / \omega_k)(\mathbf{r}^{k+1})^\top \tilde{\mathbf{r}}^0 / (\mathbf{r}^k)^\top \tilde{\mathbf{r}}^0$

9° $\quad \mathbf{p}^{k+1} := \mathbf{r}^{k+1} + \beta_{k+1}(\mathbf{p}^k - \omega_k \tilde{\mathbf{p}}^k)$

10° $\quad k := k + 1$

end

(Van der Vorst, 1992)

Find $\boldsymbol{\lambda} \in Ker\,\mathbf{G}_2$ so that $\mathbf{P}_1\mathbf{F}\boldsymbol{\lambda} = \tilde{\mathbf{d}}$, where $\tilde{\mathbf{d}} \in Ker\,\mathbf{G}_1$.

<u>*Algorithm ProjBiCGSTAB*</u> $[\epsilon, \boldsymbol{\lambda}^0, \mathbf{F}, \mathbf{P}_1, \mathbf{P}_2, \tilde{\mathbf{d}}] \to \boldsymbol{\lambda}$

Initialize: $\boldsymbol{\lambda}^0 \in Ker\,\mathbf{G}_2$, $\mathbf{r}^0 := \tilde{\mathbf{d}} - \mathbf{P}_1\mathbf{F}\boldsymbol{\lambda}^0$, $\mathbf{p}^0 := \mathbf{r}^0$, $\tilde{\mathbf{r}}^0$ arbitrary, $k := 0$

While $\|\mathbf{r}^k\| > \epsilon$

1° $\tilde{\mathbf{p}}^k := \mathbf{P}_1\mathbf{F}\mathbf{p}^k$

2° $\alpha_k := (\mathbf{r}^k)^\top\tilde{\mathbf{r}}^0 / (\tilde{\mathbf{p}}^k)^\top\tilde{\mathbf{r}}^0$

3° $\mathbf{s}^k := \mathbf{r}^k - \alpha_k\tilde{\mathbf{p}}^k$

4° $\tilde{\mathbf{s}}^k := \mathbf{P}_1\mathbf{F}\mathbf{s}^k$

5° $\omega_k := (\tilde{\mathbf{s}}^k)^\top\mathbf{s}^k / (\tilde{\mathbf{s}}^k)^\top\tilde{\mathbf{s}}^k$

6° $\boldsymbol{\lambda}^{k+1} := \boldsymbol{\lambda}^k + \alpha_k\mathbf{P}_2\mathbf{p}^k + \omega_k\mathbf{P}_2\mathbf{s}^k$

7° $\mathbf{r}^{k+1} := \mathbf{s}^k - \omega_k\tilde{\mathbf{s}}^k$

8° $\beta_{k+1} := (\alpha_k/\omega_k)(\mathbf{r}^{k+1})^\top\tilde{\mathbf{r}}^0 / (\mathbf{r}^k)^\top\tilde{\mathbf{r}}^0$

9° $\mathbf{p}^{k+1} := \mathbf{r}^{k+1} + \beta_{k+1}(\mathbf{p}^k - \omega_k\tilde{\mathbf{p}}^k)$

10° $k := k + 1$

end

Formally solve $\mathbf{P}_2\mathbf{P}_1\mathbf{F}\boldsymbol{\lambda} = \mathbf{P}_2\tilde{\mathbf{d}}$, with $\boldsymbol{\lambda}^0 \in Ker\,\mathbf{G}_2$.

