



**HAL**  
open science

# Adaptive Domain Decomposition method for Saddle Point problem in Matrix Form

F. Nataf

► **To cite this version:**

F. Nataf. Adaptive Domain Decomposition method for Saddle Point problem in Matrix Form. 2019.  
hal-02343808v1

**HAL Id: hal-02343808**

**<https://hal.science/hal-02343808v1>**

Preprint submitted on 4 Nov 2019 (v1), last revised 16 Nov 2022 (v6)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive Domain Decomposition method for Saddle Point problem in Matrix Form

F. Nataf<sup>1</sup>

<sup>1</sup>Sorbonne Université, CNRS, Université de Paris, Inria Equipe Alpines, Laboratoire  
Jacques-Louis Lions, F-75005 Paris, France, frederic.nataf@sorbonne-universite.fr

November 4, 2019

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Schur complement preconditioning</b>	<b>2</b>
<b>3</b>	<b>Preconditioning of <math>M_S</math></b>	<b>4</b>
3.1	Equivalent preconditioner of $S_1$ . . . . .	5
3.1.1	One-level DD . . . . .	5
	Surjectivity of $\mathcal{R}$ . . . . .	6
	Continuity of $\mathcal{R}$ . . . . .	6
	Stable decomposition . . . . .	6
3.1.2	Two-level DD . . . . .	6
3.2	Preconditioner for $A_0$ . . . . .	7
3.2.1	Scalability of the computation of $M_{A_0}$ . . . . .	7
<b>4</b>	<b>Recap</b>	<b>8</b>
4.1	Setup for the Schur complement preconditioner $M_S$ . . . . .	8
4.2	Preconditioning of the Schur complement $S$ by $M_S^{-1}$ . . . . .	9
4.3	DD solver for the saddle point system . . . . .	9
<b>5</b>	<b>Variants</b>	<b>10</b>

## Abstract

We introduce an adaptive domain decomposition (DD) method for solving saddle point problems defined as a block two by two matrix. The algorithm does not require any knowledge of the constrained space. We assume that all sub matrices are sparse and that the diagonal blocks are the sum of positive semi definite matrices. The latter assumption enables the design of adaptive coarse space for DD methods, see [5].

# 1 Introduction

Solving saddle point problems with parallel algorithms is very important for many branches of scientific computing: fluid (see e.g. [13]) and solid mechanics, computational electromagnetism, inverse problem and optimization.

We are interested in domain decomposition (DD) methods since they are naturally well-fitted to modern parallel architectures. For specific systems of partial differential equations with a saddle point formulation, efficient DD methods have been designed, see e.g. [16, 12, 17] and [20] references therein.

Here as in [14, 2, 4], we consider the problem in the form of a two by two block matrix. Let  $m$  and  $n$  be two integers with  $m < n$ . Let  $A$   $n \times n$  SPD matrix and  $B$  be a sparse  $m \times n$  full rank matrix of constraints and  $C$  a  $m \times m$  non negative matrix (in particular,  $C = 0$  is allowed), we consider the following saddle point matrix:

$$\mathcal{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}. \quad (1)$$

When the kernel of matrix  $B$  is known, very efficient multigrid methods have been designed in the context of finite element methods, see e.g. [10, 9, 1, 18, 6].

Here we do not assume any knowledge on the kernel of matrix  $B$ . But in order to build a scalable method, we assume that all three matrices are sparse and that  $A$  and  $C$  are the sum of positive semi definite matrices. The latter assumption enables the design of adaptive coarse space for DD methods, see [5].

## 2 Schur complement preconditioning

The sparse  $n \times n$  SPD matrix  $A$  is preconditioned by a two-level Schwarz type DD method :

$$M_A^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^N R_i^T (R_i A R_i^T)^{-1} R_i, \quad (2)$$

where  $R_0$  is full rank  $\dim(V_0) \times n$  where  $V_0$  denotes the space spanned by the columns of  $R_0^T$ . The following assumptions are crucial to ensure the final method is scalable:

### Assumption 2.1 (dimension and structure of the coarse space)

- *The coarse space dimension,  $\dim(V_0)$ , is  $O(N)$  typically 10-20 times  $N$ .*
- *The coarse space is made of extensions by zero of local vectors.*

