Adaptive Domain Decomposition method for Saddle Point problem in Matrix Form
Frédéric Nataf

To cite this version:
Frédéric Nataf. Adaptive Domain Decomposition method for Saddle Point problem in Matrix Form. 2019. hal-02343808v2

HAL Id: hal-02343808
https://hal.archives-ouvertes.fr/hal-02343808v2
Submitted on 14 Nov 2019

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.
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 \( m \leq n \). Let \( A \) be an \( n \times n \) 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:

\[
A := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}.
\]  

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:

\[
M^{-1}_A := R_0^T (R_0 AR_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T (R_i AR_i^T)^{-1} R_i ,
\]  

where \( R_0 \) is full rank \( \text{dim}(V_0) \times n \) 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, \( \text{dim}(V_0) \), is \( O(N) \) typically 10-20 times \( N \).

- The coarse space is made of extensions by zero of local vectors.
Using the GenEO metod \cite{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 $M_A^{-1}$:

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

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 $A$ (eq. (1)) where

$$S := C + B A^{-1} B^T,$$

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

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

This is done in § 3. Then in § 4.3 we will use $M_S$ to precondition the saddle point matrix $A$. Note that $M_S$ is by definition a sum of $N + 2$ positive semi definite matrices

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

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 $\Im(B R_i^T)$ 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

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

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

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

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

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

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:

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

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 $R_0 A R_0^T = L_0 L_0^T \in \mathbb{R}^{\text{dim}(V_0) \times \text{dim}(V_0)}$, we have a Cholesky factorization of $S_0$:

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

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

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

4
with \( y := L_0^{-1} \tilde{B}_0^T \tilde{R}_0 P \in \mathbb{R}^{\text{dim}(V_0)} \). This is equivalent to solving its Schur complement:

\[
A_0 y := (I + L_0^{-1} \tilde{B}_0^T \tilde{R}_0 S_1^{-1} \tilde{R}_0^T L_0^{-1}) y = L_0^{-1} \tilde{B}_0^T \tilde{R}_0 S_1^{-1} G \in \mathbb{R}^{\text{dim}(V_0)},
\]

(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 \( A_0 \in \mathbb{R}^{\text{dim}(V_0) \times \text{dim}(V_0)} \) whose dimension is small. But computing its entries amounts to solve \( \text{dim}(V_0) \) linear systems with \( S_1 \). Even if each solve is made scalable by a suitable parallel preconditioner \( M_{S_1} \), 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 an unpreconditioned Krylov method, convergence is in at most \( \text{dim}(V_0) \) 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:

\[
a : H \times H \to \mathbb{R} \quad a(P, Q) := (S_1 P, Q). \]

Let

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

and \( b \) be the following bilinear form:

\[
b : H_D \times H_D \to \mathbb{R} \quad b(P, Q) := \sum_{i=1}^N ((\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T) P_i, Q_i).
\]

We define \( R \):

\[
R : \quad H_D \to \quad H \quad \quad (P_i)_{1 \leq i \leq N} \mapsto \quad \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i P_i,
\]

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

$$\mathbf{P} = \mathcal{R}((\tilde{R}_i \mathbf{P})_{1 \leq i \leq N}).$$ (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 $\mathcal{O}(i) := \{1 \leq j \leq N \mid \tilde{R}_i \tilde{D}_i S_j \tilde{D}_j \tilde{R}_j^T \neq 0\}$:

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

On the other hand, we have by definition:

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

We can take:

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

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

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

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:

$$\tilde{D}_i \left( \sum_{j \in \mathcal{O}(i)} \tilde{R}_i \tilde{R}_j^T (\tilde{C}_j + \tilde{B}_j (R_j AR_i^T)^{-1} \tilde{B}_j^T) \tilde{R}_j \tilde{R}_i^T \right) \tilde{D}_i \mathbf{P}_{ik} = \lambda_{ik} (\tilde{C}_i + \tilde{B}_i (R_i AR_i^T)^{-1} \tilde{B}_i^T) \mathbf{P}_{ik}. $$ (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 $W_i \subset \mathbb{R}^{\text{rank}(\tilde{R}_i)}$:

$$\tilde{W}_i := \text{Span}\{\mathbf{P}_{ik} \mid \lambda_{ik} > \tau_{S_1}\}. $$ (10)
Then the coarse space \( \tilde{W}_0 \) is defined by
\[
\tilde{W}_0 := \bigoplus_{1 \leq i \leq N} \tilde{R}_i^T \tilde{D}_i \hat{W}_i.
\]

