A smooth variant of the fictitious domain approach

J. Haslinger, T. Kozubek, R. Kučera

Department of Numerical Mathematics, Charles University, Prague
Department of Applied Mathematics, VŠB-TU, Ostrava
Department of Mathematics and Descriptive Geometry, VŠB-TU, Ostrava

1 Introduction

The classical fictitious domain method (FDM) enforces boundary conditions in PDE's by Lagrange multipliers defined on the boundary γ of the original domain ω [3]. Therefore the computed solution has a singularity on γ that can result in an intrinsic error. The basic idea of our modification consists in introducing new control variables (instead of Lagrange multipliers) defined on an auxiliary boundary Γ located outside of $\overline{\omega}$ [4]. In this approach, the singularity is moved away from $\overline{\omega}$ so that the computed solution is smoother in ω and the discretization error has a significantly higher rate of convergence in ω.

The respective finite element discretization leads typically to a non-symmetric saddle-point system

$$\begin{pmatrix} A & B_1^\top \\ B_2 & 0 \end{pmatrix} \begin{pmatrix} \hat{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1}$$

where an $(n\times n)$ diagonal block A is possibly singular and $(m\times n)$ off-diagonal blocks B_1, B_2 have full row-rank and they are highly sparse. Moreover, m is much smaller than n and the defect l of A, i.e., $l = n - rank\, A$, is much smaller than m. For solving such systems, it is convenient to use a method based on the Schur complement reduction [1]. If A is singular, the reduced system has again a saddle-point structure. Fortunately after applying orthogonal projectors, we obtain an equation in terms of λ only that can be efficiently solved by the projected variant of the BiCGSTAB algorithm [4]. This procedure generalizes ideas used in FETI domain decomposition methods [2], in which A is symmetric, positive semidefinite and $B_1 = B_2$.

2 Fictitious domain method

Let us consider a non-homogeneous Dirichlet boundary value problem:

$$-\Delta u = f \quad \text{in } \omega, \qquad u = g \quad \text{on } \gamma, \tag{2}$$

where $\omega \subset R^2$ is a bounded domain with the Lipschitz boundary γ, $f \in L^2_{loc}(R^2)$ and $g \in H^{1/2}(\gamma)$ are given data, or in a weak form:

$$\left.\begin{array}{l} \textit{Find } u \in H^1_0(\omega),\ u = g \textit{ on } \gamma \textit{ such that} \\ \displaystyle\int_\omega \nabla u \cdot \nabla v \, dx = \int_\omega f v \, dx \quad \forall v \in H^1_0(\omega). \end{array}\right\} \tag{3}$$

Let $\Xi \supset \overline{\omega}$ be another domain with the Lipschitz boundary Γ, $\mathrm{dist}(\Gamma, \gamma) = \delta$ for some $\delta > 0$ given. Finally, let $\Omega \supset \overline{\Xi}$ be a fictitious domain (a box, e.g.), see Fig. 1.

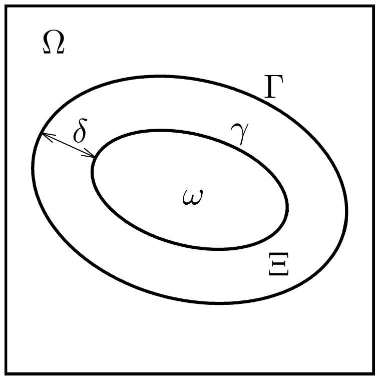

Figure 1: Geometry.

We define a problem:

$$\left.\begin{array}{l} \textit{Find } (\hat{u},\lambda) \in H_0^1(\Omega)\times H^{-1/2}(\Gamma) \textit{ such that} \\ \displaystyle\int_\Omega \nabla\hat{u}\cdot\nabla v\,dx = \int_\Omega fv\,dx + \langle\lambda, v\rangle_\Gamma \quad \forall v\in H_0^1(\Omega), \\ \langle\hat{u},\mu\rangle_\gamma = \langle g,\mu\rangle_\gamma \quad \forall\mu\in H^{-1/2}(\gamma). \end{array}\right\} \tag{4}$$

Suppose that (4) has a solution $(\hat{u},\lambda)$. It is easy to show that

$$\left.\begin{array}{rcl} -\Delta\hat{u} &=& f \quad \text{in } \Xi \text{ and } \Omega\setminus\overline{\Xi}, \\ \hat{u} &=& g \quad \text{on } \gamma, \\ \hat{u} &=& 0 \quad \text{on } \partial\Omega, \\ [\frac{\partial\hat{u}}{\partial\nu}]_\Gamma &=& \lambda \quad \text{on } \Gamma, \end{array}\right\} \tag{5}$$

where $[\ ]_\Gamma$ stands for the jump of the normal derivative $\frac{\partial\hat{u}}{\partial\nu}$ across Γ. In particular, $\hat{u}_{|\omega}$ solves (2).

