This is the author's final, peer-reviewed manuscript as accepted for publication (AAM). The version presented here may differ from the published version, or version of record, available through the publisher's website. This version does not track changes, errata, or withdrawals on the publisher's site.

A robust algebraic multilevel domain decomposition preconditioner for sparse symmetric positive definite matrices

Hussam Al Daas and Pierre Jolivet

Published version information

Citation: H Al Daas and P Jolivet. 'A robust algebraic multilevel domain decomposition preconditioner for sparse symmetric positive definite matrices'. SIAM J Sci Comput, vol. 44, no. 4 (2022): A2582-A2598.

DOI: 10.1137/21M1446320

This version is made available under a CC-BY licence. Please cite only the published version using the reference above. This is the citation assigned by the publisher at publication. Please check the publisher's website for any updates.

This item was retrieved from **ePubs**, the Open Access archive of the Science and Technology Facilities Council, UK. Please contact epublications@stfc.ac.uk or go to http://epubs.stfc.ac.uk/ for further information and policies.

A ROBUST ALGEBRAIC MULTILEVEL DOMAIN DECOMPOSITION PRECONDITIONER FOR SPARSE SYMMETRIC POSITIVE DEFINITE MATRICES[*]

HUSSAM AL DAAS[†] AND PIERRE JOLIVET[‡]

Abstract. Domain decomposition (DD) methods are widely used as preconditioner techniques. Their effectiveness relies on the choice of a locally constructed coarse space. Thus far, this construction was mostly achieved using non-assembled matrices from discretized partial differential equations (PDEs). Therefore, DD methods were mainly successful when solving systems stemming from PDEs. In this paper, we present a fully algebraic multilevel DD method where the coarse space can be constructed locally and efficiently without any information besides the coefficient matrix. The condition number of the preconditioned matrix can be bounded by a user-prescribed number. Numerical experiments illustrate the effectiveness of the preconditioner on a range of problems arising from different applications.

Key words. Algebraic domain decomposition, multilevel preconditioner, overlapping Schwarz method, sparse linear system.

1. Introduction.

We are interested in solving the linear system of equations

$$Ax = b,$$

where $A \in \mathbb{R}^{n\times n}$ is a sparse symmetric positive definite (SPD) matrix and $b \in \mathbb{R}^n$ is the right-hand side. On the one hand, despite their accuracy, direct methods [14] that are based on matrix factorizations become memory and computationally demanding for large-scale problems. Furthermore, establishing a high level of concurrency in their algorithm is challenging, which limits the effectiveness of their parallelization with many processing units, e.g., thousands of MPI processes. On the other hand, iterative methods, such as Krylov subspace methods, are attractive as they require less memory resources and parallelizing them is easier. However, their convergence depends on the coefficient matrix A. More precisely, the error at iteration k of the conjugate gradient method [23] satisfies

$$\|x_k - x_\star\|_A \leq 2\|x_0 - x_\star\|_A \left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k,$$

where $x_\star$ is the exact solution and $\kappa_2(A)$ is the spectral condition number of A. Therefore, iterative methods are usually combined with preconditioners that modify the properties of the linear system such that the convergence rate of the method is improved. A variety of preconditioning techniques have been proposed in the literature, see the recent survey [39] and references therein. We focus in this work on preconditioners for SPD matrices. In terms of construction type, these preconditioners can be split into two categories. (1) *Algebraic* preconditioners: those do not require information from the problem besides the linear system, and their construction relies only on A and b [5, 24, 33, 37, 41]. (2) *Analytic* preconditioners: in order to construct them, more information from the origin of the linear system, e.g., matrix

[*] Submitted to the editors June 7, 2022.

[†] STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX, UK (hussam.al-daas@stfc.ac.uk).

[‡] CNRS, ENSEEIHT, 2 rue Charles Camichel, 31071 Toulouse Cedex 7, France (pierre.jolivet@enseeiht.fr).

assembly procedure, is required [27, 28, 44]. Inferring how preconditioners modify the spectrum of iteration matrices provides another way to classify them. Again, two categories exist. (1) *One-level* preconditioners: those mostly rely on incomplete matrix factorizations, matrix splitting methods, approximate sparse inverse methods, and Schwarz methods [41]. One-level preconditioners usually bound from above the largest eigenvalue of the preconditioned matrix. (2) *Two-level* and *multilevel* preconditioners: those are usually a combination of a one-level method and a coarse space correction. While the one-level part can bound from above the largest eigenvalue, the coarse space is used to bound from below the smallest eigenvalue such that the condition number of the preconditioned matrix is bounded [2, 3, 4, 13, 17, 18, 20, 21, 30, 31, 33, 52, 45, 48].

When it comes to overlapping DD, most one-level preconditioners and a few two-level/multilevel preconditioners are algebraic, while most two-level preconditioners are analytic. On the one hand, analytic two-level/multilevel preconditioners construct the coarse space efficiently without requiring computations involving the global matrix. On the other hand, existing algebraic two-level/multilevel preconditioners still require global computations involving the matrix A that limit the setup scalability [2, 18]. Furthermore, certain algebraic two-level preconditioners require complicated operations that may not be easy to parallelize. Therefore, we focus in this paper on two-level/multilevel preconditioners where the coarse space can be constructed locally. Certain algebraic multigrid (AMG) methods are examples of these preconditioners [37]. Note that several AMG methods require unassembled matrices or the near-nullspace of the global matrix, which is known in some applications [11, 47]. One could argue that these methods are thus not purely algebraic. Furthermore, their effectiveness has been proved only for certain classes of matrices. An algebraic two-level preconditioner for the normal matrix equations was recently proposed in [4].

In [2], the authors presented an algebraic framework to construct robust coarse spaces and characterized a class of local symmetric positive semi-definite (SPSD) matrices that allows to construct such coarse spaces efficiently. Since then, there have been several attempts to construct algebraic two-level preconditioners with a locally computed coarse space that are theoretically effective on any sparse SPD matrix, see, e.g., [18] and references therein. Starting off with the subdomain matrices of A, the authors in [18] define an auxiliary matrix A_+ such that $A - A_+$ is low-rank and a local SPSD splitting for A_+ is easily obtained. A robust algebraic two-level preconditioner for A is then derived by a low-rank update of the robust algebraic two-level preconditioner of A_+. Despite the fact that the preconditioner proposed in [18] is fully algebraic, using it in practice may not be very attractive since the low-rank update requires the solution of linear systems with A_+ involving a large number of right-hand sides that is nearly equal to the size of the coarse space of A_+, which is prohibitive for large number of subdomains. Therefore, we believe that the question of finding efficient locally constructed coarse spaces is still open.

When information such as the near-nullspace or the subdomain non-assembled matrices are available, analytic AMG or DD preconditioners are optimal. The preconditioner presented in this paper should be used when a robust black-box solver is needed.

The manuscript is organized as follows. We introduce the notation and review the algebraic DD framework in Section 2. Section 3 presents our main contribution in finding local SPSD splitting matrices associated with each subdomain fully algebraically in an inexpensive way and starting from local data. These matrices will be used to construct a robust two-level Schwarz preconditioner. Then, we briefly discuss the straightforward extension of our approach to a multilevel preconditioner.

Afterwards, we present in Section 4 numerical experiments on problems arising from different engineering applications. Concluding remarks and future lines of research are given in Section 5.

