



Technical Report Series Center for Data and Simulation Science

Axel Klawonn, Martin Lanser, Oliver Rheinbach, Janine Weber

Preconditioning the coarse problem of BDDC methods - Three-level, algebraic multigrid, and vertex-based preconditioners

Technical Report ID: CDS-2019-14

Available at <https://kups.ub.uni-koeln.de/id/eprint/9713>

Submitted on June 18, 2019

PRECONDITIONING THE COARSE PROBLEM OF BDDC METHODS - THREE-LEVEL, ALGEBRAIC MULTIGRID, AND VERTEX-BASED PRECONDITIONERS*

AXEL KLAWONN^{†‡}, MARTIN LANSER^{†‡}, OLIVER RHEINBACH[§], AND JANINE WEBER[†]

June 17, 2019

Abstract. A fair comparison of three Balancing Domain Decomposition by Constraints (BDDC) methods with an approximate coarse space solver is attempted for the first time. The comparison is made for a BDDC method with an algebraic multigrid preconditioner for the coarse problem, a three-level BDDC method, and a BDDC method with a vertex-based coarse preconditioner which was recently introduced by Clark Dohrmann, Kendall Pierson, and Olof Widlund. For the first time, all methods are presented and discussed in a common framework. Condition number bounds are provided for all approaches. All methods are implemented in a common highly parallel scalable BDDC software package based on PETSc, to allow for a fair comparison. Numerical results showing the parallel scalability are presented for the equations of linear elasticity. For the first time, this includes parallel scalability tests for the vertex-based approximate BDDC method.

Key words. approximate BDDC, three-level BDDC, multilevel BDDC, vertex-based BDDC

AMS subject classifications.

1. Introduction. During the last decade, approximate variants of the BDDC (Balancing Domain Decomposition by Constraints) and FETI-DP (Finite Element Tearing and Interconnecting - Dual-Primal) methods became popular for the solution of various linear and nonlinear partial differential equations [19, 8, 24, 23, 1, 16, 18, 13, 15, 9]. These methods differ from their exact relatives by an approximate solution of components of the preconditioner, most notably the coarse problem. An approximate solution of the coarse problem can reduce the numerical robustness slightly, but can increase the scalability of the method significantly. While multilevel BDDC, see [24, 23, 20, 21] and, recently, [1], is constructed by applying exact BDDC recursively to the coarse problem, in other approximate BDDC variants cycles of AMG (algebraic multigrid) are applied to the coarse problem; see, e.g., [19, 8, 14]. Recently, vertex-based coarse spaces of reduced size have been suggested to approximate the original coarse system [9].

In [14], we already considered, in a common framework, several linear and nonlinear BDDC variants using AMG-based approximations, following the BDDC formulation from [19] for linear problems. We also compared their performance using our ultra scalable PETSc-based [4, 5, 6] BDDC implementation, applying BoomerAMG [11] for all AMG solves. In the current paper, we continue these efforts and include the aforementioned vertex-based BDDC as well as three-level and multilevel BDDC in our framework as well as in our software package. In addition to a description of all

*This work was supported in part by Deutsche Forschungsgemeinschaft (DFG) through the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA) under grants KL 2094/4-2 and RH 122/3-2.

[†]Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de, janine.weber@uni-koeln.de, url: <http://www.numerik.uni-koeln.de>

[‡]Center for Data and Simulation Science, University of Cologne, Germany, url: <http://www.cds.uni-koeln.de>

[§]Fakultät für Mathematik und Informatik, Institut für Numerische Mathematik und Optimierung, Technische Universität Bergakademie Freiberg, 09596 Freiberg, Germany, oliver.rheinbach@math.tu-freiberg.de

methods and their condition number bounds, we also include a numerical and parallel comparison. To the best of our knowledge, a comparison between three-level BDDC and BDDC with AMG-based coarse approximations, using implementations based on the same building blocks, to allow for a fair comparison, has not been considered before. Also, for the first time, parallel scalability tests for the vertex-based BDDC method [9] are presented.

As a common baseline in all our comparisons, we include the approximate AMG-based preconditioner which performed best in [14]. This specific variant is also related, but not identical to three preconditioners suggested in [8]. This was already discussed in [14] in detail.

The remainder of this paper is organized as follows: In section 2, we introduce the model problem, outline the domain decomposition approach, and present an exact BDDC preconditioner for the globally assembled system. In sections 3 and 4, we describe three different approximate BDDC preconditioners in a common framework. Namely, we consider an approximate BDDC preconditioner using AMG, a three-level BDDC method, and a vertex-based BDDC preconditioner using a Gauss-Seidel method. Section 5 gives the theory and the condition number bounds for all three aforementioned approximate BDDC preconditioners. In section 6, we provide some details of our parallel implementation. In particular, we implemented all three approximate preconditioners with the same building blocks, which allows us to directly compare the methods with each other regarding their computing time and parallel scalability. Finally, in section 7, we present comparing results in three spatial dimensions. For all our numerical tests, we consider linear elasticity problems.

2. Exact BDDC preconditioner and model problem.

2.1. Linear elasticity and finite elements. We consider an elastic domain $\Omega \subset \mathbb{R}^3$. We denote with $u : \Omega \rightarrow \mathbb{R}^3$ the displacement of the domain, with f a given volume force, and with g a given surface force onto the domain, respectively. In particular, we assume that one part of the boundary of the domain, $\partial\Omega_D$, is clamped, i.e., has homogeneous Dirichlet boundary conditions, and that the rest, $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$, is subject to the surface force g , i.e., a natural boundary condition.