Let us compare (4) with the classical fictitious domain formulation which uses boundary Lagrange multipliers on γ and which reads as follows:

$$\left.\begin{array}{l} \textit{Find } (\hat{u},\lambda) \in H_0^1(\Omega)\times H^{-1/2}(\gamma) \textit{ such that} \\ \displaystyle\int_\Omega \nabla\hat{u}\cdot\nabla v\,dx = \int_\Omega fv\,dx + \langle\lambda, v\rangle_\gamma \quad \forall v\in H_0^1(\Omega), \\ \langle\hat{u},\mu\rangle_\gamma = \langle g,\mu\rangle_\gamma \quad \forall\mu\in H^{-1/2}(\gamma). \end{array}\right\} \tag{6}$$

In (6), the component λ plays the role of Lagrange multipliers releasing the constraint $u = g$ on γ. On other hand, λ in (4) can be viewed to be a control variable enforcing $\hat{u}$ to match g on γ. If γ and Γ are smooth enough then $\hat{u}_{|\Xi}\in H^2(\Xi)$, $\hat{u}_{|\Omega\setminus\overline{\Xi}}\in H^2(\Omega\setminus\overline{\Xi})$ if $\hat{u}$ solves (4) and, similarly, $\hat{u}_{|\omega}\in H^2(\omega)$, $\hat{u}_{|\Omega\setminus\overline{\omega}}\in H^2(\Omega\setminus\overline{\omega})$ for $\hat{u}$ solving (6), while $\hat{u}\in H^{3/2-\epsilon}(\Omega)$ $\forall\epsilon>0$ in both cases. This means that the singularity of $\hat{u}$ is located either on Γ or on γ accordingly where generally a non-zero jump of $\frac{\partial\hat{u}}{\partial\nu}$ occurs. Since the singularity of $\hat{u}$ solving (4) is on Γ having a positive distance from γ, one can expect that the new variant (4) will increase the convergence rate of computed solutions in ω.

Now let us comment on a solvability of (4). This is closely related to a controlability type problem. To this end let us consider the following problem:

$$\left.\begin{array}{l} \textit{given } \lambda\in H^{-1/2}(\Gamma); \\ \textit{Find } u := u(\lambda)\in H_0^1(\Omega) \textit{ such that} \\ \displaystyle\int_\Omega \nabla u\cdot\nabla v\,dx = \int_\Omega fv\,dx + \langle\lambda, v\rangle_\Gamma \quad \forall v\in H_0^1(\Omega). \end{array}\right\} \tag{7}$$

Let $\Phi : H^{-1/2}(\Gamma) \mapsto H^{1/2}(\gamma)$ be a mapping defined by

$$\Phi(\lambda) = u(\lambda)_{|\gamma} \quad \forall \lambda \in H^{-1/2}(\Gamma),$$

where $u(\lambda)$ solves (7) and denote $\mathcal{V} = \Phi(H^{-1/2}(\Gamma))$. It can be shown that $\overline{\mathcal{V}} = H^{1/2}(\gamma)$ (for the proof see [4]). This means that if $g \in \mathcal{V}$ then (4) has a solution. If not, one can approximate g by $\widetilde{g} \in \mathcal{V}$ such that (4) with $\widetilde{g}$ replacing g has a solution again.

3 Algorithms

Let us return to the system (1) resulting from a finite element discretization of (4), where we use same notation for the discrete analogies of $\hat{u}$, λ, f and g.

Denote $\mathbb{N}(B|\mathbb{V})$ the null-space and $\mathbb{R}(B|\mathbb{V})$ the range-space of an $(m \times n)$ matrix B in a subspace $\mathbb{V} \subset R^n$. If $\mathbb{V} = R^n$, we simply write $\mathbb{N}(B) := \mathbb{N}(B|R^n)$ and $\mathbb{R}(B) := \mathbb{R}(B|R^n)$. The system (1) has a unique solution iff [4]

$$\mathbb{N}(A) \cap \mathbb{N}(B_2) = \{0\}, \tag{8}$$

$$\mathbb{R}(A|\mathbb{N}(B_2)) \cap \mathbb{R}(B_1^\top) = \{0\}. \tag{9}$$

Suppose that A is singular with the defect $l = \dim \mathbb{N}(A)$, $l \geq 1$ and consider $(n \times l)$ matrices N and M whose columns span the null-space $\mathbb{N}(A)$ and $\mathbb{N}(A^\top)$, respectively. Finally, denote by $A^\dagger$ a generalized inverse to A. In what follows we will consider an arbitrary but fixed selections of $A^\dagger$, N and M.