Notation. We end our introduction by defining notations that will be used in this paper. Let $1 \leq n \leq m$ and let $B \in \mathbb{R}^{m \times n}$. Let $S_1 \subset [\![1, m]\!]$ and $S_2 \subset [\![1, n]\!]$ be two sets of integers. $B(S_1, :)$ is the submatrix of B formed by the rows whose indices belong to S_1 and $B(:, S_2)$ is the submatrix of B formed by the columns whose indices belong to S_2. The matrix $B(S_1, S_2)$ is formed by taking the rows whose indices belong to S_1 and only retaining the columns whose indices belong to S_2. The concatenation of any two sets of integers S_1 and S_2 is represented by $[S_1, S_2]$. Note that the order of the concatenation is important. The set of the first p positive integers is denoted by $[\![1, p]\!]$. The identity matrix of size n is denoted by I_n.

2. Domain decomposition.

Throughout this section, we assume that C is a general $n \times n$ sparse SPD matrix. Let the nodes V in the corresponding adjacency graph $\mathcal{G}(C)$ be numbered from 1 to n. A graph partitioning algorithm can be used to split V into $N \ll n$ disjoint subsets Ω_{Ii} $(1 \leq i \leq N)$ of size n_{Ii}. These sets are called nonoverlapping subdomains.

2.1. Abstract setting for two-level overlapping Schwarz methods.

Defining first a one-level Schwarz preconditioner requires overlapping subdomains. Let $\Omega_{\Gamma i}$ be the subset of size $n_{\Gamma i}$ of nodes that are distance one in $\mathcal{G}(C)$ from the nodes in Ω_{Ii} $(1 \leq i \leq N)$. The overlapping subdomain Ω_i is defined to be $\Omega_i = [\Omega_{Ii}, \Omega_{\Gamma i}]$, with size $n_i = n_{\Gamma i} + n_{Ii}$. The complement of Ω_i in $[\![1, n]\!]$ is denoted by $\Omega_{\mathrm{c}_\Gamma i}$.

Associated with Ω_i is a restriction (or projection) matrix $R_i \in \mathbb{R}^{n_i \times n}$ given by $R_i = I_n(\Omega_i, :)$. R_i maps from the global domain to subdomain Ω_i. Its transpose $R_i^\top$ is a prolongation matrix that maps from subdomain Ω_i to the global domain.

The theory in this paper requires a decomposition of the graph of C^2. Hence, in addition to the previous subsets, we define the following ones. We denote $\Omega_{\Delta i}$ the subset of size $n_{\Delta i}$ containing nodes that are not in Ω_{Ii} and distance one in $\mathcal{G}(C)$ from the nodes in $\Omega_{\Gamma i}$ $(1 \leq i \leq N)$. The extended overlapping subdomain $\widetilde{\Omega}_i$ is defined to be $\widetilde{\Omega}_i = [\Omega_{Ii}, \Omega_{\Gamma i}, \Omega_{\Delta i}]$ and it is of size $\widetilde{n}_i$. We denote the complement of $\widetilde{\Omega}_i$ in $[\![1, n]\!]$ by $\Omega_{\mathrm{c}_\Delta i}$. Associated with $\widetilde{\Omega}_i$ is a restriction matrix $\widetilde{R}_i \in \mathbb{R}^{\widetilde{n}_i \times n}$ given by $\widetilde{R}_i = I_n(\widetilde{\Omega}_i, :)$. $\widetilde{R}_i$ maps from the global domain to the extended overlapping subdomain $\widetilde{\Omega}_i$. Its transpose $\widetilde{R}_i^\top$ is a prolongation matrix that maps from the extended overlapping subdomain $\widetilde{\Omega}_i$ to the global domain.

The *one-level additive Schwarz preconditioner* [13] is defined to be

$$M_{\text{ASM}}^{-1} = \sum_{i=1}^{N} R_i^\top C_{ii}^{-1} R_i, \qquad C_{ii} = R_i C R_i^\top.$$

Applying this preconditioner to a vector involves solving concurrent local problems in the overlapping subdomains. Increasing N reduces the sizes n_i of the overlapping subdomains, leading to smaller local problems and faster computations. However, in practice, the preconditioned system using M_{ASM}^{-1} may not be well-conditioned, inhibiting convergence of the iterative solver. In fact, the local nature of this preconditioner can lead to a deterioration in its effectiveness as the number of subdomains increases because of the lack of global information from the matrix C [13, 17]. To maintain robustness with respect to N, a coarse space is added to the preconditioner (also known as second-level correction) that includes global information.

Let $0 < n_C \ll n$. If $R_0 \in \mathbb{R}^{n_C \times n}$ is of full row rank, the *two-level additive Schwarz preconditioner* [13] is defined to be

$$M_{\text{additive}}^{-1} = \sum_{i=0}^{N} R_i^\top C_{ii}^{-1} R_i = R_0^\top C_{00}^{-1} R_0 + M_{\text{ASM}}^{-1}, \qquad C_{00} = R_0 C R_0^\top. \tag{2.1}$$

Observe that, since C and R_0 are of full rank, C_{00} is also of full rank. For any full rank R_0, it is possible to cheaply obtain upper bounds on the largest eigenvalue of the preconditioned matrix, independently of n and N [2]. However, bounding the smallest eigenvalue is highly dependent on R_0. Therefore, the choice of R_0 is key to obtaining a well-conditioned system and building efficient two-level Schwarz preconditioners. Two-level Schwarz preconditioners have been used to solve a large class of systems arising from a range of engineering applications (see, for example, [20, 26, 32, 34, 43, 49] and references therein).

Following [2], we denote by $D_i \in \mathbb{R}^{n_i \times n_i}$ $(1 \le i \le N)$ any non-negative diagonal matrices such that

$$\sum_{i=1}^{N} R_i^\top D_i R_i = I_n.$$

We refer to $(D_i)_{1 \le i \le N}$ as an *algebraic partition of unity.* In [2], Al Daas and Grigori show how to select local subspaces $Z_i \in \mathbb{R}^{n_i \times p_i}$ with $p_i \ll n_i$ $(1 \le i \le N)$ such that, if $R_0^\top$ is defined to be $R_0^\top = [R_1^\top D_1 Z_1, \ldots, R_N^\top D_N Z_N]$, the spectral condition number of the preconditioned matrix $M_{\text{additive}}^{-1} C$ is bounded from above independently of N and n.

2.2. Algebraic local SPSD splitting of an SPD matrix. We now recall the definition of an algebraic local SPSD splitting of an SPD matrix given in [2] and generalized in [3].

An *algebraic local SPSD splitting* of the SPD matrix C with respect to subdomain i is defined to be any SPSD matrix $\widetilde{C}_i \in \mathbb{R}^{n \times n}$ that satisfies the following

$$\begin{aligned} &0 \le u^\top \widetilde{C}_i u \le u^\top C u, \quad \text{for all } u \in \mathbb{R}^n, \\ &R_{\text{c}\Gamma i} \widetilde{C}_i = 0. \end{aligned}$$

We denote the nonzero submatrix of $\widetilde{C}_i$ by $\widetilde{C}_{ii}$ so that

$$\widetilde{C}_i = R_i^\top \widetilde{C}_{ii} R_i.$$

Associated with the local SPSD splitting matrices, we define a multiplicity constant k_m that satisfies the inequality

$$0 \le \sum_{i=1}^{N} u^\top \widetilde{C}_i u \le k_m u^\top C u, \quad \text{for all } u \in \mathbb{R}^n. \tag{2.2}$$

Note that, for any set of SPSD splitting matrices, $k_m \le N$.

The main motivation for defining splitting matrices is to find local seminorms that are bounded from above by the C-norm. These seminorms will be used to determine a subspace that contains the eigenvectors of C associated with its smallest eigenvalues.