With $\mathbf{H}^1(\Omega) := (H^1(\Omega))^3$, the appropriate space for a variational formulation is the Sobolev space $\mathbf{H}_0^1(\Omega, \partial\Omega_D) := \{\mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v} = 0 \text{ on } \partial\Omega_D\}$. The problem of linear elasticity then consists in finding the displacement $\mathbf{u} \in \mathbf{H}_0^1(\Omega, \partial\Omega_D)$, such that

$$(2.1) \quad \int_{\Omega} G(\mathbf{x}) \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) \, d\mathbf{x} + \int_{\Omega} G(\mathbf{x}) \beta(\mathbf{x}) \operatorname{div} \mathbf{u} \operatorname{div} \mathbf{v} \, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle,$$

for all $\mathbf{v} \in \mathbf{H}_0^1(\Omega, \partial\Omega_D)$ for given material parameters G and β and the right-hand side

$$\langle \mathbf{F}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{f}^T \mathbf{v} \, d\mathbf{x} + \int_{\partial\Omega_N} \mathbf{g}^T \mathbf{v} \, d\sigma.$$

The material parameters G and β depend on the Young modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2)$ by $G = E/(1 + \nu)$ and $\beta = \nu/(1 - 2\nu)$. Furthermore, the linearized strain tensor $\varepsilon = (\varepsilon_{ij})_{ij}$ is defined by $\varepsilon_{ij}(\mathbf{u}) := \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$, and we use the notation

$$\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) := \sum_{i,j=1}^3 \varepsilon_{ij}(\mathbf{u}) \varepsilon_{ij}(\mathbf{v}) \quad \text{and} \quad (\varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} := \int_{\Omega} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) \, d\mathbf{x}.$$

The corresponding bilinear form associated with linear elasticity can then be written as

$$a(\mathbf{u}, \mathbf{v}) = (G \varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} + (G\beta \operatorname{div} \mathbf{u}, \operatorname{div} \mathbf{v})_{L_2(\Omega)}.$$

We discretize our elliptic problem of linear elasticity by low order, conforming finite elements and thus obtain the linear system of equations

$$(2.2) \quad K_g u = f_g.$$

2.2. Exact BDDC preconditioner for the assembled system. The exact BDDC preconditioner formulation from [19] is applied directly to the system (2.2).

Given is a nonoverlapping domain decomposition Ω_i , $i = 1, \dots, N$, of Ω such that $\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i$. Each subdomain Ω_i is a union of finite elements, W_i , $i = 1, \dots, N$, are the local finite element spaces, and the product space is defined by $W = W_1 \times \dots \times W_N$. The global finite element space V^h corresponds to the discretization of Ω and we assume to have an assembly operator R^T , where $R^T : W \rightarrow V^h$. By discretization of the given partial differential equation restricted to Ω_i , we obtain a set of local problems

$$K_i u_i = f_i, \quad i = 1, \dots, N.$$

Defining the block operators

$$K = \begin{pmatrix} K_1 & & \\ & \ddots & \\ & & K_N \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix},$$

we can write $K_g := R^T K R$ and $f_g := R^T f$. Finally, the interface between the subdomains is defined as $\Gamma := \bigcup_{i=1}^N \partial\Omega_i \setminus \partial\Omega$.

We use the index Γ for degrees of freedom on Γ and for the remaining degrees of freedom despite the Dirichlet boundary $\partial\Omega_D$, we use the index I . For the construction of a BDDC preconditioner directly applicable to the assembled linear system $K_g u = f_g$, the interface Γ is split into primal (Π) and the remaining dual (Δ) degrees of freedom. Usually, vertices are chosen as primal variables and the coarse space is augmented by averages over edges and/or faces.

Let us introduce the space $\tilde{W} \subset W$ of functions, which are continuous in all primal variables, and the assembly operators \tilde{R}^T and \check{R}^T with $\tilde{R}^T : W \rightarrow \tilde{W}$ and $\check{R}^T : \tilde{W} \rightarrow V^h$. Using \check{R} , we can form the partially assembled system

$$(2.3) \quad \tilde{K} := \check{R}^T K \check{R}$$

and can also obtain the globally assembled finite element matrix K_g from \tilde{K} by

$$(2.4) \quad K_g = \tilde{R}^T \tilde{K} \tilde{R}.$$

We denote the interior and interface variables with the indices I and Γ , respectively. Ordering the interior variables first and the interface variables last, we obtain

$$(2.5) \quad \tilde{K} = \begin{pmatrix} K_{II} & \tilde{K}_{\Gamma I}^T \\ \tilde{K}_{\Gamma I} & \tilde{K}_{\Gamma\Gamma} \end{pmatrix}.$$

The matrix K_{II} is block-diagonal and applications of K_{II}^{-1} only require local solves on the interior parts of the subdomains and are thus easily parallelizable. We further introduce the union of subdomain interior (I) and dual (Δ) interface degrees of freedom as an extra set of degrees of freedom denoted by the index B . The index B thus leads to an alternative representation of the partially assembled system \tilde{K} as

$$(2.6) \quad \tilde{K} = \begin{pmatrix} K_{BB} & \tilde{K}_{\Pi B}^T \\ \tilde{K}_{\Pi B} & \tilde{K}_{\Pi\Pi} \end{pmatrix}.$$

Like K_{II} , the matrix K_{BB} is a block-diagonal matrix and applications of K_{BB}^{-1} only require local solves.

Adding usual scalings, e.g., ρ -scaling [17] or deluxe-scaling [7], to the prolongation operators and thus defining $\tilde{R}_D : V^h \rightarrow \tilde{W}$, we obtain the BDDC preconditioner for K_g by

$$(2.7) \quad M_{\text{BDDC}}^{-1} := \left(\tilde{R}_D^T - \mathcal{H} P_D \right) \tilde{K}^{-1} \left(\tilde{R}_D - P_D^T \mathcal{H}^T \right);$$

see [19]. Here, the operator $\mathcal{H} : \tilde{W} \rightarrow V^h$ is the discrete harmonic extension to the interior of the subdomains given by

$$(2.8) \quad \mathcal{H} := \begin{pmatrix} 0 & -(K_{II})^{-1} \tilde{K}_{\Gamma I}^T \\ 0 & 0 \end{pmatrix}.$$