Using the GenEO method [19], it is possible to fix in advance two constants  $0 < \lambda_m < 1 < \lambda_M$  and then build a coarse space  $V_0$  such that  $A^{-1}$  is spectrally equivalent to  $M_A^{-1}$ :

$$\frac{1}{\lambda_M} M_A^{-1} \leq A^{-1} \leq \frac{1}{\lambda_m} M_A^{-1},$$

The coarse space  $V_0$  is made extensions by zero of local generalized eigenvalue problems and its dimension is typically proportional to the number of subdomains. This corresponds to Assumption 2.1.

Our aim is first to precondition the Schur complement  $-S$  of matrix  $\mathcal{A}$  (eq. (1)) where

$$S := C + B A^{-1} B^T, \quad (3)$$

via the spectrally equivalent preconditioning of the spectrally equivalent Schur complement  $M_S$ ,

$$M_S := C + B M_A^{-1} B^T. \quad (4)$$

This is done in § 3. Then in § 4.3 we will use  $M_S$  to precondition the saddle point matrix  $\mathcal{A}$ .

Note that  $M_S$  is by definition a sum of  $N + 2$  positive semi definite matrices

$$M_S := B R_0^T (R_0 A R_0^T)^{-1} R_0 B^T + C + \sum_{i=1}^N B R_i^T (R_i A R_i^T)^{-1} R_i B^T. \quad (5)$$

Since  $B$  is a sparse matrix, it is interesting to introduce, for all  $0 \leq i \leq N$ ,  $\tilde{R}_i$  the restriction operator on the support of  $\mathfrak{S}(B R_i^T)$  so that  $\tilde{R}_i^T \tilde{R}_i B R_i^T = B R_i^T$ . Then by defining for  $0 \leq i \leq N$ ,

$$\tilde{B}_i := \tilde{R}_i B R_i^T,$$

the operator  $M_S$  is rewritten as

$$M_S := \tilde{R}_0^T \tilde{B}_0 (R_0 A R_0^T)^{-1} \tilde{B}_0^T \tilde{R}_0 + C + \sum_{i=1}^N \tilde{R}_i^T \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \tilde{R}_i. \quad (6)$$

We consider a partition of unity on  $H := \mathbb{R}^m$  defined with local diagonal matrices  $(\tilde{D}_i)_{1 \leq i \leq N} \in \mathbb{R}^{\dim(\text{Im}(B R_i^T)) \times \dim(\text{Im}(B R_i^T))}$ :

$$\sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \tilde{R}_i = I_H.$$

**Remark 2.1** *This partition of unity exists since*

$$B = \sum_{i=1}^N B R_i^T \tilde{D}_i R_i = \sum_{i=1}^N \tilde{R}_i^T \tilde{R}_i (B R_i^T \tilde{D}_i R_i)$$

*is full rank.*

### 3 Preconditioning of $M_S$

We make the following assumption

**Assumption 3.1** *There exist symmetric positive semidefinite matrices  $(\tilde{C}_i)_{1 \leq i \leq N}$  such that*

$$C = \sum_{i=1}^N \tilde{R}_i^T \tilde{C}_i \tilde{R}_i.$$

This assumption is not so restrictive. Indeed, if  $C = 0$ , it is automatically satisfied, this corresponds to a minimization problem with constraints enforced exactly without penalization nor relaxation. Moreover, we have:

**Lemma 3.1** *If  $C$  is a diagonal matrix, Assumption 3.1 is satisfied.*

**Proof** If  $C$  is a diagonal matrix, it suffices to take

$$\tilde{C}_i := \tilde{R}_i C \tilde{R}_i^T \tilde{D}_i,$$

which is a diagonal non negative matrix. Indeed, we have then:

$$C \mathbf{P} = \sum_{i=1}^N C \tilde{R}_i^T \tilde{D}_i \tilde{R}_i \mathbf{P} = \sum_{i=1}^N \tilde{R}_i^T (\tilde{R}_i C \tilde{R}_i^T \tilde{D}_i) \tilde{R}_i \mathbf{P}.$$