The *generalized Schur complement* of A in (1) is defined by

$$\mathcal{S} = \begin{pmatrix} -B_2 A^\dagger B_1^\top & B_2 N \\ M^\top B_1^\top & 0 \end{pmatrix}.$$

Notice that $\mathcal{S}$ is invertible provided that (8), (9) are satisfied. The following theorem describes the Schur complement reduction.

Theorem 1 [4] *Assume that both B_1, B_2 have full row-ranks and (8), (9) are satisfied. Then the second component λ of a solution to (1) is the first component of a solution to*

$$\begin{pmatrix} F & G_1^\top \\ G_2 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix}, \tag{10}$$

where $F := B_2 A^\dagger B_1^\top, G_1 := -N^\top B_2^\top, G_2 := -M^\top B_1^\top, d := B_2 A^\dagger f - g$ and $e := -M^\top f$. The first component u of a solution to (1) is given by the formulae

$$\hat{u} = A^\dagger (f - B_1^\top \lambda) + N\alpha.$$

Let us point out that (10) is formally the same saddle-point system as (1), but its size is considerably smaller. We will modify the new system (10) by two orthogonal projectors

$$P_1 := I - G_1^\top (G_1 G_1^\top)^{-1} G_1, \qquad P_2 := I - G_2^\top (G_2 G_2^\top)^{-1} G_2,$$

on $\mathbb{N}(G_1)$, $\mathbb{N}(G_2)$, respectively. Our algorithm is based on the following results.

Lemma 1 [4] *The linear operator* $P_1F : \mathbb{N}(G_2) \mapsto \mathbb{N}(G_1)$ *is invertible.*

Theorem 2 [4] *Let* $\lambda_{\mathbb{N}} \in \mathbb{N}(G_2)$, $\lambda_{\mathbb{R}} \in \mathbb{R}(G_2^\top)$. *Then* $\lambda = \lambda_{\mathbb{N}} + \lambda_{\mathbb{R}}$ *is the first component of a solution to (10) iff*

$$\lambda_{\mathbb{R}} = G_2^\top (G_2G_2^\top)^{-1} e$$

and

$$P_1F\lambda_{\mathbb{N}} = P_1(d - F\lambda_{\mathbb{R}}).$$

The second component α *is given by*

$$\alpha = (G_1G_1^\top)^{-1} G_1(d - F\lambda).$$

Let us summarize the previous results in the algorithm scheme. It turns out to be reasonable to form and store the $(l \times m)$ matrices G_1, G_2 and the $(l \times l)$ matrices $H_1 := (G_1G_1^\top)^{-1}$, $H_2 := (G_2G_2^\top)^{-1}$ because l is small. On the other hand, the $(m \times m)$ matrices F, P_1 and P_2 are not assembled explicitly.

Algorithm: Projected Schur Complement Method (PSCM)

Step 1.a: Assemble $G_1 = -N^\top B_2^\top$, $G_2 = -M^\top B_1^\top$, $d = B_2A^\dagger f - g$ and $e = -M^\top f$.
Step 1.b: Assemble $H_1 = (G_1G_1^\top)^{-1}$ and $H_2 = (G_2G_2^\top)^{-1}$.
Step 1.c: Assemble $\lambda_{\mathbb{R}} = G_2^\top H_2 e$.
Step 1.d: Assemble $\tilde{d} = P_1(d - F\lambda_{\mathbb{R}})$.
Step 1.e: Solve the equation $P_1F\lambda_{\mathbb{N}} = \tilde{d}$ on $\mathbb{N}(G_2)$.
Step 1.f: Compute $\lambda = \lambda_{\mathbb{N}} + \lambda_{\mathbb{R}}$.
Step 2: Compute $\alpha = H_1G_1(d - F\lambda)$.
Step 3: Compute $\hat{u} = A^\dagger(f - B_1^\top\lambda) + N\alpha$.