<u>*Algorithm ProjBiCGSTAB*</u> $[\epsilon, \boldsymbol{\lambda}^0, \mathbf{F}, \mathbf{P}_1, \mathbf{P}_2, \tilde{\mathbf{d}}] \rightarrow \boldsymbol{\lambda}$

Initialize: $\boldsymbol{\lambda}^0 \in Ker\,\mathbf{G}_2$, $\mathbf{r}^0 := \mathbf{P}_2\tilde{\mathbf{d}} - \mathbf{P}_2\mathbf{P}_1\mathbf{F}\boldsymbol{\lambda}^0$, $\mathbf{p}^0 := \mathbf{r}^0$, $\tilde{\mathbf{r}}^0$, $k := 0$

While $\|\mathbf{r}^k\| > \epsilon$

1° $\tilde{\mathbf{p}}^k := \mathbf{P}_2\mathbf{P}_1\mathbf{F}\mathbf{p}^k$

2° $\alpha_k := (\mathbf{r}^k)^\top\tilde{\mathbf{r}}^0/(\tilde{\mathbf{p}}^k)^\top\tilde{\mathbf{r}}^0$

3° $\mathbf{s}^k := \mathbf{r}^k - \alpha_k\tilde{\mathbf{p}}^k$

4° $\tilde{\mathbf{s}}^k := \mathbf{P}_2\mathbf{P}_1\mathbf{F}\mathbf{s}^k$

5° $\omega_k := (\tilde{\mathbf{s}}^k)^\top\mathbf{s}^k/(\tilde{\mathbf{s}}^k)^\top\tilde{\mathbf{s}}^k$

6° $\boldsymbol{\lambda}^{k+1} := \boldsymbol{\lambda}^k + \alpha_k\mathbf{p}^k + \omega_k\mathbf{s}^k$

7° $\mathbf{r}^{k+1} := \mathbf{s}^k - \omega_k\tilde{\mathbf{s}}^k$

8° $\beta_{k+1} := (\alpha_k/\omega_k)(\mathbf{r}^{k+1})^\top\tilde{\mathbf{r}}^0/(\mathbf{r}^k)^\top\tilde{\mathbf{r}}^0$

9° $\mathbf{p}^{k+1} := \mathbf{r}^{k+1} + \beta_{k+1}(\mathbf{p}^k - \omega_k\tilde{\mathbf{p}}^k)$

10° $k := k + 1$

end

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

Consider a family of nested partitions of the fictitious domain Ω with stepsizes:

$$h_j, \quad 0 \leq j \leq J$$

- the first iterate is determined by the result from the nearest lower level
- the terminating tolerance ϵ on each level is $\epsilon := \nu h_j^p$

Algorithm: Hierarchical Multigrid Scheme

Initialize: Let $\boldsymbol{\lambda}_{Ker}^{0,(0)} \in Ker\, \mathbf{G}_2^{(0)}$ be given.

ProjBiCGSTAB$[\nu h_0^p, \boldsymbol{\lambda}_{Ker}^{0,(0)}, \mathbf{F}^{(0)}, \mathbf{P}_1^{(0)}, \mathbf{P}_2^{(0)}, \tilde{\mathbf{d}}^{(0)}] \longrightarrow \boldsymbol{\lambda}_{Ker}^{(0)}$.

For $j = 1, \ldots, J,$

1° prolongate $\boldsymbol{\lambda}_{Ker}^{(j-1)} \longrightarrow \tilde{\boldsymbol{\lambda}}_{Ker}^{0,(j)}$

2° project $\tilde{\boldsymbol{\lambda}}_{Ker}^{0,(j)} \longrightarrow \boldsymbol{\lambda}_{Ker}^{0,(j)} := \mathbf{P}_2^{(j)} \tilde{\boldsymbol{\lambda}}_{Ker}^{0,(j)}$

3° ProjBiCGSTAB$[\nu h_j^p, \boldsymbol{\lambda}_{Ker}^{0,(j)}, \mathbf{F}^{(j)}, \mathbf{P}_1^{(j)}, \mathbf{P}_2^{(j)}, \tilde{\mathbf{d}}^{(j)}] \longrightarrow \boldsymbol{\lambda}_{Ker}^{(j)}$

end

Return: $\boldsymbol{\lambda}_{Ker} := \boldsymbol{\lambda}_{Ker}^{(J)}$.