■

Then, the operator  $M_S$  is the sum of a non local but low rank matrix  $S_0$ :

$$S_0 := \tilde{R}_0^T \tilde{B}_0 (R_0 A R_0^T)^{-1} \tilde{B}_0^T \tilde{R}_0,$$

and of  $S_1$  which is a sum of  $N$  local positive semi definite matrices:

$$S_1 := \sum_{i=1}^N \tilde{R}_i^T (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \tilde{R}_i,$$

that is

$$M_S = S_0 + S_1.$$

Note that we may assume that  $S_1$  is invertible whereas it does not make sense for  $S_0$ . By factorizing  $R_0 A R_0^T = L_0 L_0^T \in \mathbb{R}^{\dim(V_0) \times \dim(V_0)}$ , we have a Cholevsky factorization of  $S_0$ :

$$S_0 = \tilde{R}_0^T \tilde{B}_0 L_0^{T-1} L_0^{-1} \tilde{B}_0^T \tilde{R}_0.$$

By Sherman-Morrison's technique, solving  $M_S \mathbf{P} = \mathbf{G}$  for some  $\mathbf{G} \in \mathbb{R}^m$ , amounts to solving

$$\begin{pmatrix} S_1 & \tilde{R}_0^T \tilde{B}_0 L_0^{T-1} \\ L_0^{-1} \tilde{B}_0^T \tilde{R}_0 & -I \end{pmatrix} \begin{pmatrix} \mathbf{P} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{G} \\ \mathbf{0} \end{pmatrix},$$

with  $\mathbf{y} := L_0^{-1} \tilde{B}_0^T \tilde{R}_0 \mathbf{P} \in \mathbb{R}^{\dim(V_0)}$ . This is equivalent to solving its Schur complement:

$$A_0 \mathbf{y} := (I + L_0^{-1} \tilde{B}_0^T \tilde{R}_0 S_1^{-1} \tilde{R}_0^T \tilde{B}_0 L_0^{T-1}) \mathbf{y} = L_0^{-1} \tilde{B}_0^T \tilde{R}_0 S_1^{-1} \mathbf{G} \in \mathbb{R}^{\dim(V_0)}. \quad (7)$$

In order to solve this equation, we consider two options: a direct solver or a Krylov method.

As for the first method, it requires the computation of the entries of the matrix  $A_0 \in \mathbb{R}^{\dim(V_0) \times \dim(V_0)}$  whose dimension is small. But computing its entries amounts to solve  $\dim(V_0)$  linear systems with  $S_1$ . Even if each solve is made scalable by a suitable parallel preconditioner  $M_{S_1}$  defined below in § 3.1, the whole thing has a  $O(N)$  complexity and is thus not scalable.

The second option, using a Krylov method, is interesting since the matrix-vector product with  $A_0$  is scalable using the preconditioner  $M_{S_1}$  when solving a linear system with operator  $S_1$ . Now when solving (7) with a unpreconditioned Krylov method, convergence is in at most  $\dim(V_0)$  iterations which is still not scalable. In order to build a spectrally equivalent preconditioner  $M_{A_0}^{-1}$  to  $A_0$ , we use again the preconditioner  $M_{S_1}^{-1}$  of  $S_1$ . and then use it in a preconditioned conjugate gradient algorithm to solve eq. (7), see § 3.2.

We consider next the construction of a preconditioner  $M_{S_1}^{-1}$  to  $S_1$ .

### 3.1 Equivalent preconditioner of $S_1$

#### 3.1.1 One-level DD

As in [5] chapter 7, we begin with a one-level Neumann-Neumann type DD method defined in terms of the Fictitious Space Lemma (FSL) [15, 7]. This study will be the basis for constructing the two-level preconditioner. Recall

$$H := \mathbb{R}^m$$

and let  $a$  be the following bilinear form:

$$a : H \times H \rightarrow \mathbb{R} \quad a(\mathbf{P}, \mathbf{Q}) := (S_1 \mathbf{P}, \mathbf{Q}).$$

Let

$$H_D := \prod_{i=1}^N \mathbb{R}^{\text{rank}(\tilde{B}_i)},$$

and  $b$  be the following bilinear form:

$$b : H_D \times H_D \rightarrow \mathbb{R} \quad b(\mathcal{P}, \mathcal{Q}) := \sum_{i=1}^N ((\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_i, \mathbf{Q}_i).$$

We define  $\mathcal{R}$ :

$$\mathcal{R} : \quad H_D \rightarrow \quad H \\ (\mathbf{P}_i)_{1 \leq i \leq N} \mapsto \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i,$$

We now check the three assumptions of the FSL.