The heart of the algorithm consists in Step 1.e. Its solution can be computed by a *projected* Krylov subspace method. The projected BiCGSTAB algorithm [4] can be derived from the non-projected one [6] by choosing an initial iterate $\lambda_{\mathbb{N}}^0$ on $\mathbb{N}(G_2)$, projecting the initial residual in $\mathbb{N}(G_2)$ and replacing the operator P_1F by its projected version P_2P_1F. We will denote applications of this algorithm by

$$\text{ProjBiCGSTAB}[\epsilon, \lambda_{\mathbb{N}}^0, F, P_1, P_2, \widetilde{d}] \to \lambda_{\mathbb{N}}$$

and we will assume that its iterations are terminated whenever the norm of the k-th residual is smaller than ϵ.

As the fictitious domain Ω has a simple geometry, it is easy to define a multilevel family of nested partitions with stepsizes h_j, $0 \le j \le J$, so that $h_{j+1} < h_j$ (e.g., $h_{j+1} = h_j/2$). In order to accelerate BiCGSTAB iterations on the finest J-th level, one can apply the hierarchical multigrid scheme, which is formulated below. Note that upper indices denote the affiliation to the j-th level.

The computation starts on the coarsest level, $j = 0$, with the first iterate $\lambda_{\mathbb{N}}^{0,(0)}$ arbitrarily chosen in $\mathbb{N}(G_2^{(0)})$ (e.g., $\lambda_{\mathbb{N}}^{0,(0)} = 0$). The first iterate on each subsequent level is determined as the prolongated and projected result from the nearest lower level. The terminating tolerance ϵ on the j-th level is set proportionally to an expected discretization error that is $\epsilon := ch_j^p$, where p is an expected convergence rate (in the $L^2(\omega)$-norm) and c is a control parametr. The result obtained with such ϵ can be viewed as an inexact solution of (1) with the same convergence rate as the exact one.

Algorithm: Hierarchical Multigrid Scheme

Initialize: Let $\lambda_{\mathbb{N}}^{0,(0)} \in \mathbb{N}(G_2^{(0)})$ be given.

ProjBiCGSTAB$[ch_0^p, \lambda_{\mathbb{N}}^{0,(0)}, F^{(0)}, P_1^{(0)}, P_2^{(0)}, \widetilde{d}^{(0)}] \to \lambda_{\mathbb{N}}^{(0)}$.

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

prolongate $\lambda_{\mathbb{N}}^{(j-1)} \to \widetilde{\lambda}_{\mathbb{N}}^{0,(j)}$,

project $\widetilde{\lambda}_{\mathbb{N}}^{0,(j)} \to \lambda_{\mathbb{N}}^{0,(j)} := P_2^{(j)} \widetilde{\lambda}_{\mathbb{N}}^{0,(j)}$,

ProjBiCGSTAB$[ch_j^p, \lambda_{\mathbb{N}}^{0,(j)}, F^{(j)}, P_1^{(j)}, P_2^{(j)}, \widetilde{d}^{(j)}] \to \lambda_{\mathbb{N}}^{(j)}$,

end.

Return: $\lambda_{\mathbb{N}} := \lambda_{\mathbb{N}}^{(J)}$.

4 Numerical experiments

Let ω be the ellipse, $\omega \equiv \{(x,y) \in R^2 |\ (x-0.5)^2/0.4^2 + (y-0.5)^2/0.2^2 < 1\}$, and the fictitious domain $\Omega = (0,1) \times (0,1)$. We will assume that the right hand-sides f and g in (2) are chosen appropriately to the exact solution $\hat{u}_{ex}(x,y) = 100\left((x-0.5)^3 - (y-0.5)^3\right) - x^2$; see Figs. 2-4.

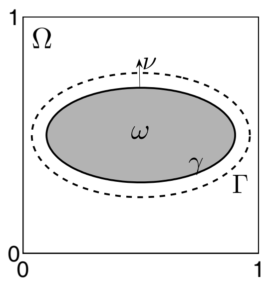

Figure 2: Geometry of ω.

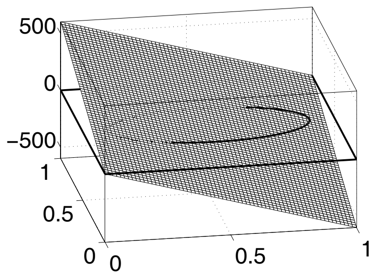

Figure 3: Right hand side f.

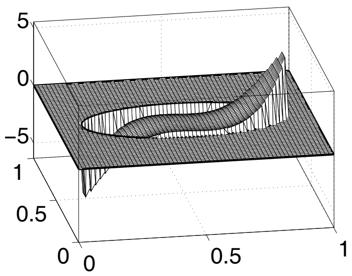

Figure 4: Ex. solution $\hat{u}_{ex|_\omega}$.