Motivation

$\mathbf{u}^*$ ... exact solution of PDE problem
$\mathbf{u}$ ... FEM approximation with respect to h with the convergence rate p

$$\|\mathbf{u}^* - \mathbf{u}\| \leq Ch^p, \qquad \mathbf{Au} = \mathbf{f}$$

$\mathbf{u}^k$ ... the k-th iteration

$$\mathbf{u}^k \longrightarrow \mathbf{u}, \qquad \mathbf{Au}^k = \mathbf{f} + \mathbf{r}^k$$

When should be iterations terminated? $\|\mathbf{r}^k\| \leq \epsilon$, $\epsilon = ???$

$$\begin{aligned} \|\mathbf{u}^* - \mathbf{u}^k\| &\leq \|\mathbf{u}^* - \mathbf{u}\| + \|\mathbf{u} - \mathbf{u}^k\| \\ &\leq Ch^p + \|\mathbf{A}^{-1}\mathbf{r}^k\| \\ &\leq Ch^p + \|\mathbf{A}^{-1}\| \cdot \epsilon \\ &\leq (C + \|\mathbf{A}^{-1}\|\nu)h^p \quad \text{if} \quad \epsilon := \nu h^p \end{aligned}$$

Control parameter ν may by choosen experimentally; $\nu \approx KC/\|\mathbf{A}^{-1}\|$.

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

Circulant matrices and Fourier transform

$$\mathbf{A} = \begin{pmatrix} a_1 & a_n & \dots & a_2 \\ a_2 & a_1 & \dots & a_3 \\ a_3 & a_2 & \dots & a_4 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_1 \end{pmatrix} = \left(\mathbf{a}, \mathbf{Ta}, \mathbf{T}^2\mathbf{a}, \cdots, \mathbf{T}^{n-1}\mathbf{a}\right)$$

$$\widehat{\mathcal{T}_k f}(\omega) = \int_R f(x-k)e^{-ix\omega}\, dx = e^{-ik\omega}\widehat{f}(\omega)$$

$$\mathbf{XA} = \left(\mathbf{Dx}_0, \mathbf{Dx}_1, \mathbf{Dx}_2, \ \cdots, \mathbf{Dx}_{n-1}\right) = \mathbf{DX}$$

Lamma: Let **A** be circulant. Then

$$\mathbf{A} = \mathbf{X}^{-1}\mathbf{DX},$$

where **X** is the DFT matrix and $\mathbf{D} = diag(\widehat{\mathbf{a}})$, $\widehat{\mathbf{a}} = \mathbf{Xa}$, $\mathbf{a} = \mathbf{A}(:, 1)$.

Multiplying procedure: $\mathbf{A}^{\dagger}\mathbf{v} := \mathbf{X}^{-1}\left(\mathbf{D}^{\dagger}\left(\mathbf{X}\mathbf{v}\right)\right)$ ... Moore-Penrose

$$\left.\begin{array}{ll} 0^\circ & \mathbf{d} := \texttt{fft}(\mathbf{a}) \\ 1^\circ & \mathbf{v} := \texttt{fft}(\mathbf{v}) \\ 2^\circ & \mathbf{v} := \mathbf{v}.*\mathbf{d}^{-1} \\ 3^\circ & \mathbf{A}^{\dagger}\mathbf{v} := \texttt{ifft}(\mathbf{v}) \end{array}\right\} \mathcal{O}(2n\log_2 n)$$

Multiplying procedures: $\mathbf{N}\boldsymbol{\alpha}$, $\mathbf{N}^{\top}\mathbf{v}$ (and $\mathbf{M}\boldsymbol{\alpha}$, $\mathbf{M}^{\top}\mathbf{v}$)

As $\mathbf{AN} = \mathbf{0}$, the matrix $\mathbf{N}$ may be formed by eigenvectors corresponding to zero eigenvalues.

$$\mathbf{I} - \mathbf{D}\mathbf{D}^{\dagger} = \mathrm{diag}(1,1,1,0,\ldots,0) \implies \mathbf{X}^{-1} = (\mathbf{N},\mathbf{Y}), \quad \mathbf{X}^{-1} = \begin{pmatrix} \mathbf{N}^{\top} \\ \mathbf{Y} \end{pmatrix}$$