We remind the reader of the result presented in [2] on the upper bound of the condition number of the preconditioned matrix by using the two-level Schwarz preconditioner with the coarse space constructed with local SPSD splitting matrices.

THEOREM 2.1. *[2, Theorem 4.5] Let $\widetilde{A}_{ii}$ be a local SPSD splitting matrix associated with subdomain i, for $i = 1, \dots, N$.Let $\tau > 0$, and let R_0 be a matrix whose columns span the subspace $Z = \bigoplus_{i=1}^{N} R_i^\top D_i Z_i$, where Z_i is defined as:*

$$Z_i = \mathit{span}\{u \mid D_i A_{ii} D_i u = \lambda \widetilde{A}_{ii} u, \text{ and } \lambda > 1/\tau\}.$$

Then, the condition number of the preconditioned matrix $M_{\mathit{additive}}^{-1}A$, noted $\kappa\left(M_{\mathit{additive}}^{-1}A\right)$, where $M_{\mathit{additive}}^{-1}$ is defined in (2.1), satisfies

$$\kappa\left(M_{\mathit{additive}}^{-1}A\right) \le (k_c + 1)\left(2 + (2k_c + 1)\frac{k_m}{\tau}\right),$$

where k_c is the number of colors required to color the graph of A such that any two neighboring subdomains have different colors and k_m is the multiplicity constant that satisfies (2.2).

3. Local SPSD splitting matrices.

In this section we show how to construct local SPSD splitting matrices of a sparse SPD matrix efficiently using only local subdomain information.

3.1. From normal equations matrices to general SPD matrices.

In [4], the authors presented how to compute local SPSD splitting matrices for the normal equations matrix $C = B^\top B$ where $B \in \mathbb{R}^{m \times n}$. Considering the case $B = A$, we have $C = A^2$. Thus, provided the theory developed in [4], we can compute local SPSD splitting matrices of A^2 efficiently. Using the permutation matrix $P_i = I(\Omega_{Ii}, \Omega_{\Gamma i}, \Omega_{\Delta i}, \Omega_{\mathrm{c}_\Delta i}], :)$, we can write

$$P_i A P_i^\top = \begin{pmatrix} A_{Ii} & A_{I\Gamma i} & & \\ A_{\Gamma I i} & A_{\Gamma i} & A_{\Gamma\Delta i} & \\ & A_{\Delta\Gamma i} & A_{\Delta i} & A_{\Delta \mathrm{c}_\Delta i} \\ & & A_{\mathrm{c}_\Delta \Delta i} & A_{\mathrm{c}_\Delta i} \end{pmatrix},$$

and

$$\widetilde{C}_i = \widetilde{R}_i^\top X_i^\top X_i \widetilde{R}_i$$

is an SPSD splitting of A^2, where X_i is given as

$$X_i = R_i A \widetilde{R}_i^\top = \begin{pmatrix} A_{Ii} & A_{I\Gamma i} & \\ A_{\Gamma I i} & A_{\Gamma i} & A_{\Gamma\Delta i} \end{pmatrix}. \tag{3.1}$$

Remark 3.1. All terms from (3.1) stem from the original coefficient matrix A, in the sense that there is no connection with the underlying discretization scheme or matrix assembly procedure. In a parallel computing context, e.g., if A is distributed following a contiguous one-dimensional row partitioning among MPI processes, all terms may be retrieved using peer-to-peer communication between neighboring processes.

Lemma 3.2 demonstrates how to obtain a local SPSD splitting of A with respect to the extended overlapping subdomains given an SPSD splitting of A^2.

Lemma 3.2. *Let $\widetilde{C}_i$ be a local SPSD splitting of $C = A^2$, and let $\underline{\widetilde{A}}_i$ be the square root SPSD matrix of $\widetilde{C}_i$ such that $\underline{\widetilde{A}}_i^2 = \widetilde{C}_i$. Then, $\underline{\widetilde{A}}_i$ is a local SPSD splitting of A with respect to the extended overlapping subdomain $\widetilde{\Omega}_i$.*

Proof. First, observe that for any vector $u \in \mathbb{R}^n$,

$$u^\top (A^2 - \underline{\widetilde{A}}_i^2) u = u^\top (A + \underline{\widetilde{A}}_i)(A - \underline{\widetilde{A}}_i) u.$$

Since $A + \underline{\widetilde{A}}_i$ is SPD, we can write $A + \underline{\widetilde{A}}_i = W_i^\top W_i$, and we have

$$\begin{aligned} u^\top W_i (A - \underline{\widetilde{A}}_i) W_i^{-1} u &= u^\top W_i^{-\top} W_i^\top W_i (A - \underline{\widetilde{A}}_i) W_i^{-1} u \\ &= v^\top W_i^\top W_i (A - \underline{\widetilde{A}}_i) v \\ &= v^\top (A + \underline{\widetilde{A}}_i)(A - \underline{\widetilde{A}}_i) v \\ &= v^\top (A^2 - \underline{\widetilde{A}}_i^2) v \\ &\geq 0, \end{aligned}$$

where $v = W_i^{-1} u$. Since $W_i (A - \underline{\widetilde{A}}_i) W_i^{-1}$ and $A - \underline{\widetilde{A}}_i$ have the same eigenvalues, we conclude that $A - \underline{\widetilde{A}}_i$ is SPSD. The locality of $\underline{\widetilde{A}}_i$ stems from the locality of $\widetilde{C}_i$. □

We note that the SPSD splitting $\underline{\widetilde{A}}_i$ obtained from the SPSD splitting of A^2 is local with respect to the extended overlapping subdomain $\widetilde{\Omega}_i$. A Schur complement technique can be applied to obtain the locality to the subdomain Ω_i. Lemma 3.3 presents how to obtain a local SPSD splitting matrix of A with respect to the subdomain Ω_i from the local SPSD splitting of A with respect to the extended overlapping subdomain $\widetilde{\Omega}_i$.

Lemma 3.3. *Let $\underline{\widetilde{A}}_i = \widetilde{R}_i^\top \underline{\widetilde{A}}_{ii} \widetilde{R}_i$ be a local SPSD splitting of A with respect to the extended overlapping subdomain $\widetilde{\Omega}_i$. Let $\underline{\widetilde{A}}_{ii}$ be written as a $(2,2)$ block matrix such that the $(1,1)$ block corresponds to the overlapping subdomain Ω_i and the $(2,2)$ block corresponds to $\Omega_{\Delta i}$, i.e.,*

$$\underline{\widetilde{A}}_{ii} = \begin{pmatrix} X_{i,11} & X_{i,12} \\ X_{i,21} & X_{i,22} \end{pmatrix},$$

and let

$$\widetilde{A}_{ii} = X_{i,11} - X_{i,12} X_{i,22}^{-1} X_{i,21}, \tag{3.2}$$

where we assume that $X_{i,22}$ is SPD. Then, $\widetilde{A}_i = R_i^\top \widetilde{A}_{ii} R_i$ is an SPSD splitting of A with respect to the subdomain Ω_i.