Finally, let $P_D : \tilde{W} \rightarrow \tilde{W}$ be a scaled jump operator defined by

$$(2.9) \quad P_D = I - E_D := I - \tilde{R} \tilde{R}_D^T.$$

The original definition often used in the literature is $P_D := B_D^T B$; see [22, Chapter 6] and [19] for more details. There, B is the jump matrix used in FETI-type methods. Please note that in the standard definition, the BDDC preconditioner is formulated for the reduced interface problem, i.e., as

$$(2.10) \quad M_{\text{BDDC-}\Gamma}^{-1} S_{\Gamma\Gamma} := \tilde{R}_{D,\Gamma}^T \tilde{S}_{\Gamma\Gamma}^{-1} \tilde{R}_{D,\Gamma} S_{\Gamma\Gamma}.$$

Here, the prolongation operator $\tilde{R}_{D,\Gamma}$ is formed in the same way as \tilde{R}_D only restricted to the interface variables Γ , and $S_{\Gamma\Gamma}$ and $\tilde{S}_{\Gamma\Gamma}$ are the subdomain interface Schur complements of the matrices K_g and \tilde{K} , respectively. Let us remark that the preconditioned system $M_{\text{BDDC}}^{-1} K_g$ has, except for some eigenvalues equal to 1, the same spectrum as the standard BDDC preconditioner formulated on the Schur complement; see [19, Theorem 1]. Here, we provide a related but slightly more direct proof: We first explicitly write the BDDC preconditioner M_{BDDC}^{-1} as

$$\begin{aligned} M_{\text{BDDC}}^{-1} &:= \left(\tilde{R}_D^T - \mathcal{H} P_D \right) \tilde{K}^{-1} \left(\tilde{R}_D - P_D^T \mathcal{H}^T \right) \\ &= \begin{pmatrix} I & K_{II}^{-1} \tilde{K}_{\Gamma I}^T (I - \tilde{R}_\Gamma \tilde{R}_{D,\Gamma}^T) \\ 0 & \tilde{R}_{D,\Gamma}^T \end{pmatrix} \tilde{K}^{-1} \begin{pmatrix} I & 0 \\ (I - \tilde{R}_{D,\Gamma} \tilde{R}_\Gamma^T) \tilde{K}_{\Gamma I} K_{II}^{-1} & \tilde{R}_{D,\Gamma} \end{pmatrix} \\ &= \begin{pmatrix} I & K_{II}^{-1} \tilde{K}_{\Gamma I}^T (I - E_{D,\Gamma}) \\ 0 & \tilde{R}_{D,\Gamma}^T \end{pmatrix} \tilde{K}^{-1} \begin{pmatrix} I & 0 \\ (I - E_{D,\Gamma}^T) \tilde{K}_{\Gamma I} K_{II}^{-1} & \tilde{R}_{D,\Gamma} \end{pmatrix}. \end{aligned}$$

Using the block factorization

$$\tilde{K}^{-1} = \begin{pmatrix} I & -K_{II}^{-1}\tilde{K}_{\Gamma I}^T \\ 0 & I \end{pmatrix} \begin{pmatrix} K_{II}^{-1} & 0 \\ 0 & \tilde{S}_{\Gamma\Gamma}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -\tilde{K}_{\Gamma I}K_{II}^{-1} & I \end{pmatrix},$$

by a direct computation we obtain the alternative representation

$$M_{\text{BDDC}}^{-1} = \begin{pmatrix} K_{II}^{-1} + K_{II}^{-1}\tilde{K}_{\Gamma I}^T E_{D,\Gamma} \tilde{S}_{\Gamma\Gamma}^{-1} E_{D,\Gamma}^T \tilde{K}_{\Gamma I} K_{II}^{-1} & -K_{II}^{-1}\tilde{K}_{\Gamma I}^T E_{D,\Gamma} \tilde{S}_{\Gamma\Gamma}^{-1} \tilde{R}_{D,\Gamma} \\ -\tilde{R}_{D,\Gamma}^T \tilde{S}_{\Gamma\Gamma}^{-1} E_{D,\Gamma}^T \tilde{K}_{\Gamma I} K_{II}^{-1} & \tilde{R}_{D,\Gamma}^T \tilde{S}_{\Gamma\Gamma}^{-1} \tilde{R}_{D,\Gamma} \end{pmatrix}.$$

The multiplication $M_{\text{BDDC}}^{-1}K_g$ finally yields

$$M_{\text{BDDC}}^{-1}K_g = \begin{pmatrix} I & U \\ 0 & M_{\text{BDDC}-\Gamma}^{-1}S_{\Gamma\Gamma} \end{pmatrix}$$

with $U = K_{II}^{-1}K_{\Gamma I} - K_{II}^{-1}K_{\Gamma I}^T \tilde{R}_{D,\Gamma}^T \tilde{S}_{\Gamma\Gamma}^{-1} \tilde{R}_{D,\Gamma} S_{\Gamma\Gamma}$, with $E_{D,\Gamma} = \tilde{R}_{\Gamma} \tilde{R}_{D,\Gamma}^T$, and using $K_{\Gamma I} = \tilde{R}_{\Gamma}^T \tilde{K}_{\Gamma I}$. Here, $M_{\text{BDDC}-\Gamma}^{-1}$ is the classical BDDC preconditioner for the Schur complement; see (2.10). The result then follows from the fact, that the eigenvalues of a block-triangular matrix equal the union of the set of eigenvalues of the diagonal blocks.

3. Approximate BDDC Preconditioners. All approximate BDDC methods considered in this paper are based on an approximate solution of the coarse problem of BDDC. To ensure a simple and fair comparison, all approximate preconditioners are implemented using the same software framework; see also [13, 14].