**Surjectivity of  $\mathcal{R}$**  For any  $\mathbf{P} \in H$ , we have:

$$\mathbf{P} = \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \tilde{R}_i \mathbf{P},$$

so that

$$\mathbf{P} = \mathcal{R}((\tilde{R}_i \mathbf{P})_{1 \leq i \leq N}). \quad (8)$$

**Continuity of  $\mathcal{R}$**  On one hand, we have using  $k_0$  the number of neighbours of a subdomain plus one,  $k_0 := \max_{1 \leq i \leq N} \#\mathcal{O}(i)$  where  $\mathcal{O}(i) := \{1 \leq j \leq N \mid \tilde{R}_i \tilde{D}_i S_1 \tilde{D}_j \tilde{R}_j^T \neq 0\}$ :

$$\begin{aligned} a(\mathcal{R}(\mathcal{P}), \mathcal{R}(\mathcal{P})) &= \|(\sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i)\|_a^2 \leq k_0 \sum_{i=1}^N \|\tilde{R}_i^T \tilde{D}_i \mathbf{P}_i\|_a^2 \\ &= k_0 \left( \left( \sum_{j \in \mathcal{O}(i)} \tilde{R}_j^T (\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T) \tilde{R}_j \right) \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i, \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i \right). \end{aligned}$$

On the other hand, we have by definition:

$$b(\mathcal{P}, \mathcal{P}) = \sum_{i=1}^N \left( (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_i, \mathbf{P}_i \right).$$

We can take:

$$c_R := \max_{1 \leq i \leq N} \max_{P_i \in \mathbb{R}^{\text{rank}(\tilde{B}_i)}} \frac{(\sum_{j \in \mathcal{O}(i)} \tilde{R}_j \tilde{R}_j^T (\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T) \tilde{R}_j \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i, \tilde{D}_i \mathbf{P}_i)}{((\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_i, \mathbf{P}_i)}.$$

**Stable decomposition** Let  $\mathbf{P} \in H$ , we start from its decomposition (8) and estimate its  $b$ -norm

$$b(\mathcal{P}, \mathcal{P}) = \sum_{i=1}^N \left( (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \tilde{R}_i \mathbf{P}, \tilde{R}_i \mathbf{P} \right) = a(\mathbf{P}, \mathbf{P}),$$

so that we can take  $c_T = 1$ .

### 3.1.2 Two-level DD

In order to control the value of  $c_R$  defined above, we introduce a two-level preconditioner exactly as in [5], § 7.8.3, page 197. In our case, the generalized eigenvalue value problem in each subdomain  $1 \leq i \leq N$  to be solved to build the coarse space reads:

$$\begin{aligned} \tilde{D}_i \left( \sum_{j \in \mathcal{O}(i)} \tilde{R}_j \tilde{R}_j^T (\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T) \tilde{R}_j \tilde{R}_i^T \right) \tilde{D}_i \mathbf{P}_{ik} & \\ = \lambda_{ik} (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_{ik}. & \end{aligned} \quad (9)$$

It can be solved in  $O(1)$  communications. The coarse space is defined as follows. Let  $\tau_{S_1}$  be a user-defined threshold; for each subdomain  $1 \leq i \leq N$ , we introduce a subspace  $W_i \subset \mathbb{R}^{\text{rank}(\tilde{B}_i)}$ :

$$\tilde{W}_i := \text{Span}\{\mathbf{P}_{ik} \mid \lambda_{ik} > \tau_{S_1}\}. \quad (10)$$

Then the coarse space  $\tilde{W}_0$  is defined by

$$\tilde{W}_0 := \bigoplus_{1 \leq i \leq N} \tilde{R}_i^T \tilde{D}_i \tilde{W}_i.$$