Proof. We have

$$\begin{aligned} \underline{\widetilde{A}}_{ii} &= \begin{pmatrix} X_{i,11} & X_{i,12} \\ X_{i,21} & X_{i,22} \end{pmatrix} \\ &= \begin{pmatrix} X_{i,11} - X_{i,12} X_{i,22}^{-1} X_{i,21} & \\ & \end{pmatrix} + \begin{pmatrix} X_{i,12} X_{i,22}^{-1} X_{i,21} & X_{i,12} \\ X_{i,21} & X_{i,22} \end{pmatrix}. \end{aligned}$$

Since $X_{i,22}$ is SPD and $\underline{\widetilde{A}}_{ii}$ is SPSD, $X_{i,11} - X_{i,12} X_{i,22}^{-1} X_{i,21}$ is SPSD. Therefore,

$$\begin{aligned} 0 \leq u^\top \widetilde{A}_i u &= u^\top R_i^\top \widetilde{A}_{ii} R_i u \\ &\leq u^\top \widetilde{R}_i^\top \underline{\widetilde{A}}_{ii} \widetilde{R}_i u \\ &\leq u^\top A u. \end{aligned}$$

□

Remark 3.4. Since the SPSD splitting will be used to construct a preconditioner, the assumption in Lemma 3.3 that $X_{i,22}$ is SPD can be obtained by shifting its diagonal elements by a small value such as $\|X_{i,22}\|_2\varepsilon$, where ε is the floating-point machine precision. One can also shift the diagonal values of the matrix $\underline{\widetilde{A}}_{ii}$ by a small value $\|\underline{\widetilde{A}}_{ii}\|_2\varepsilon$ so that the Schur complement can be well defined.

In the following section, we explain how to compute the local SPSD splitting matrices efficiently.

3.2. Practical construction of local SPSD matrices. The construction of robust two-level overlapping Schwarz preconditioners is based on computing the coarse space projection operator R_0. Following Theorem 2.1 and using the local SPSD splitting matrices of A, R_0 can be chosen as the matrix that spans the space

$$Z = \bigoplus_{i=1}^{N} R_i^\top D_i Z_i,$$

where Z_i is defined to be

$$Z_i = \operatorname{span}\{u \mid D_i A_{ii} D_i u = \lambda \widetilde{A}_{ii} u, \text{ and } \lambda > 1/\tau\}, \tag{3.3}$$

where $\tau > 0$ is a user-specified number. The condition number of the preconditioned matrix $M_{\text{additive}}^{-1}A$ is bounded from above by $(k_c+1)\left(2+(2k_c+1)\frac{k_m}{\tau}\right)$, where k_c is the number of colors required to color the graph of A such that any two neighboring subdomains have different colors and k_m is the multiplicity constant that satisfies (2.2).

Solving the generalized eigenvalue problem in (3.3) using iterative solvers such as the Krylov–Schur method [46] requires solving linear systems of the form $\widetilde{A}_{ii}u = v$. The matrix $\widetilde{A}_{ii}$ is the Schur complement of the matrix $\underline{\widetilde{A}}_{ii} = \left(X_i^\top X_i\right)^{\frac{1}{2}}$, where $X_i = R_i A \widetilde{R}_i^\top$. Let $X_i = U_i \Sigma_i V_i^\top$ be the economic singular-value decomposition of X_i and let $V_i^\perp$ be an orthonormal matrix whose columns form a complementary basis of the columns of V_i, i.e., $[V_i, V_i^\perp]$ is an orthogonal matrix. Note that $V_i^\perp (V_i^\perp)^\top = I_{\tilde{n}_i} - V_i V_i^\top$. Using Remark 3.4, $\underline{\widetilde{A}}_{ii}$ can be chosen as

$$\begin{aligned}
\underline{\widetilde{A}}_{ii} &= V_i \Sigma_i V_i^\top + \sigma_{1i}\varepsilon I_{\tilde{n}_i} \\
&= V_i \Sigma_i V_i^\top + \sigma_{1i}\varepsilon [V_i, V_i^\perp][V_i, V_i^\perp]^\top \\
&= V_i(\Sigma_i + \sigma_{1i}\varepsilon I_{n_i})V_i^\top + \sigma_{1i}\varepsilon V_i^\perp (V_i^\perp)^\top \\
&= V_i(\Sigma_i + \sigma_{1i}\varepsilon I_{n_i})V_i^\top + \sigma_{1i}\varepsilon (I_{\tilde{n}_i} - V_i V_i^\top),
\end{aligned}$$

where σ_{1i} is the largest singular value of X_i. One way to solve the linear system $\widetilde{A}_{ii}u = v$ is thus to solve the augmented linear system

$$\underline{\widetilde{A}}_{ii} \begin{pmatrix} u \\ y \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix}.$$

Given the singular-value decomposition of $\underline{\widetilde{A}}_{ii}$, the solution u can be obtained efficiently. Indeed, the inverse of $\underline{\widetilde{A}}_{ii}$ is

$$\underline{\widetilde{A}}_{ii}^{-1} = V_i(\Sigma_i + \sigma_{1i}\varepsilon I_{n_i})^{-1} V_i^\top + \sigma_{1i}^{-1}\varepsilon^{-1}(I_{\tilde{n}_i} - V_i V_i^\top). \tag{3.4}$$

In our current implementation, the singular-value decomposition is computed concurrently using LAPACK [6]. This implies that the sparse matrix X_i, see (3.1), is converted to a dense representation. Then, $\widetilde{\underline{A}}_{ii}$ is never assembled, and instead, the action of its inverse is applied in a matrix-free fashion using (3.4). Since these operations are local to each subdomain, they remain tractable. However, it could be beneficial to leverage the lower memory-footprint of iterative sparse singular-value solvers, e.g., PRIMME_SVDS [51]. To the best of our knowledge, no such solver may be used to retrieve the complete economic singular-value decomposition of a sparse matrix.

Since the construction of the two-level method is fully algebraic, one can successively apply the same approach on the coarse space matrix to obtain a multilevel preconditioner in which, the condition number of each preconditioned matrix is bounded from above by a prescribed number. Note that if the matrices $\widetilde{A}_{ii}$ for $i = 1, \ldots, N$ are formed explicitly as in (3.2), we can use the strategy that we proposed in [3] to construct a multilevel preconditioner with the same properties.

3.3. Computational complexity. We present in this section the computational complexity of the operations involving the SPSD splitting matrix since all other computations required to construct the preconditioner are identical to the ones in a classic two-level DD method.

The main computation involved in the proposed SPSD splitting is the economic singular-value decomposition of the matrix $A(\Omega_i, \widetilde{\Omega}_i)$, which is of size $n_i \times \widetilde{n}_i$, computed concurrently for each subdomain. As we currently compute its SVD explicitly, the computational cost is $O(\widetilde{n}_i n_i^2)$, equivalent to $O(n_i^3)$ for most sparse matrices. Once the SVD is obtained, we use it as presented in (3.4) to solve linear systems arising in the Krylov–Schur method to solve the generalized eigenvalue problem defined in (3.3). Applying $\widetilde{\underline{A}}_{ii}^{-1}$ as in (3.4) to a vector v of size $\widetilde{n}_i$ requires $O(\widetilde{n}_i n_i)$ operations.

We would like to stress that due to the high computational cost of computing the dense SVD, we expect that this will dominate runtime. However, since n_i and $\widetilde{n}_i$ decrease approximately linearly as the number of subdomains increases, we expect a linear strong scalability of the SVD computation and the application of $\widetilde{\underline{A}}_{ii}^{-1}$ as in (3.4).