By block factorization, we obtain

$$(3.1) \quad \tilde{K}^{-1} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -K_{BB}^{-1}\tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \tilde{S}_{\Pi\Pi}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B}K_{BB}^{-1} & I \end{pmatrix},$$

where $\tilde{S}_{\Pi\Pi}$ is the Schur complement

$$\tilde{S}_{\Pi\Pi} = \tilde{K}_{\Pi\Pi} - \tilde{K}_{\Pi B} K_{BB}^{-1} \tilde{K}_{\Pi B}^T.$$

Note that $\tilde{S}_{\Pi\Pi}$ represents the BDDC coarse operator. Replacing $\tilde{S}_{\Pi\Pi}^{-1}$ by an approximation $\hat{S}_{\Pi\Pi}^{-1}$ in (3.1), we obtain an approximation for \tilde{K}^{-1} by

$$(3.2) \quad \hat{K}^{-1} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -K_{BB}^{-1}\tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \hat{S}_{\Pi\Pi}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B}K_{BB}^{-1} & I \end{pmatrix}.$$

Replacing \tilde{K}^{-1} in (2.7) by \hat{K}^{-1} , we define an approximation to the BDDC preconditioner, i.e.,

$$(3.3) \quad \hat{M}^{-1} := \left(\tilde{R}_D^T - \mathcal{H}P_D \right) \hat{K}^{-1} \left(\tilde{R}_D - P_D^T \mathcal{H}^T \right).$$

For the remainder of the article, all approximate BDDC preconditioners are marked with a hat. In the following sections, we compare three different approaches to form $\hat{S}_{\Pi\Pi}^{-1}$, e.g., for the approximation of the coarse solve:

- a) using AMG (algebraic multigrid) denoted by $\hat{M}_{\text{BDDC,AMG}}^{-1}$;
- b) using exact BDDC recursively denoted by $\hat{M}_{\text{BDDC,3L}}^{-1}$
- c) using an exact solution of a smaller vertex-based coarse space denoted by $\hat{M}_{\text{BDDC,VB}}^{-1}$.

Let us remark that $\hat{M}_{\text{BDDC,AMG}}^{-1}$ was denoted \hat{M}_3^{-1} in [14].

4. Examples of approximate BDDC preconditioners. In this section, we give three examples of approximate BDDC preconditioners presented in the notation introduced in [section 3](#). First, we consider an approximate BDDC preconditioner using AMG to precondition $\tilde{S}_{\Pi\Pi}$, second, a three-level BDDC method using BDDC itself to precondition $\tilde{S}_{\Pi\Pi}$, and third, a vertex-based BDDC preconditioner using a Jacobi/Gauss-Seidel method in combination with a vertex-based coarse space to precondition $\tilde{S}_{\Pi\Pi}$.

4.1. BDDC Preconditioner with AMG coarse preconditioner. Let us denote the application of a fixed number of V-cycles of an AMG method to $\tilde{S}_{\Pi\Pi}$ by M_{AMG}^{-1} . By choosing M_{AMG}^{-1} in [\(3.2\)](#) as an approximation of $\tilde{S}_{\Pi\Pi}$, i.e., by choosing $\hat{S}_{\Pi\Pi}^{-1} := M_{\text{AMG}}^{-1}$, we obtain

$$(4.1) \quad \hat{K}_{\text{AMG}}^{-1} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} M_{\text{AMG}}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B} K_{BB}^{-1} & I \end{pmatrix}.$$

Again, by using $\hat{K}_{\text{AMG}}^{-1}$ as an approximation for \tilde{K}^{-1} in [\(3.3\)](#), we obtain the inexact reduced preconditioner $\hat{M}_{\text{BDDC,AMG}}^{-1}$.

4.2. A Three-level BDDC. Alternatively, if we construct an exact BDDC preconditioner $\hat{S}_{\Pi\Pi}^{-1}$ for the Schur complement matrix $\tilde{S}_{\Pi\Pi}$, [\(3.3\)](#) will become a three-level BDDC preconditioner $\hat{M}_{\text{BDDC,3L}}^{-1}$. This approach is equivalent to the three-level preconditioner introduced in [\[23\]](#), but formulated for the original matrix K_g . In [\[23\]](#), the BDDC formulation for the Schur complement system on the interface is used and applied recursively. Since we use the BDDC formulation for the complete system matrix K_g , we consequently apply this approach to form the third level. We thus follow [Section 2.2](#) and mark all operators and spaces defined for the third level with bars, e.g., \bar{I} are the interior variables on the third level, while I are those on the second level. In [Section 5](#), we derive the same condition number bound as in [\[24, 23\]](#).

Let us now describe the application of BDDC to $\tilde{S}_{\Pi\Pi}$ in some more details. The basic idea of the three-level BDDC preconditioner is to recursively introduce a further level of the decomposition of the domain Ω into \bar{N} subregions $\Omega^1, \dots, \Omega^{\bar{N}}$. Each subregion comprises a given number of subdomains. All primal variables Π on the subdomain level are then again partitioned into interior, primal, and dual variables, i.e., \bar{I} , $\bar{\Pi}$, and $\bar{\Delta}$, with respect to the subregions; see also [Figure 1](#) for a possible selection in 2D. Now, in principle, the subdomains take over the role of finite elements on the third level and the subregions the role of the subdomains. The basis functions of the third level are the coarse basis functions of the second level, localized to the subregions.

We therefore first define the space \bar{V}^h , which is spanned by all coarse basis functions of the second level and denote by \bar{W}_i , $i = 1, \dots, \bar{N}$ the spaces which are spanned by the restrictions of the coarse basis functions to the subregions Ω^i , $i = 1, \dots, \bar{N}$. The product space \bar{W} is now defined as $\bar{W} = \bar{W}_1 \times \dots \times \bar{W}_{\bar{N}}$.

Using local Schur complements $S_{\Pi\Pi}^{(i)} = K_{\Pi\Pi}^{(i)} - K_{\Pi B}^{(i)} K_{BB}^{(i)-1} K_{\Pi B}^{(i)T}$ on the subdomains and the block matrix $S_{\Pi\Pi} = \text{diag}(S_{\Pi\Pi}^{(1)}, \dots, S_{\Pi\Pi}^{(N)})$, we can redefine

$$\tilde{S}_{\Pi\Pi} = \sum_{i=1}^N R_{\Pi}^{(i)T} S_{\Pi\Pi}^{(i)} R_{\Pi}^{(i)},$$

where $R^T = (R^{(1)T}, \dots, R^{(N)T})$ and $R^{(i)} = \text{diag}(R_B^{(i)}, R_{\Pi}^{(i)})$, $i = 1, \dots, N$. Now we

can perform this assembly process only on the subregions, i.e.,

$$(4.2) \quad \bar{S}_j = \sum_{i=1}^{N_j} R_{\Pi}^{(i)T} S_{\Pi\Pi}^{(i)} R_{\Pi}^{(i)}, \quad \forall j = 1, \dots, \bar{N},$$

where N_j is the number of subdomains belonging to subregion Ω^j . Obviously, $\tilde{S}_{\Pi\Pi}$ takes over the role of K_g on the third level, while \bar{S}_j takes over the role of K_i . Consequently, defining a prolongation $\bar{R} : \bar{V}^h \rightarrow \bar{W}$, we can also write

$$\tilde{S}_{\Pi\Pi} = \bar{R}^T \bar{S} \bar{R},$$

with $\bar{S} = \text{diag}(\bar{S}_1, \dots, \bar{S}_{\bar{N}})$.