Therefore we can define the operation: $\texttt{ind}(\boldsymbol{\alpha}) = \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^n$

$$\begin{array}{ll} 1^\circ & \mathbf{v}_{\boldsymbol{\alpha}} := \texttt{ind}(\boldsymbol{\alpha}) \\ 2^\circ & \mathbf{N}\boldsymbol{\alpha} := \texttt{ifft}(\mathbf{v}_{\boldsymbol{\alpha}}) \end{array} \qquad \left.\begin{array}{ll} 1^\circ & \mathbf{v} := \texttt{ifft}(\mathbf{v}) \\ 2^\circ & \mathbf{N}^{\top}\mathbf{v} := \texttt{ind}^{-1}(\mathbf{v}) \end{array}\right\} \mathcal{O}(n\log_2 n)$$

Kronecker product of matrices: $\mathbf{A}_x \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{A}_y \in \mathbb{R}^{n_y \times n_y}$

$$\mathbf{A}_x \otimes \mathbf{A}_y = \begin{pmatrix} a^y_{11}\mathbf{A}_x & \dots & a^y_{1n_y}\mathbf{A}_x \\ \vdots & \ddots & \vdots \\ a^y_{n_y1}\mathbf{A}_x & \dots & a^y_{n_yn_y}\mathbf{A}_x \end{pmatrix}$$

Lemma 1:

$$\begin{aligned} (\mathbf{A}_x \otimes \mathbf{A}_y)(\mathbf{B}_x \otimes \mathbf{B}_y) &= \mathbf{A}_x\mathbf{B}_x \otimes \mathbf{A}_y\mathbf{B}_y \\ (\mathbf{A}_x \otimes \mathbf{A}_y)^\dagger &= \mathbf{A}_x^\dagger \otimes \mathbf{A}_y^\dagger \\ \mathbf{N} &= \mathbf{N}_x \otimes \mathbf{N}_y \end{aligned}$$

Lemma 2: $(\mathbf{A}_x \otimes \mathbf{A}_y)\mathbf{v} = \mathtt{vec}(\mathbf{A}_x\mathbf{V}\mathbf{A}_y^\top)$, where $\mathbf{V} = \mathtt{vec}^{-1}(\mathbf{v})$.

$$\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{n_y}) \in \mathbb{R}^{n_x \times n_y} \iff \mathtt{vec}(\mathbf{V}) = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n_y} \end{pmatrix} \in \mathbb{R}^{n_x n_y}$$

Kronecker product and circulant matrices: Let $\mathbf{A}_x, \mathbf{A}_y$ be circulant then:

$$
\begin{aligned}
\mathbf{A} &= \mathbf{A}_x \otimes \mathbf{I}_y + \mathbf{I}_x \otimes \mathbf{A}_y \\
&= \mathbf{X}_x^{-1}\mathbf{D}_x\mathbf{X}_x \otimes \mathbf{X}_y^{-1}\mathbf{X}_y + \mathbf{X}_x^{-1}\mathbf{X}_x \otimes \mathbf{X}_y^{-1}\mathbf{D}_y\mathbf{X}_y \\
&= \left(\mathbf{X}_x^{-1} \otimes \mathbf{X}_y^{-1}\right)\left(\mathbf{D}_x \otimes \mathbf{I}_y + \mathbf{I}_x \otimes \mathbf{D}_y\right)\left(\mathbf{X}_x \otimes \mathbf{X}_y\right) \\
&= \mathbf{X}^{-1}\mathbf{D}\mathbf{X}
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{X} &= \mathbf{X}_x \otimes \mathbf{X}_y \qquad \text{(DFT matrix in 2D)} \\
\mathbf{D} &= \mathbf{D}_x \otimes \mathbf{I}_y + \mathbf{I}_x \otimes \mathbf{D}_y \qquad \text{(diagonal matrix)}
\end{aligned}
$$

where $\mathbf{X}_x$, $\mathbf{X}_y$ are the DFT matrices, $\mathbf{D}_x = \mathrm{diag}(\mathbf{X}_x\mathbf{a}_x)$, $\mathbf{D}_y = \mathrm{diag}(\mathbf{X}_y\mathbf{a}_y)$ and $\mathbf{a}_x = \mathbf{A}_x(:,1)$, $\mathbf{a}_y = \mathbf{A}_y(:,1)$, respectively.