4. Numerical experiments. In this section, we present a variety of numerical experiments that show the effectiveness and efficiency of the proposed preconditioner. First, we compare it against state-of-the-art algebraic multigrid preconditioners including AGMG [36, 37], BoomerAMG [16], and GAMG [1]. Then, we include numerical experiments where the proposed preconditioner is used to solve coarse problems from other multilevel solvers, thus emphasizing the algebraic and robust traits of our method. Except for AGMG which is used through its MATLAB interface, all these experiments are performed using PETSc [7]. In particular, the proposed preconditioner is a natural extension of the PCHPDDM infrastructure [26] which we use to solve the concurrent generalized eigenvalue problems from (3.3) via SLEPc [22], and then to define our multilevel preconditioner by selecting the appropriate local eigenmodes depending on the user-specified value of τ. With respect to Remark 3.1, we use the PETSc routine `MatCreateSubMatrices`, see https://petsc.org/release/docs/manualpages/Mat/MatCreateSubMatrices.html. Instead of using M_{additive}^{-1} as defined in (2.1), we will use M_{deflated}^{-1}, defined as

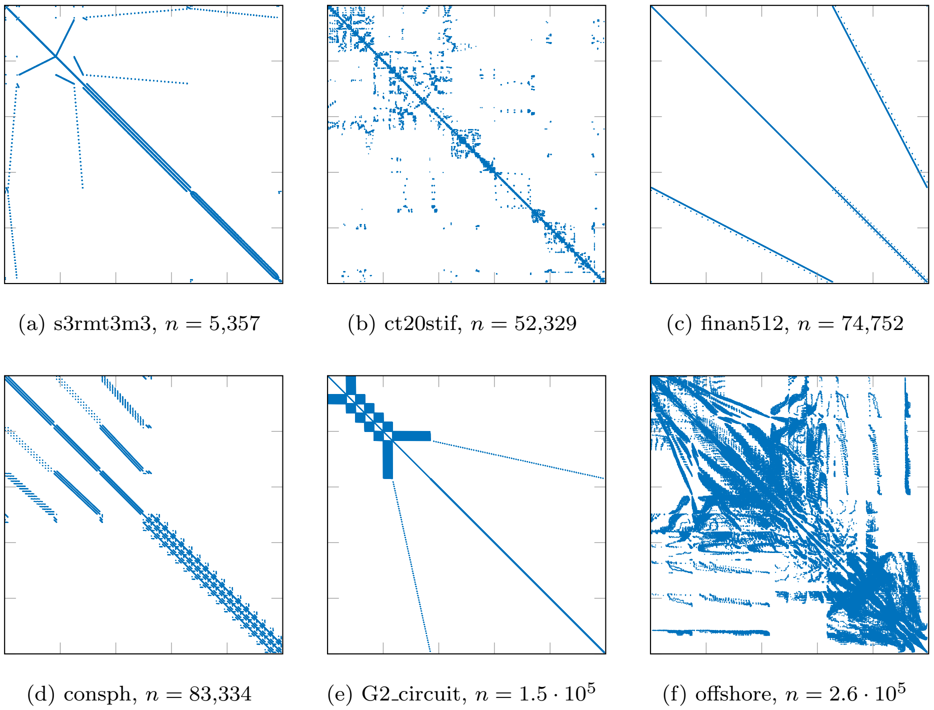
$$M_{\text{deflated}}^{-1} = R_0^\top C_{00}^{-1} R_0 + M_{\text{RAS}}^{-1}(I_n - C R_0^\top C_{00}^{-1} R_0),$$

Table 1
Test matrices taken from the SuiteSparse Matrix Collection.

Identifier	n	nnz(A)	condest(A)
s3rmt3m3	5,357	207,123	$4.4 \cdot 10^{10}$
vanbody	47,072	2,329,056	$9.4 \cdot 10^{18}$
gridgena	48,962	512,084	$7.1 \cdot 10^{5}$
ct20stif	52,329	2,600,295	$2.2 \cdot 10^{14}$
nasasrb	54,870	2,677,324	$1.5 \cdot 10^{9}$
Dubcova2	65,025	1,030,225	10,411
finan512	74,752	596,992	98.4
consph	83,334	6,010,480	$3.2 \cdot 10^{7}$
s3dkt3m2	90,449	3,686,223	$6.3 \cdot 10^{11}$
shipsec8	114,919	3,303,553	$1.5 \cdot 10^{14}$
ship_003	121,728	3,777,036	$2.6 \cdot 10^{16}$
boneS01	127,224	5,516,602	$4.2 \cdot 10^{7}$
bmwcra_1	148,770	10,641,602	$9.7 \cdot 10^{8}$
G2_circuit	150,102	726,674	$2 \cdot 10^{7}$
pwtk	217,918	11,524,432	$5 \cdot 10^{12}$
offshore	259,789	4,242,673	$2.3 \cdot 10^{13}$
af_4_k101	503,625	17,550,675	$6.5 \cdot 10^{8}$
parabolic_fem	525,825	3,674,625	$2.1 \cdot 10^{5}$
apache2	715,176	4,817,870	$5.3 \cdot 10^{6}$
tmt_sym	726,713	5,080,961	$1.1 \cdot 10^{9}$
ecology2	999,999	4,995,991	$6.7 \cdot 10^{7}$

where M_{RAS}^{-1} is the well-known one-level restricted additive Schwarz method [10]. The choice of M_{deflated}^{-1} over M_{additive}^{-1} is motivated by previous results from the literature [48], which exhibit better numerical property of the former over the latter. Table 1 presents the set of test matrices from the SuiteSparse Matrix Collection [12] that are used first. They represent a subset of the matrices from the collection which satisfy both criteria "*Special Structure* equal to *Symmetric*" and "*Positive Definite* equal to *Yes*". We highlight the fact that our proposed preconditioner can handle unstructured systems, not necessarily stemming from standard PDE discretization schemes, by displaying some nonzero patterns in Figure 1.

4.1. The algebraic two-level case. In this section, we present a numerical comparison between our proposed preconditioner and three algebraic multigrid solvers: AGMG, BoomerAMG, and GAMG. Even though matrices from Table 1 are SPD, all three AMG solvers encounter difficulties in solving many of the associated linear systems with random right-hand sides. On the contrary, our algebraic two-level preconditioner M_{deflated}^{-1} is more robust and always reaches the prescribed tolerance of 10^{-8}. Note that a simple one-level preconditioner such as M_{RAS}^{-1} with a minimal overlap of one does not converge for these problems. The outer Krylov method is the right-preconditioned GMRES(30) [42]. For preconditioners used within PETSc (all except AGMG), the systems are solved using 256 MPI processes and are first renumbered by ParMETIS [29]. For our DD method, a single subdomain is mapped to each process, i.e., $N = 256$ in (2.1). Furthermore, exact subdomain and second-level operator Cholesky factorizations are computed. In the second to last column of Table 2, sizes of second-level operator are reported. Eventually, in the last column,



(a) s3rmt3m3, $n = 5{,}357$

(b) ct20stif, $n = 52{,}329$

(c) finan512, $n = 74{,}752$

(d) consph, $n = 83{,}334$

(e) G2_circuit, $n = 1.5 \cdot 10^5$

(f) offshore, $n = 2.6 \cdot 10^5$

FIG. 1. *Nonzero sparsity pattern of some of the test matrices from Table 1.*

grid complexities (GC), as usually defined in the multigrid community [15], are reported. One may notice that they fluctuate among matrices. Indeed, for small-sized problem s3rmt3m3, the grid complexity is $\frac{5{,}357+5{,}321}{5{,}357} = 1.99$, while for problem parabolic_fem, it is $\frac{5.26 \cdot 10^5+21{,}736}{5.26 \cdot 10^5} = 1.04$.

Corresponding to the numerical results of our preconditioner in Table 2, we report in Table 3 the runtime on the first process of the most significant steps during the setup and solve phases. As predicted in subsection 3.3, the computation of the dense SVD dominates other computations for almost all problems, especially for the ones where the local problem size $n_i \approx \frac{n}{N}$ is sufficiently large. We present in subsection 4.3 the scalability of the setup and solve phases.