Let us introduce the space $\widetilde{\bar{W}} \subset \bar{W}$ of functions, which are continuous in all primal variables $\bar{\Pi}$ on the third level, and the assembly operators $\widetilde{\bar{R}}^T : \bar{W} \rightarrow \widetilde{\bar{W}}$ and $\widetilde{\bar{R}} : \widetilde{\bar{W}} \rightarrow \bar{V}^h$. Using $\widetilde{\bar{R}}$, we can form the partially assembled system

$$(4.3) \quad \widetilde{\bar{S}} := \widetilde{\bar{R}}^T \bar{S} \widetilde{\bar{R}}.$$

Adding scalings to the prolongations as before and thus defining $\widetilde{\bar{R}}_D : \bar{V}^h \rightarrow \widetilde{\bar{W}}$, we obtain the BDDC preconditioner for the third level by

$$(4.4) \quad \bar{M}_{\text{BDDC}}^{-1} := \left(\widetilde{\bar{R}}_D^T - \bar{\mathcal{H}} \bar{P}_D \right) \widetilde{\bar{S}}^{-1} \left(\widetilde{\bar{R}}_D - \bar{P}_D^T \bar{\mathcal{H}}^T \right).$$

The operator $\bar{\mathcal{H}} : \widetilde{\bar{W}} \rightarrow \bar{V}^h$ is the discrete harmonic extension to the interior of the subregions and writes

$$(4.5) \quad \bar{\mathcal{H}} := \begin{pmatrix} 0 & -(\bar{S}_{\bar{\Pi}\bar{\Pi}})^{-1} \widetilde{\bar{S}}_{\bar{\Gamma}\bar{\Pi}}^T \\ 0 & 0 \end{pmatrix},$$

with the blocks $\bar{S}_{\bar{\Pi}\bar{\Pi}}$ and $\widetilde{\bar{S}}_{\bar{\Gamma}\bar{\Pi}}$ of the partially assembled matrix

$$(4.6) \quad \widetilde{\bar{S}} = \begin{pmatrix} \bar{S}_{\bar{\Pi}\bar{\Pi}} & \widetilde{\bar{S}}_{\bar{\Gamma}\bar{\Pi}}^T \\ \widetilde{\bar{S}}_{\bar{\Gamma}\bar{\Pi}} & \widetilde{\bar{S}}_{\bar{\Gamma}\bar{\Gamma}} \end{pmatrix},$$

and the jump operator defined as $\bar{P}_D := I - \widetilde{\bar{R}} \widetilde{\bar{R}}_D^T$.

Now, by choosing $\hat{S}_{\Pi\Pi}^{-1} := \bar{M}_{\text{BDDC}}^{-1}$ as approximation for $\tilde{S}_{\Pi\Pi}^{-1}$ in (3.2), i.e., by

$$(4.7) \quad \hat{K}_{3L}^{-1} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \bar{M}_{\text{BDDC}}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B} K_{BB}^{-1} & I \end{pmatrix},$$

we can define

$$(4.8) \quad \widehat{M}_{\text{BDDC},3L}^{-1} := \left(\widetilde{\bar{R}}_D^T - \mathcal{H} P_D \right) \hat{K}_{3L}^{-1} \left(\widetilde{\bar{R}}_D - P_D^T \mathcal{H}^T \right).$$

Instead of inverting $\widetilde{\bar{S}}$ directly, we again can use a block factorization

$$(4.9) \quad \widetilde{\bar{S}}^{-1} = \begin{pmatrix} \bar{S}_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -\bar{S}_{BB}^{-1} \widetilde{\bar{S}}_{\Pi B}^T \\ I \end{pmatrix} \widetilde{\bar{T}}_{\Pi\Pi}^{-1} \begin{pmatrix} -\widetilde{\bar{S}}_{\Pi B} \bar{S}_{BB}^{-1} & I \end{pmatrix},$$

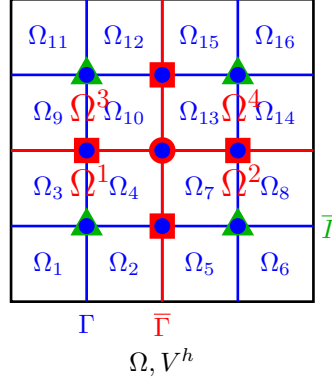


FIG. 1. Example of a domain decomposition in 2D in 16 subdomains and 4 subregions recursively. We mark in blue the interface Γ between subdomains and in red the interface $\bar{\Gamma}$ between subregions. Primal nodes $\bar{\Pi}$ w.r.t. the subregions are depicted as red circles, while primal nodes Π w.r.t. the subdomains are depicted as blue circles. Inner or dual nodes w.r.t. the subregions, i.e., $\bar{\Gamma}$ or, respectively, $\bar{\Delta}$, are depicted as green triangles or, respectively, red squares.

where the primal Schur complement on subregion level is

$$\tilde{T}_{\bar{\Pi}\bar{\Pi}} = \tilde{S}_{\bar{\Pi}\bar{\Pi}} - \tilde{S}_{\bar{\Pi}\bar{B}} \tilde{S}_{\bar{B}\bar{B}}^{-1} \tilde{S}_{\bar{B}\bar{\Pi}}^T.$$