The space $H_0^1(\Omega)$ in (4) is replaced by $H_{per}^1(\Omega)$ enabling us to use the Fourier direct method [5] to compute actions of $A^\dagger$, where A is the positive semidefinite discrete Laplacian resulting from the discretization of $H_{per}^1(\Omega)$ by piecewise bilinear functions defined on a rectangulation of Ω with a stepsize h. The spaces $H^{-1/2}(\Gamma)$ and $H^{-1/2}(\gamma)$ are approximated by piecewise constant functions defined on partitions of polygonal approximations of Γ and γ, respectively.

In Table 1, we report the errors of the approximate solutions with respect to the stepsize h in the $H^1(\omega)$-norm together with the number of BiCGSTAB iterations. We compare the classical FDM (6) and our modification (4), in which the auxiliary boundary Γ arises by shifting γ in the direction of the outward normal vector ν with $\delta = 8h$. From the computed errors, we determine the convergence rates (the last row of the table) that are considerably higher for our modification of the FDM.

Fig. 5 illustrates a smoothing effect of δ. If the auxiliary boundary Γ is shifted far enough from the original γ, the smoothness of the computed solution increases that results in smaller discretization errors. On the other hand, Fig. 6 shows that the condition number of $P_1 F$ (on $\mathbb{N}(G_2)$) increases exponentially with respect to δ.

Table 1: Numerical results; $\epsilon = h^2\|\widetilde{d}\|$, $\delta = 8h$.

	Classical		Modified		Modified+Multigrid	
Step h	Iters.	$\mathrm{Err}_{H^1(\omega)}$	Iters.	$\mathrm{Err}_{H^1(\omega)}$	Iters.	$\mathrm{Err}_{H^1(\omega)}$
1/128	8	1.9647e+0	13	1.6878e-2	11	1.8988e-2
1/256	9	1.2884e+0	25	7.7891e-3	13	7.6303e-3
1/512	12	8.6517e-1	40	4.0160e-3	19	3.8638e-3
1/1024	18	6.0510e-1	58	1.9098e-3	21	1.7758e-3
1/2048	25	4.4015e-1	86	9.9299e-4	31	9.8213e-4
Conv. rates:		0.54		1.02		1.07

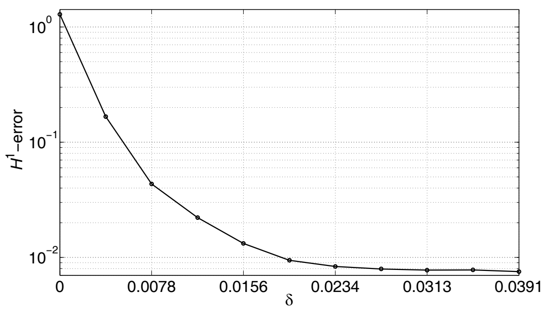

Figure 5: $H^1(\omega)$-error sensitivity on δ.

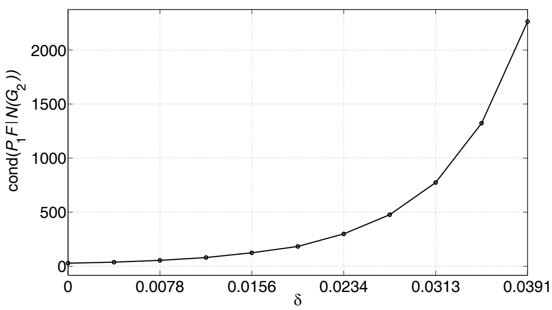

Figure 6: $cond(P_1F|\mathbb{N}(G_2))$ sensitivity on δ.

Acknowledgement: This work has been supported by the National Program of Research "Information Society" under project 1ET400300415, by the grant IAA1075402 of the Grant Agency of the Czech Academy of Sciences and by the Research Project MSM6198910027 of the Czech Ministry of Education.

References

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

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

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

[4] J. Haslinger, T. Kozubek, R. Kučera, G. Peichl: *Projected Schur complement method for solving non-symmetric saddle-point systems arising from fictitious domain approaches.* Submitted to Num. Lin. Alg. Appl. (2007).

[5] R. Kučera: *Complexity of an algorithm for solving saddle-point systems with singular blocks arising in wavelet-Galerkin discretizations.* Appl. Math. 50(2005), 3, 291–308.

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