Let  $Z_{S_1}$  be a rectangular matrix whose columns span the coarse space  $\tilde{W}_0$ . Let  $\tilde{P}_0$  be the  $S_1$  orthogonal projection from  $\mathbb{R}^m$  on  $\tilde{W}_0$  whose formula is

$$\tilde{P}_0 = Z_{S_1} (Z_{S_1}^T S_1 Z_{S_1})^{-1} Z_{S_1}^T S_1. \quad (11)$$

Finally, the preconditioner for  $S_1$  reads

$$\begin{aligned} M_{S_1}^{-1} &:= Z_{S_1} (Z_{S_1}^T S_1 Z_{S_1})^{-1} Z_{S_1}^T + (I - \tilde{P}_0) \\ &\times \left( \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)^\dagger \tilde{D}_i \tilde{R}_i \right) (I - \tilde{P}_0^T). \end{aligned} \quad (12)$$

Recall that we have for  $\alpha := \max(1, \frac{k_0}{\tau_{S_1}})$ :

$$\frac{1}{\alpha} M_{S_1}^{-1} \leq S^{-1} \leq M_{S_1}^{-1}.$$

**Remark 3.1** *Solving a linear system with the Schur complement*

$$(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_i = \mathbf{G}_i,$$

amounts to solving an augmented sparse system of the form

$$- \begin{pmatrix} R_i A R_i^T & \tilde{B}_i^T \\ \tilde{B}_i & -\tilde{C}_i \end{pmatrix} \begin{pmatrix} \mathbf{U}_i \\ \mathbf{P}_i \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{G}_i \end{pmatrix}.$$

### 3.2 Preconditioner for $A_0$

Let us define

$$M_{A_0} := (I + L_0^{-1} \tilde{B}_0^T \tilde{R}_0 M_{S_1}^{-1} \tilde{R}_0^T \tilde{B}_0 L_0^{T-1}), \quad (13)$$

which is spectrally equivalent to matrix  $A_0$ :

$$\frac{1}{\alpha} M_{A_0} \leq A_0 \leq M_{A_0}.$$

#### 3.2.1 Scalability of the computation of $M_{A_0}$

Compared to  $A_0$ , the computation of the entries of  $M_{A_0}$  is scalable since  $\tilde{R}_0^T \tilde{B}_0 = B^T R_0$  has a sparse structure, see Assumption 2.1. Consider for instance the term corresponding to the coarse space in the formula for  $M_{S_1}^{-1}$  (see eq. (12)):

$$L_0^{-1} \tilde{B}_0^T \tilde{R}_0 Z_{S_1} (Z_{S_1}^T S_1 Z_{S_1})^{-1} Z_{S_1}^T \tilde{R}_0^T \tilde{B}_0 L_0^{T-1}.$$



Matrix  $Z_{S_1}^T \tilde{R}_0^T \in \mathbb{R}^{\dim(\tilde{W}_0) \times \dim(V_0)}$  is the product of two DD coarse spaces so that its computation is  $O(1)$ .

For the other computations, we need extra assumptions 3.2 and 3.3. Indeed, let's look at the sum over the subdomains in (12) and we isolate one subdomain  $1 \leq i \leq N$ :

$$L_0^{-1} \tilde{B}_0^T \tilde{R}_0 (I - \tilde{P}_0) \tilde{R}_i^T \tilde{D}_i (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)^\dagger \tilde{D}_i \tilde{R}_i (I - \tilde{P}_0^T) \tilde{R}_0^T \tilde{B}_0 L_0^{T-1}.$$

This can be decomposed into a four term sum, the first term being:

$$L_0^{-1} \tilde{B}_0^T \tilde{R}_0 \tilde{R}_i^T \tilde{D}_i (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)^\dagger \tilde{D}_i \tilde{R}_i \tilde{R}_0^T \tilde{B}_0 L_0^{T-1},$$

**Assumption 3.2** We assume that, for all  $1 \leq i \leq N$ , the local matrix

$$\tilde{D}_i \tilde{R}_i \tilde{R}_0^T \tilde{B}_0 = \tilde{D}_i \tilde{R}_i B^T R_0$$

has  $O(1)$  non zero columns.