Note that, following [19, Theorem 1], the preconditioned system $\bar{M}_{\text{BDDC}}^{-1} \tilde{S}_{\bar{\Pi}\bar{\Pi}}$ on the subregion level has the same eigenvalues as $\tilde{R}_{D,\bar{\Gamma}}^T \tilde{T}_{\bar{\Gamma}\bar{\Gamma}}^{-1} \tilde{R}_{D,\bar{\Gamma}} \tilde{T}_{\bar{\Gamma}\bar{\Gamma}}$ except for some eigenvalues equal to 1. Here, we have the Schur complement $\tilde{T}_{\bar{\Gamma}\bar{\Gamma}}$ of $\tilde{S}_{\bar{\Pi}\bar{\Pi}}$ on the interface of the subregions, the primally assembled Schur complement $\tilde{T}_{\bar{\Gamma}\bar{\Gamma}}$ of \tilde{S} on the interface of the subregions, and the splitting $\tilde{R}_D = \text{diag}(\tilde{I}_{\bar{\Gamma}}, \tilde{R}_{D,\bar{\Gamma}})$. Therefore, we can use the condition number estimations provided in [24, 23] analogously in Section 5.

4.3. Vertex-Based BDDC Preconditioner. We further describe the following vertex-based preconditioner for the coarse problem, as introduced by Dohrmann, Pierson, and Widlund [9], in our framework. We denote the respective vertex-based preconditioner with $\hat{M}_{\text{BDDC},\text{VB}}^{-1}$. Here, the preconditioner for the coarse problem can be interpreted as a standard two-level additive or multiplicative Schwarz approach. In particular, the direct solution of the coarse problem $\tilde{S}_{\bar{\Pi}\bar{\Pi}}^{-1}$ is replaced by a preconditioner M_{VB}^{-1} based on a smaller vertex-based coarse space.

It was shown early in the history of FETI-DP and BDDC, that vertex nodes alone as coarse nodes are often not sufficient to obtain robust algorithms [10, 17]. Thus, coarse degrees of freedom for BDDC or FETI-DP are often associated with average values over certain equivalence classes, i.e., edges and/or faces. The basic idea of the coarse component of the preconditioner M_{VB}^{-1} is to approximate the averages over edges or faces using adjacent vertex values.

We denote the vertex-based coarse space by \tilde{W}_{Ψ} and the original coarse space by \tilde{W}_{Π} . Then, analogously to [9], we define $\Psi : \tilde{W}_{\Psi} \rightarrow \tilde{W}_{\Pi}$ as the coarse interpolant between the coarse space based on vertices and the original coarse space based on certain equivalence classes. It is important that the coarse basis functions of \tilde{W}_{Ψ} , i.e., the columns of Ψ , build a partition of unity in the original coarse space \tilde{W}_{Π} . This is, e.g., fulfilled for the following definition of Ψ suggested in [9]. Let us first assume

\widetilde{W}_Π consists of edge constraints only. Then, each row of Ψ corresponds to a single edge constraint and has, in the case of an inner edge, two entries of 0.5 in the two columns corresponding to the two vertices located at the endpoints of the edge. All other entries of the row are zero. In case of an edge touching the Dirichlet boundary with one endpoint, the corresponding row has a single entry of 1 in the column corresponding to the vertex located at the other end of the edge. Analogously, a partition of unity can be formed for coarse spaces \widetilde{W}_Π consisting of face constraints.

Again analogously to [9], we define $\widetilde{S}_{\Pi\Pi,r} := \Psi^T \widetilde{S}_{\Pi\Pi} \Psi$ as the reduced coarse matrix. Note that the number of rows and columns of $\widetilde{S}_{\Pi\Pi,r}$ equals the number of vertices for scalar problems. The preconditioner M_{VB}^{-1} for the coarse matrix $\widetilde{S}_{\Pi\Pi}$ is then given as

$$(4.10) \quad M_{\text{VB}}^{-1} = \Psi \widetilde{S}_{\Pi\Pi,r}^{-1} \Psi^T + \text{GS}(\widetilde{S}_{\Pi\Pi}),$$

where GS denotes the application of a Gauss-Seidel preconditioner. In particular, M_{VB}^{-1} is simply a Gauss-Seidel preconditioner with an additive coarse correction. In [9], solely edge averages or solely face averages are used which are each reduced to vertex-based coarse spaces as described above. In general, also the combination of vertices, edge, and face averages as coarse components can be considered and can be reduced to a solely vertex-based coarse space.

Now we can define the vertex-based approximate BDDC preconditioner by choosing $\widehat{S}_{\Pi\Pi}^{-1} := M_{\text{VB}}^{-1}$ as approximation for $\widetilde{S}_{\Pi\Pi}^{-1}$ in (3.2). Then, we obtain the approximation $\widehat{K}_{\text{VB}}^{-1}$ of \widetilde{K}^{-1} by

$$(4.11) \quad \widehat{K}_{\text{VB}}^{-1} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \\ I \end{pmatrix} M_{\text{VB}}^{-1} \begin{pmatrix} -\widetilde{K}_{\Pi B} K_{BB}^{-1} & I \end{pmatrix},$$

and finally

$$(4.12) \quad \widehat{M}_{\text{BDDC},\text{VB}}^{-1} = \left(\widetilde{R}_D^T - \mathcal{H} P_D \right) \widehat{K}_{\text{VB}}^{-1} \left(\widetilde{R}_D - P_D^T \mathcal{H}^T \right);$$

using the notation from (3.3); see also [9].

5. Condition number bounds. First, we need to make two assumptions, which are equivalent to Assumptions 1 and 2 in [19].

ASSUMPTION 1. *For the averaging operator $E_{D,2} := \widetilde{R}(\widetilde{R}_D^T - \mathcal{H} P_D)$ we have*

$$|E_{D,2}|_{\widetilde{K}}^2 \leq \Phi(H, h) |w|_{\widetilde{K}}^2, \quad \forall w \in \widetilde{W},$$

with $\Phi(H, h)$ being a function of the maximal mesh size h and the maximal subdomain diameter H .