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
\[
\tilde{P}_0 = Z_{S_1}(Z_{S_1}^T S_{S_1})^{-1} Z_{S_1}^T S_{S_1}.
\] (11)

Finally, the preconditioner for \( S_1 \) reads
\[
M_{S_1}^{-1} := Z_{S_1}(Z_{S_1}^T S_{S_1})^{-1} Z_{S_1}^T + (I - \tilde{P}_0) \\
\times \left( \sum_{i=1}^{N} \tilde{R}_i^T \tilde{D}_i (\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)^\dagger \tilde{D}_i \tilde{R}_i \right) (I - \tilde{P}_0^T). \] (12)

Recall that we have for \( \alpha := \max(1, \frac{k_0}{\tau S_{S_1}}) \):
\[
\frac{1}{\alpha} M_{S_1}^{-1} \leq S^{-1} \leq M_{S_1}^{-1}.
\]

Remark 3.1

Solving a linear system with the Schur complement
\[
(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T)P_i = G_i,
\]
amounts to solving an augmented sparse system of the form
\[
- \begin{pmatrix} R_i A R_i^T & \tilde{B}_i^T \\ \tilde{B}_i & -\tilde{C}_i \end{pmatrix} \begin{pmatrix} U_i \\ P_i \end{pmatrix} = \begin{pmatrix} 0 \\ G_i \end{pmatrix}.
\]

3.2 Preconditioner for \( A_0 \)

Let us define
\[
M_{A_0} := (I + L_0^{-1} \tilde{B}_0^T \tilde{R}_0 M_{S_1}^{-1} \tilde{R}_0^T \tilde{B}_0 L_0^{-1})^T,
\] (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)):
\[
L_0^{-1} \tilde{B}_0^T \tilde{R}_0 Z_{S_1}(Z_{S_1}^T S_{S_1})^{-1} Z_{S_1}^T \tilde{R}_0^T \tilde{B}_0 L_0^{-1}.
\]
Matrix $Z_T^T \tilde{R}_0^T \in \mathbb{R}^{\dim(\tilde{W}_0) \times \dim(V_0)}$ 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 $M_S P = G \in \mathbb{R}^m$ with several Krylov solves with spectrally equivalent preconditioners:

\[ \text{Algorithm 1} \quad M_S^{-1} \text{ matvec product} \]

**INPUT:** $G \in \mathbb{R}^m$
1. Solve $S_1 G' = G$ by using the two-level preconditioner $M_S^{-1}$.
2. Compute the right handside $L_0^{-1} \hat{B}_0^T \hat{R}_0 G'$ of eq. (7).
3. Solve eq. (7) for $y$ 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 $M_S^{-1}$.
4. Solve $S_1 P = G - \hat{B}_0^T \hat{R}_0 L_0 T^{-1} y$ for $P$ using the two-level preconditioner $M_S^{-1}$.

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

4.3 DD solver for the saddle point system

We now consider the solving of the saddle point problem:

\[
\begin{pmatrix}
A & B^T \\
B & -C
\end{pmatrix}
\begin{pmatrix}
U \\
P
\end{pmatrix} =
\begin{pmatrix}
F_U \\
F_P
\end{pmatrix}.
\tag{14}
\]

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

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

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:

\[ \text{Algorithm 2} \quad \text{DD saddle point solver} \]

**INPUT:** $\begin{pmatrix}
F_U \\
F_P
\end{pmatrix} \in \mathbb{R}^{n+m}$
1. Solve $A G_U = F_U$ by a PCG with $M_A^{-1}$ as a preconditioner.
   Compute $G_P := F_P - B G_U$
2. Solve $(C + BA^{-1}B^T)P = -G_P$ by a PCG with $M_S^{-1}$ as a preconditioner, see Algorithm 1.
   Compute $G_U := F_U - B^T P$
3. Solve $A U = G_U$ by a PCG with $M_A^{-1}$ as a preconditioner.

**OUTPUT:** $\begin{pmatrix}
U \\
P
\end{pmatrix}$ 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:

$$
M^{-1}_{A_{SORAS}} := R_0^T (R_0AR_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T D_i A_i^{Rob^{-1}} D_i R_i,
$$

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

References