4.2. The nested-level case. Since our proposed preconditioner is fully algebraic, we now use it recursively to solve the second-level operator from the previous section using yet another two-level method instead of using an exact Cholesky factorization. This is referred to as K-cycle in the multigrid community [38]. This thus yields an algebraic three-level preconditioner. PCHPDDM has the capability of automatically redistributing coarse operators on a subset of MPI processes on which the initial coefficient matrix A is distributed [25]. We still use 256 MPI processes for the fine-level decomposition, then use four processes for the second-level decomposition, and the third-level operator is centralized on a single process. The outer solver is now the flexible GMRES(30) [40]. Second-level systems are this time solved with the right-preconditioned GMRES(30), with a higher tolerance set to 10^{-4}, compared to the outer-solver tolerance of 10^{-8}. We investigate problems

Table 2

Preconditioner comparison: iteration counts are reported in the columns 2–5 if convergence to the prescribed tolerance of 10^{-8} is achieved in 100 iterations or less. In column 6, sizes of the second-level operator generated by our proposed preconditioner are reported. In column 7, grid complexities (GC) are reported.

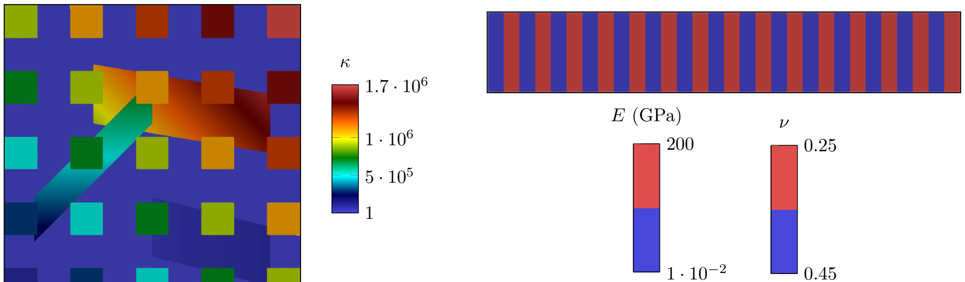
Identifier	AGMG	BoomerAMG	GAMG	M^{-1}_{deflated}	n_C	GC
s3rmt3m3				4	5,321	1.99
vanbody				18	25,600	1.54
gridgena				2	16,706	1.34
ct20stif				4	49,421	1.94
nasasrb				10	25,600	1.47
Dubcova2		76	56	5	12,729	1.2
finan512	9	7	8	4	15,271	1.2
consph				26	25,600	1.31
s3dkt3m2				49	25,592	1.28
shipsec8				7	76,800	1.67
ship_003				9	76,759	1.63
boneS01				16	25,600	1.2
bmwcra_1				20	76,800	1.52
G2_circuit	29	11	26	19	21,602	1.14
pwtk				47	25,600	1.12
offshore				7	76,800	1.3
af_4_k101				18	76,800	1.15
parabolic_fem	12	8	16	17	21,736	1.04
apache2	14	11	35	7	76,800	1.11
tmt_sym	14	10	17	14	32,000	1.04
ecology2	18	12	18	45	33,261	1.03

s3rmt3m3 and parabolic_fem which are the two extremes from the previous section in terms of grid complexity. Iteration counts are reported in Table 4. One may notice that the number of outer iterations is exactly the same as in the fifth column of Table 2, meaning that the switch to an inexact second-level solver does not hinder the overall convergence. Also, the number of inner iterations is small, so our proposed preconditioner applied to the second-level operator is indeed robust. Eventually, as we decrease the number of subdomains for the second-level decomposition, the grid coarsening improves as well, especially for small-sized problem s3rmt3m3.

In another context, we use our proposed preconditioner to solve coarse systems yield by two other multilevel preconditioners. The following three-dimensional problems are discretized by FreeFEM [19] using 4,096 MPI processes. First, we use GenEO [44] to assemble a two-level analytic preconditioner for a scalar diffusion equation using order-two Lagrange finite elements. The number of unknowns is $4.17 \cdot 10^6$, and the second-level operator generated by GenEO is of dimension $n_{C,2} =$ 60,144. It is redistributed among 512 processes, and our preconditioner constructs a third-level operator of dimension $n_{C,3} =$ 12,040. Then, we use GAMG to assemble a four-level quasi-algebraic (the near-nullspace is provided by the discretization kernel) preconditioner for the system of linear elasticity using order-two Lagrange finite elements. The number of unknowns is $3.06 \cdot 10^7$. The coarse operator from GAMG grid hierarchy is of dimension $n_{C,2} =$ 14,880. It is redistributed among 256 processes using the telescope infrastructure [35] and our preconditioner constructs a final-level

TABLE 3
Preconditioner cost (in seconds): singular value decomposition, generalized eigenvalue problem, and Krylov solver.

Identifier	SVDSolve	EPSSolve	KSPSolve
s3rmt3m3	1.3	$2.7 \cdot 10^{-3}$	4.1
vanbody	$5.2 \cdot 10^{-1}$	$9.5 \cdot 10^{-2}$	3.6
gridgena	1.1	$3.6 \cdot 10^{-2}$	$6.1 \cdot 10^{-1}$
ct20stif	1.4	$2.9 \cdot 10^{-1}$	12.1
nasasrb	1.3	$2.1 \cdot 10^{-1}$	4.2
Dubcova2	$2.8 \cdot 10^{-1}$	$6.9 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$
finan512	$1.4 \cdot 10^{-1}$	$2.2 \cdot 10^{-2}$	$2.6 \cdot 10^{-1}$
consph	14.4	4.5	10.7
s3dkt3m2	$4.8 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	2.7
shipsec8	8.3	8.1	27.2
ship_003	20.5	13.6	27.5
boneS01	9.6	3.4	7.8
bmwcra_1	54.7	33.8	17.9
G2_circuit	1.1	$2.7 \cdot 10^{-1}$	1.9
pwtk	13.5	4.9	2.2
offshore	66.2	39.1	32.9
af_4_k101	75.7	48.7	6.9
parabolic_fem	42.0	7.8	$2.9 \cdot 10^{-1}$
apache2	211.3	96.2	8.5
tmt_sym	121.1	28.8	2.2
ecology2	204.8	53.0	2.9



(a) Scalar diffusion in the unit cube with the coefficient κ extruded in one dimension.

(b) Elongated (10× ratio) three-dimensional beam with Young's modulus (E) and Poisson's ratio (ν) extruded in one dimension.

FIG. 2. *Variations of the material coefficients for problems from Table 5.*

operator of dimension $n_{C,3} = 5{,}120$. This is similar, in principle, to AMG-DD [8]. Unlike what is traditionally done with smoothed-aggregation AMG [50], we do not transfer explicitly the near-nullspace from GAMG coarse level for setting up our preconditioner. These results are gathered in Table 5. Again, one may notice that the fast and accurate convergence of the inner solves (third column) does not hinder the overall convergence (second column). For both the scalar diffusion equation $\nabla \cdot \kappa \nabla$ and the system of linear elasticity, highly heterogeneous material coefficients are used, see Figure 2a and Figure 2b, respectively.

Table 4

Algebraic multilevel preconditioner: Outer iterations is the FGMRES iteration count, Inner iterations is the average GMRES iteration count to solve coarse systems, n is the size of the linear system, $n_{C,2}$ (resp. $n_{C,3}$) is the size of the second-level (resp. third-level) operator, GC is the grid complexity.