Under Assumption 1, the condition number of the exactly preconditioned system is bounded by

$$(5.1) \quad \kappa(M_{\text{BDDC}}^{-1} K_g) \leq \Phi(H, h);$$

see, e.g., Theorem 3 in [19]. For our homogeneous linear elasticity test case (see section 6), if appropriate primal constraints, e.g., edge averages and vertex constraints, are chosen, we obtain the condition number bound with $\Phi(H, h) = C(1 + \log(H/h))^2$.

ASSUMPTION 2. *There are positive constants \tilde{c} and \tilde{C} , which might depend on h and H , such that*

$$\tilde{c}u^T \tilde{K}u \leq u^T \hat{K}u \leq \tilde{C}u^T \tilde{K}u, \quad \forall u \in \tilde{W}.$$

Now, we can prove the following [Theorem 5.1](#) for the preconditioned operator $\hat{M}^{-1}K_g$. In the proof, we basically follow the arguments in the proof of Theorem 4 in [\[19\]](#), but here we use exact discrete harmonic extension operators, i.e., an exact $E_{D,2}$. This is in contrast to Theorem 4 in [\[19\]](#), where inexact discrete harmonic extensions are used, which is not necessary in our case. Although large parts of the proof are identical, we include the complete line of arguments here for the convenience of the reader.

THEOREM 5.1. *Let Assumptions 1 and 2 hold. Then, the preconditioned operator $\hat{M}^{-1}K_g$ is symmetric, positive definite with respect to the bilinear form $\langle \cdot, \cdot \rangle_{K_g}$ and we have*

$$\frac{1}{\tilde{C}} \langle u, u \rangle_{K_g} \leq \langle \hat{M}^{-1}K_g u, u \rangle_{K_g} \leq \frac{\Phi(H, h)}{\tilde{c}} \langle u, u \rangle_{K_g}, \quad \forall u \in V^h.$$

Therefore, we obtain the condition number bound $\kappa(\hat{M}^{-1}K_g) \leq \frac{\tilde{C}}{\tilde{c}} \Phi(H, h)$.

Proof. Let $u \in V^h$ be given. We define

$$(5.2) \quad w = \hat{K}^{-1}(\tilde{R}_D - P_D^T \mathcal{H}^T)K_g u \in \tilde{W}$$

and thus also have

$$\hat{K}w = (\tilde{R}_D - P_D^T \mathcal{H}^T)K_g u.$$

Using $\tilde{R}^T \tilde{R}_D = I$, yields $\tilde{R}^T P_D^T = \tilde{R}^T(I - \tilde{R}_D \tilde{R}^T) = 0$ and thus $\text{range}(P_D^T) \subset \text{null}(\tilde{R}^T)$. Hence, we obtain

$$(5.3) \quad \langle u, u \rangle_{K_g} = u^T \tilde{R}^T (\tilde{R}_D - P_D^T \mathcal{H}^T)K_g u = u^T \tilde{R}^T \hat{K}w = \langle w, \tilde{R}u \rangle_{\hat{K}}.$$

Using the Cauchy-Schwarz inequality and Assumption 2, we can further estimate

$$(5.4) \quad \begin{aligned} \langle w, \tilde{R}u \rangle_{\hat{K}} &\leq \langle w, w \rangle_{\hat{K}}^{1/2} \langle \tilde{R}u, \tilde{R}u \rangle_{\hat{K}}^{1/2} \stackrel{\text{Asm. 2}}{\leq} \sqrt{\tilde{C}} \langle w, w \rangle_{\hat{K}}^{1/2} \langle \tilde{R}u, \tilde{R}u \rangle_{\hat{K}}^{1/2} \\ &\stackrel{(2.4)}{=} \sqrt{\tilde{C}} \langle w, w \rangle_{\hat{K}}^{1/2} \langle u, u \rangle_{K_g}^{1/2}. \end{aligned}$$

Combining equations (5.3) and (5.4), we have $\langle u, u \rangle_{K_g} \leq \tilde{C} \langle w, w \rangle_{\hat{K}}$. Using (5.2) and (3.3), we can prove the lower bound.

$$(5.5) \quad \begin{aligned} \frac{1}{\tilde{C}} \langle u, u \rangle_{K_g} &\leq \langle w, w \rangle_{\hat{K}} \\ &\stackrel{(5.2)}{=} u^T K_g (\tilde{R}_D^T - \mathcal{H}P_D) \hat{K}^{-1} \hat{K} \hat{K}^{-1} (\tilde{R}_D - P_D^T \mathcal{H}^T) K_g u \\ &= \langle u, (\tilde{R}_D^T - \mathcal{H}P_D) \hat{K}^{-1} (\tilde{R}_D - P_D^T \mathcal{H}^T) K_g u \rangle_{K_g} \\ &\stackrel{(3.3)}{=} \langle u, \hat{M}^{-1} K_g u \rangle_{K_g} \end{aligned}$$

Let us now prove the upper bound using Assumption 1, (5.2), and (3.3).

$$(5.6) \quad \begin{aligned} \langle \hat{M}^{-1} K_g u, \hat{M}^{-1} K_g u \rangle_{K_g} &= \langle (\tilde{R}_D^T - \mathcal{H}P_D)w, (\tilde{R}_D^T - \mathcal{H}P_D)w \rangle_{K_g} \\ &= \langle \tilde{R}(\tilde{R}_D^T - \mathcal{H}P_D)w, \tilde{R}(\tilde{R}_D^T - \mathcal{H}P_D)w \rangle_{\tilde{K}} \\ &= \langle E_{D,2}w, E_{D,2}w \rangle_{\tilde{K}} = |E_{D,2}w|_{\tilde{K}}^2 \\ &\stackrel{\text{Asm. 1}}{\leq} \Phi(H, h) |w|_{\tilde{K}}^2 \end{aligned}$$