The second term being:

$$-L_0^{-1} \tilde{B}_0^T \tilde{R}_0 \tilde{R}_i^T \tilde{D}_i (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)^\dagger \tilde{D}_i \tilde{R}_i \tilde{P}_0^T \tilde{R}_0^T \tilde{B}_0 L_0^{T-1}.$$

We focus on the term

$$\tilde{D}_i \tilde{R}_i \tilde{P}_0^T \tilde{R}_0^T \tilde{B}_0 = \tilde{D}_i \tilde{R}_i S_1 Z_{S_1} (Z_{S_1}^T S_1 Z_{S_1})^{-1} Z_{S_1}^T B^T R_0.$$

**Assumption 3.3** We assume that, for all  $1 \leq i \leq N$ , the local matrix  $\tilde{D}_i \tilde{R}_i S_1 Z_{S_1}$  has  $O(1)$  non zero columns

Using this assumption, the computation of second term is scalable. The two other terms can also be computed in a scalable way.

Note that, the resulting matrix  $M_{A_0}$  being of small size, it can be factorized by a direct method.

## 4 Recap

### 4.1 Setup for the Schur complement preconditioner $M_S$

We have a setup phase which is composed of three stages, the first two ones can be done concurrently:

1. Build the two-level preconditioner  $M_A^{-1}$  for  $A$ , see eq. (2),
2. Build the two-level preconditioner  $M_{S_1}^{-1}$  for  $S_1$ , see eq. (12),
3. Compute the entries of matrix  $M_{A_0}$ , see eq. (13) and factorize it.

## 4.2 Preconditioning of the Schur complement $S$ by $M_S^{-1}$

Applying preconditioner  $M_S^{-1}$  is performed by solving  $M_S \mathbf{P} = \mathbf{G} \in \mathbb{R}^m$  with several Krylov solves with spectrally equivalent preconditioners :

---

### Algorithm 1 $M_S^{-1}$ matvec product

---

INPUT:  $\mathbf{G} \in \mathbb{R}^m$

1. Solve  $S_1 \mathbf{G}' = \mathbf{G}$  by using the two-level preconditioner  $M_{S_1}^{-1}$ .

Compute the right handside  $L_0^{-1} \tilde{B}_0^T \tilde{R}_0 \mathbf{G}'$  of eq. (7).

2. Solve eq. (7) for  $\mathbf{y}$  using the inverse of  $M_{A_0}$  as preconditioner and the matrix vector product with  $A_0$  is made using a Krylov solver for  $S_1^{-1}$  preconditioned by  $M_{S_1}^{-1}$ .

3. Solve  $S_1 \mathbf{P} = \mathbf{G} - \tilde{R}_0^T \tilde{B}_0 L_0^{T-1} \mathbf{y}$  for  $\mathbf{P}$  using the two-level preconditioner  $M_{S_1}^{-1}$

OUTPUT:  $\mathbf{P} = M_S^{-1} \mathbf{G}$

---

## 4.3 DD solver for the saddle point system

We now consider the solving of the saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_U \\ \mathbf{F}_P \end{pmatrix}. \quad (14)$$

The following three factor factorization, see e.g. [3]:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & -(C + BA^{-1}B^T) \end{pmatrix} \begin{pmatrix} I & A^{-1}B^T \\ 0 & I \end{pmatrix},$$

shows that solving eq. (14) can be performed by solving sequentially three linear systems, two with  $A$  and one with  $C + BA^{-1}B^T$ . This leads to the following algorithm:

---

### Algorithm 2 DD saddle point solver

---

INPUT:  $\begin{pmatrix} \mathbf{F}_U \\ \mathbf{F}_P \end{pmatrix} \in \mathbb{R}^{n+m}$

1. Solve  $A \mathbf{G}_U = \mathbf{F}_U$  by a PCG with  $M_A^{-1}$  as a preconditioner

Compute  $\mathbf{G}_P := \mathbf{F}_P - B \mathbf{G}_U$

2. Solve  $(C + BA^{-1}B^T) \mathbf{P} = -\mathbf{G}_P$  by a PCG with  $M_S^{-1}$  as a preconditioner, see Algorithm 1.