Identifier	Outer iterations	Inner iterations	n	$n_{C,2}$	$n_{C,3}$	GC
s3rmt3m3	4	10	5,357	5,321	2,240	2.41
parabolic_fem	17	3	525,825	21,736	3,838	1.05

Table 5

Hybrid multilevel preconditioner: Outer iterations is the FGMRES iteration count, Inner iterations is the average GMRES iteration count to solve coarse systems, n is the size of the linear system, $n_{C,2}$ is the size of the coarse-level operator assembled by either GenEO (for problem diffusion) or GAMG (for problem elasticity), $n_{C,3}$ is the size of the second-level operator assembled by our algebraic preconditioner to solve the aforementioned coarse systems.

Identifier	Outer iterations	Inner iterations	n	$n_{C,2}$	$n_{C,3}$
diffusion	11	5	4,173,281	60,144	12,040
elasticity	8	11	30,633,603	14,880	5,120

Furthermore, as in subsection 4.1, note that using a simple one-level preconditioner such as M^{-1}_{RAS} with a minimal overlap of one for solving coarse systems from Tables 4 and 5 does not yield accurate enough inner solutions, thus preventing the outer solvers from converging. Coupling GAMG with our preconditioner is a good assessment of the composability of PETSc solvers [9], for the interested reader, we provide next in Figure 3 the exact options used to setup such a multilevel solver.

4.3. Strong scalability. We present here a strong scalability experiment of the proposed preconditioner to assess the computational complexity discussed in subsection 3.3. We consider the problem tmt_sym, see Table 1. Starting with $N = 128$ subdomains, we increase N by a factor of 2 up to 2,048 and report in Figure 4 the runtime on processor 0, as reported by PETSc -log_view option. We also plot the ideal linear decrease with slope -3. The runtime is comprised of both setup and solve times. Setup time includes dense SVD computation, generalized eigenvalue problem solve, computing and factoring the coarse space matrix, and factoring the local one-level subdomain matrix. Solve time accounts for solving the linear system by using GMRES preconditioned by the two-level method M^{-1}_{deflated}. We notice that setup times (which include exact SVD computation) dominates. However, we observe the linear decrease in the setup and the overall runtime which behaves similarly to the ideal scaling. Indeed, as discussed in subsection 3.3, the local explicit SVD costs $O(n_i^3)$, and doubling the number of subdomains reduces n_i by a factor of approximately 2 which yields a cubic decrease in the SVD cost.

We would like to remind that the system of equations associated with this matrix can be solved by using GMRES preconditioned by BoomerAMG, and the corresponding runtime is around one second independently of the number of processors. Therefore, as long as solving the generalized eigenvalue problem in (3.3) requires computing the dense SVD, the proposed preconditioner should not be considered, from a performance point-of-view, when other preconditioners are effective.

```
-ksp_type           fgmres
-ksp_rtol           1.0e-8

-pc_type            gamg
-pc_gamg_threshold 0.01
-pc_gamg_repartition
-pc_mg_levels       4

-prefix_push mg_coarse_
 -pc_type telescope
 -prefix_push pc_telescope_
  -reduction_factor 16
 -prefix_pop
-prefix_pop
# continue on the right column
```

```
# continued from the left column
-prefix_push mg_coarse_telescope_
 -ksp_converged_reason
 -ksp_type      gmres
 -ksp_pc_side   right
 -ksp_norm_type unpreconditioned
 -ksp_rtol      1.0e-4

 -pc_type       hpddm
 -prefix_push pc_hpddm_
  -define_subdomains
  -levels_1_pc_type     asm # $M_{\text{RAS}}^{-1}$
  -levels_1_sub_pc_type cholesky # subdomain solvers
  -levels_1_eps_nev     20 # smallest $\lambda$ in (3.3)
  -levels_1_st_type     mat # $\widetilde{\underline{A}_{ii}}^{-1}$ from (3.4)
  -coarse_pc_type       cholesky # coarse solver
 -prefix_pop
-prefix_pop
```

Fig. 3. *PETSc command-line options for coupling GAMG and the proposed preconditioner.*

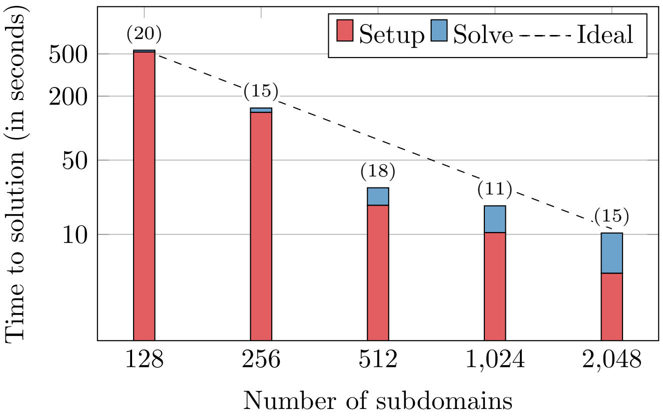


Fig. 4. *Strong scaling analysis of the proposed preconditioner for solving problem tmt_sym with an increasing number of MPI processes. Numbers in parentheses are iteration counts. For reference, this problem is solved in around one second with BoomerAMG, independently of the number of processes. The dashed line with a slope of -3 represents the ideal linear decrease, see subsection* 3.3.

5. Conclusion. We presented in this paper a fully algebraic and locally constructed multilevel overlapping Schwarz preconditioner that can bound from above the condition number of the preconditioned matrix given a user-defined number. The construction of the preconditioner relies on finding local SPSD splitting matrices of the matrix A. Computing these splitting matrices involves the computation of the right singular vectors of the local block row matrix which might be considered costly on the fine level. However, the locality of computations and the robustness of the preconditioner provide a very powerful and scalable preconditioner that can be used as a black-box solver especially when other black-box preconditioners fail to achieve a desired convergence rate. Our implementation is readily available in the PETSc library. Again, the proposed preconditioner is not meant to replace analytic multilevel preconditioners such as smoothed-aggregation algebraic multigrid and GenEO. When these work, they will be more efficient algorithmically. However, employing the proposed preconditioner to solve the corresponding coarse problems proved to be effective and efficient. As a future work, we would like to investigate less expensive constructions of SPSD matrices for specific classes of SPD matrices that arise from the discretization of PDEs.

Acknowledgments. This work was granted access to the GENCI-sponsored HPC resources of TGCC@CEA under allocation A0090607519. H. Al Daas was partially supported by EPSRC grant EP/W009676/1. The authors would like to thank J. E. Roman for interesting discussions concerning the solution of (3.3) and S. Zampini for helping us improve the quality of the article. The authors are grateful to the two reviewers for their constructive comments and recommendation that helped us improve the clarity and the presentation of the manuscript.

REFERENCES

[1] M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos, *Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom*, in Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, SC04, IEEE Computer Society, 2004, pp. 34:1–34:15.

[2] H. Al Daas and L. Grigori, *A class of efficient locally constructed preconditioners based on coarse spaces*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 66–91.

[3] H. Al Daas, L. Grigori, P. Jolivet, and P.-H. Tournier, *A multilevel Schwarz preconditioner based on a hierarchy of robust coarse spaces*, SIAM Journal on Scientific Computing, 43 (2021), pp. A1907–A1928.

[4] H. Al Daas, P. Jolivet, and J. A. Scott, *A robust algebraic domain decomposition preconditioner for sparse normal equations*, SIAM Journal on Scientific Computing, (2022), pp. A1047–A1068.

[5] H. Al Daas, T. Rees, and J. A. Scott, *Two-level Nyström–Schur preconditioner for sparse symmetric positive definite matrices*, SIAM Journal on Scientific Computing, 43 (2021), pp. A3837–A3861.