Multiplying procedure: $\mathbf{A}^\dagger\mathbf{v} := \mathbf{X}^{-1}\left(\mathbf{D}^\dagger\left(\mathbf{X}\mathbf{v}\right)\right)$

$$
\begin{array}{ll}
0^\circ & \mathbf{d}_x := \mathtt{fft}(\mathbf{a}_x),\ \mathbf{d}_y := \mathtt{fft}(\mathbf{a}_y) \\
 & \mathbf{V} := \mathtt{vec}^{-1}(\mathbf{v}) \\
1^\circ & \mathbf{V} := \mathtt{fft}(\mathbf{V}) \\
2^\circ & \mathbf{V} := \mathtt{fft}(\mathbf{V}^\top)^\top \\
3^\circ & \mathbf{V} := \mathtt{vec}^{-1}(\mathbf{D}^\dagger\mathtt{vec}(\mathbf{V})) \\
4^\circ & \mathbf{V} := \mathtt{ifft}(\mathbf{V}) \\
5^\circ & \mathbf{V} := \mathtt{ifft}(\mathbf{V}^\top)^\top \\
 & \mathbf{A}^\dagger\mathbf{v} := \mathtt{vec}(\mathbf{V})
\end{array}
$$

Number of arithmetic operations :

$$\mathcal{O}(2n(\log_2 n_x + \log_2 n_y) + n) \approx \mathcal{O}(n \log_2 n), \qquad n = n_x n_y$$

Multiplying procedures: $\mathbf{N}\boldsymbol{\alpha}$, $\mathbf{N}^\top\mathbf{v}$, $\mathbf{M}\boldsymbol{\alpha}$, $\mathbf{M}^\top\mathbf{v}$ ... analogous

OUTLINE

Motivation: Fictitious domain method

Algorithm PSCM: Schur complement method + Null-space method

Inner solver: Projected BiCGSTAB

Preconditioning: Hierarchical multigrid

Singular matrices: Poisson-like solver based on circulants

Numerical experiments

CONCLUSIONS

- The method for solving non-symmetric sadddle-point systems with singular diagonal blocks was presented. It combines the Schur complement reduction with the null-space method.

- It can be understood as a generalization of the algebraic description of FETI DDM for non-symmetric and possibly indefinite cases.

- In connection with FDM, it presents the highly efficient solver for solving separable PDE problems. The fast implementation based on the Poisson-like solver is ”matrix free” as the stiffness matrix is not needed to be formed explicitly.

REFERENCES

J. Haslinger, T. Kozubek, R.K., G. Peichel (2007): *Projected Schur complement method for solving non-symmetric systems arising from a smooth fictitious domain approach*, acceptted in Num. Lin. Algebra Appl..

Farhat, C., Mandel, J., Roux, F., X. (1994): *Optimal convergence properties of the FETI domain decomposition method*, Comput. Meth. Appl. Mech. Engrg., 115, 365–385.

R. Glowinski, T. Pan, J. Periaux (1994): *A fictitious domain method for Dirichlet problem and applications.* Comput. Meth. Appl. Mech. Engrg. 111, 283–303.

R. K. (2005): *Complexity of an algorithm for solving saddle-point systems with singular blocks arising in wavelet-Galerkin discretizations*, Appl. Math. 50, 291–308.

H. A. Van der Vorst (1992): *BiCGSTAB: a fast and smoothly converging variant of BiCG for solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13, 631–644.

G. Meurant(1999): *Computer solution of large linear systems*, North Holland.

M. Benzi, G. H. Golub, J. Liesen(2005): *Numerical solution of saddle point systems*, Acta Numerica, pp. 1–137.