Compute  $\mathbf{G}_U := \mathbf{F}_U - B^T \mathbf{P}$

3. Solve  $A \mathbf{U} = \mathbf{G}_U$  by a PCG with  $M_A^{-1}$  as a preconditioner

OUTPUT:  $\begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix}$  the solution to (14).

---

## 5 Variants

In § (2), we start with a two level additive Schwarz method (ASM) eq. (2) as a preconditioner for matrix  $A$  in (1). Another possibility is to start from a balancing Neumann-Neumann (BNN) or more generally a SORAS [8] or BDD-H [11] type method:

$$M_{ASORAS}^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^N R_i^T A_i^R ob^{-1} R_i, \quad (15)$$

where for each subdomain  $1 \leq i \leq N$ ,  $A_i^R ob$  is a local Neumann matrix (BNN algorithm) or an arbitrary invertible matrix. In order to build the preconditioners for matrix  $S$  and then  $\mathcal{A}$ , it is sufficient to replace matrices  $R_i A R_i^T$  with  $A_i^R ob$  in the above sections.

## References

- [1] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Multigrid in  $h$  (div) and  $h$  (curl). *Numerische Mathematik*, 85(2):197–217, Apr 2000.
- [2] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.
- [3] Michele Benzi and Andrew J. Wathen. Some preconditioning techniques for saddle point problems. In *Model order reduction: theory, research aspects and applications*, volume 13 of *Math. Ind.*, pages 195–211. Springer, Berlin, 2008.
- [4] Eric de Sturler and Jörg Liesen. Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. I. Theory. *SIAM J. Sci. Comput.*, 26(5):1598–1619, 2005.
- [5] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. *An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation*. SIAM, 2015.
- [6] Patrick E. Farrell, Lawrence Mitchell, and Florian Wechsung. An Augmented Lagrangian Preconditioner for the 3D Stationary Incompressible Navier–Stokes Equations at High Reynolds Number. *SIAM J. Sci. Comput.*, 41(5):A3073–A3096, 2019.
- [7] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numer. Math.*, 70(2):163–180, 1995.
- [8] R. Haferssas, P. Jolivet, and F. Nataf. An additive Schwarz method type theory for Lions’s algorithm and a symmetrized optimized restricted

- additive Schwarz method. *SIAM J. Sci. Comput.*, 39(4):A1345–A1365, 2017.
- [9] R. Hiptmair. Multigrid method for Maxwell’s equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1998.
- [10] Ralf Hiptmair. Multigrid method for  $h$  (div) in three dimensions. *Electron. Trans. Numer. Anal.*, 6(1):133–152, 1997.
- [11] Jung-Han Kimn and Marcus Sarkis. Restricted overlapping balancing domain decomposition methods and restricted coarse problems for the Helmholtz problem. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1507–1514, 2007.
- [12] Axel Klawonn. An optimal preconditioner for a class of saddle point problems with a penalty term. *SIAM Journal on Scientific Computing*, 19(2):540–552, 1998.
- [13] A. El maliki, M. Fortin, J. Deteix, and A. Fortin. Preconditioned iteration for saddle-point systems with bound constraints arising in contact problems. *Computer Methods in Applied Mechanics and Engineering*, 254:114 – 125, 2013.
- [14] Malcolm F. Murphy, Gene H. Golub, and Andrew J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, 21(6):1969–1972, 2000.
- [15] Sergey V. Nepomnyaschikh. Mesh theorems of traces, normalizations of function traces and their inversions. *Sov. J. Numer. Anal. Math. Modeling*, 6:1–25, 1991.
- [16] Joseph E Pasciak and Jun Zhao. Overlapping schwarz methods in  $h$  (curl) on polyhedral domains. *Journal of Numerical Mathematics*, 10(3):221–234, 2002.
- [17] Luca F. Pavarino and Olof B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.
- [18] S. Reitzinger and J. Schöberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numerical Linear Algebra with Applications*, 9(3):223–238, 2002.
- [19] Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédéric Nataf, Clemens Pechstein, and Robert Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.*, 126(4):741–770, 2014.

- [20] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.