[6] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK users' guide*, Society for Industrial and Applied Mathematics, 1999.

[7] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang, *PETSc web page*, 2021, https://petsc.org.

[8] R. Bank, R. D. Falgout, T. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid domain and range decomposition (AMG-DD/AMG-RD)*, SIAM Journal on Scientific Computing, 37 (2015), pp. S113–S136.

[9] J. Brown, M. G. Knepley, D. A. May, L. C. McInnes, and B. F. Smith, *Composable linear solvers for multiphysics*, in 2012 11th International Symposium on Parallel and Distributed Computing, 2012, pp. 55–62.

[10] X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 792–797.

[11] T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. A. Manteuffel, S. F. McCormick, J. W. Ruge, and P. S. Vassilevski, *Spectral AMGe (ρAMGe)*, SIAM Journal on Scientific Computing, 25 (2003), pp. 1–26.

[12] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Transactions on Mathematical Software, 38 (2011), pp. 1–28.

[13] V. Dolean, P. Jolivet, and F. Nataf, *An introduction to domain decomposition methods. Algorithms, theory, and parallel implementation*, Society for Industrial and Applied Mathematics, 2015.

[14] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Oxford University Press, 2017.

[15] R. D. Falgout, *An introduction to algebraic multigrid*, Tech. Report UCRL-JRNL-220851,

Lawrence Livermore National Laboratory, California, United States, 2006.

[16] R. D. Falgout and U. M. Yang, hypre*: a library of high performance preconditioners*, Computational Science—ICCS 2002, (2002), pp. 632–641.

[17] M. J. Gander and A. Loneland, *SHEM: an optimal coarse space for RAS and its multiscale approximation*, in Domain Decomposition Methods in Science and Engineering XXIII, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., Cham, 2017, Springer International Publishing, pp. 313–321.

[18] L. Gouarin and N. Spillane, *Fully algebraic domain decomposition preconditioners with adaptive spectral bounds*. Preprint, June 2021, https://hal.archives-ouvertes.fr/hal-03258644.

[19] F. Hecht, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012), pp. 251–265.

[20] A. Heinlein, C. Hochmuth, and A. Klawonn, *Reduced dimension GDSW coarse spaces for monolithic Schwarz domain decomposition methods for incompressible fluid flow problems*, International Journal for Numerical Methods in Engineering, 121 (2020), pp. 1101–1119.

[21] A. Heinlein, A. Klawonn, J. Knepper, O. Rheinbach, and O. B. Widlund, *Adaptive GDSW coarse spaces of reduced dimension for overlapping Schwarz methods*, technical report, Universität zu Köln, September 2020, https://kups.ub.uni-koeln.de/12113/.

[22] V. Hernandez, J. E. Roman, and V. Vidal, *SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems*, ACM Transactions on Mathematical Software, 31 (2005), pp. 351–362, https://slepc.upv.es.

[23] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems.*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.

[24] N. J. Higham and T. Mary, *A new preconditioner that exploits low-rank approximations to factorization error*, SIAM Journal on Scientific Computing, 41 (2019), pp. A59–A82.

[25] P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme, *Scalable domain decomposition preconditioners for heterogeneous elliptic problems*, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '13, ACM, 2013, pp. 80:1–80:11.

[26] P. Jolivet, J. E. Roman, and S. Zampini, *KSPHPDDM and PCHPDDM: extending PETSc with advanced Krylov methods and robust multilevel overlapping Schwarz preconditioners*, Computers & Mathematics with Applications, 84 (2021), pp. 277–295.

[27] T. B. Jönsthövel, M. B. van Gijzen, C. Vuik, C. Kasbergen, and A. Scarpas, *Preconditioned conjugate gradient method enhanced by deflation of rigid body modes applied to composite materials*, Computer Modeling in Engineering & Sciences, 47 (2009), pp. 97–118.

[28] T. B. Jönsthövel, M. B. van Gijzen, C. Vuik, and A. Scarpas, *On the use of rigid body modes in the deflated preconditioned conjugate gradient method*, SIAM Journal on Scientific Computing, 35 (2013), pp. B207–B225.

[29] G. Karypis and V. Kumar, *Multilevel k-way partitioning scheme for irregular graphs*, Journal of Parallel and Distributed computing, 48 (1998), pp. 96–129.

[30] A. Klawonn, M. Kühn, and O. Rheinbach, *Adaptive coarse spaces for FETI-DP in three dimensions*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2880–A2911.

[31] A. Klawonn, P. Radtke, and O. Rheinbach, *FETI-DP methods with an adaptive coarse space*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 297–320.

[32] F. Kong and X.-C. Cai, *A scalable nonlinear fluid–structure interaction solver based on a Schwarz preconditioner with isogeometric unstructured coarse spaces in 3D*, Journal of Computational Physics, 340 (2017), pp. 498–518.

[33] R. Li, Y. Xi, and Y. Saad, *Schur complement-based domain decomposition preconditioners with low-rank corrections*, Numerical Linear Algebra with Applications, 23 (2016), pp. 706–729.

[34] P. Marchand, X. Claeys, P. Jolivet, F. Nataf, and P.-H. Tournier, *Two-level preconditioning for h-version boundary element approximation of hypersingular operator with GenEO*, Numerische Mathematik, 146 (2020), pp. 597–628.

[35] D. A. May, P. Sanan, K. Rupp, M. G. Knepley, and B. F. Smith, *Extreme-scale multigrid components within PETSc*, in Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '16, New York, NY, USA, 2016, Association for Computing Machinery.

[36] A. Napov and Y. Notay, *An algebraic multigrid method with guaranteed convergence rate*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1079–A1109.

[37] Y. Notay, *An aggregation-based algebraic multigrid method*, Electronic Transactions on Numerical Analysis, 37 (2010), pp. 123–146, http://agmg.eu.

[38] Y. Notay and P. S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numerical Linear Algebra with Applications, 15 (2008), pp. 473–487.

[39] J. W. Pearson and J. Pestana, *Preconditioners for Krylov subspace methods: an overview*, GAMM-Mitteilungen, 43 (2020), p. e202000015.

[40] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific Computing, 14 (1993), pp. 461–469.

[41] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, 2003.

[42] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[43] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.

[44] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl, *Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps*, Numerische Mathematik, 126 (2014), pp. 741–770.

[45] N. Spillane and D. Rixen, *Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms*, International Journal for Numerical Methods in Engineering, 95 (2013), pp. 953–990.

[46] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 601–614.

[47] R. Tamstorf, T. Jones, and S. F. McCormick, *Smoothed aggregation multigrid for cloth simulation*, ACM Transactions on Graphics, 34 (2015).

[48] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, *Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods*, Journal of Scientific Computing, 39 (2009), pp. 340–370.

[49] J. Van lent, R. Scheichl, and I. G. Graham, *Energy-minimizing coarse spaces for two-level Schwarz methods for multiscale PDEs*, Numerical Linear Algebra with Applications, 16 (2009), pp. 775–799.

[50] P. Vaněk, *Acceleration of convergence of a two-level algorithm by smoothing transfer operators*, Applications of Mathematics, 37 (1992), pp. 265–274.

[51] L. Wu, E. Romero, and A. Stathopoulos, *PRIMME_SVDS: a high-performance preconditioned SVD solver for accurate large-scale computations*, SIAM Journal on Scientific Computing, 39 (2017), pp. S248–S271.

[52] S. Zampini, *PCBDDC: a class of robust dual-primal methods in PETSc*, SIAM Journal on Scientific Computing, 38 (2016), pp. S282–S306.