Together with Assumption 2, we obtain

$$(5.7) \quad \begin{aligned} \langle \widehat{M}^{-1} K_g u, \widehat{M}^{-1} K_g u \rangle_{K_g} &\stackrel{(5.6)}{\leq} \Phi(H, h) |w|_{\widehat{K}}^2 \stackrel{A_{sm.2}}{\leq} \frac{1}{\tilde{c}} \Phi(H, h) |w|_{\widehat{K}}^2 \\ &\stackrel{(5.5)}{=} \frac{1}{\tilde{c}} \Phi(H, h) \langle u, \widehat{M}^{-1} K_g u \rangle_{K_g}. \end{aligned}$$

Using a Cauchy-Schwarz inequality in combination with (5.7), we finally obtain

$$\langle u, \widehat{M}^{-1} K_g u \rangle_{K_g} \leq \frac{\Phi(H, h)}{\tilde{c}} \langle u, u \rangle_{K_g}. \quad \square$$

For the preconditioners considered here, we replace the inverse operator of the Schur complement in the primal variables $\tilde{S}_{\Pi\Pi}^{-1}$ by an approximation $\widehat{S}_{\Pi\Pi}^{-1}$. Therefore, we have to show that Assumption 2 used in the proof of Theorem 5.1 is still relevant and holds under certain assumptions.

ASSUMPTION 3. *There are positive constants \hat{c} and \widehat{C} , which might depend on h and H , such that*

$$\hat{c} \tilde{u}_{\Pi}^T \tilde{S}_{\Pi\Pi} \tilde{u}_{\Pi} \leq \tilde{u}_{\Pi}^T \widehat{S}_{\Pi\Pi} \tilde{u}_{\Pi} \leq \widehat{C} \tilde{u}_{\Pi}^T \tilde{S}_{\Pi\Pi} \tilde{u}_{\Pi}, \quad \forall \tilde{u}_{\Pi} \in \widetilde{W}_{\Pi}.$$

We can now prove the following lemma.

LEMMA 5.2. *Let Assumption 3 hold and \widehat{K}^{-1} be defined as in equation (3.2). Then, Assumption 2 holds with $\tilde{c} := \min(\hat{c}, 1)$ and $\tilde{C} := \max(\widehat{C}, 1)$.*

Proof. We first split $\widehat{K}^{-1} = A_1 + A_2$ into its two additive parts

$$A_1 := \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$A_2 := \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \widehat{S}_{\Pi\Pi}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B} K_{BB}^{-1} & I \end{pmatrix}.$$

The multiplication $A_1 \tilde{K}$ yields

$$(5.8) \quad A_1 \tilde{K} = \begin{pmatrix} K_{BB}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} K_{BB} & \tilde{K}_{\Pi B}^T \\ \tilde{K}_{\Pi B} & \tilde{K}_{\Pi\Pi} \end{pmatrix} = \begin{pmatrix} I & K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ 0 & 0 \end{pmatrix}.$$

By a direct computation we obtain

$$(5.9) \quad \begin{aligned} A_2 \tilde{K} &= \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \widehat{S}_{\Pi\Pi}^{-1} \begin{pmatrix} -\tilde{K}_{\Pi B} K_{BB}^{-1} & I \end{pmatrix} \begin{pmatrix} K_{BB} & \tilde{K}_{\Pi B}^T \\ \tilde{K}_{\Pi B} & \tilde{K}_{\Pi\Pi} \end{pmatrix} \\ &= \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \widehat{S}_{\Pi\Pi}^{-1} \begin{pmatrix} 0 & \tilde{S}_{\Pi\Pi} \end{pmatrix} \\ &= \begin{pmatrix} -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \\ I \end{pmatrix} \begin{pmatrix} 0 & \widehat{S}_{\Pi\Pi}^{-1} \tilde{S}_{\Pi\Pi} \end{pmatrix} \\ &= \begin{pmatrix} 0 & -K_{BB}^{-1} \tilde{K}_{\Pi B}^T \widehat{S}_{\Pi\Pi}^{-1} \tilde{S}_{\Pi\Pi} \\ 0 & \widehat{S}_{\Pi\Pi}^{-1} \tilde{S}_{\Pi\Pi} \end{pmatrix} \end{aligned}$$

Adding (5.8) and (5.9), yields the final result

$$\hat{K}^{-1}\tilde{K} = \begin{pmatrix} I & G \\ 0 & \hat{S}_{\Pi\Pi}^{-1}\tilde{S}_{\Pi\Pi} \end{pmatrix}$$

with $G = K_{BB}^{-1}\tilde{K}_{\Pi B}^T(I - \hat{S}_{\Pi\Pi}^{-1}\tilde{S}_{\Pi\Pi})$. Therefore, besides of additional eigenvalues equal to 1, $\hat{K}^{-1}\tilde{K}$ and $\hat{S}_{\Pi\Pi}^{-1}\tilde{S}_{\Pi\Pi}$ have the same spectrum, and we have $\lambda_{\min}(\hat{K}^{-1}\tilde{K}) = \min(\lambda_{\min}(\hat{S}_{\Pi\Pi}^{-1}\tilde{S}_{\Pi\Pi}), 1)$ and $\lambda_{\max}(\hat{K}^{-1}\tilde{K}) = \max(\lambda_{\max}(\hat{S}_{\Pi\Pi}^{-1}\tilde{S}_{\Pi\Pi}), 1)$. Consequently, Assumption 2 holds with $\tilde{c} := \min(\hat{c}, 1)$ and $\tilde{C} := \max(\hat{C}, 1)$. \square

For the preconditioner $\hat{M}_{\text{BDDC,AMG}}^{-1}$, we now get \hat{C} and \hat{c} depending on the properties of the AMG V-cycle used and therefore

$$(5.10) \quad \kappa(\hat{M}_{\text{BDDC,AMG}}^{-1}K_g) \leq \frac{\tilde{C}}{\tilde{c}}\Phi(H, h) = \frac{\max(\hat{C}, 1)}{\min(\hat{c}, 1)}\Phi(H, h).$$

For the three-level BDDC preconditioner $\hat{M}_{\text{BDDC,3L}}^{-1}$ we obtain, with Lemma 4.6 in [24] in two spatial dimensions and Lemma 4.7 in [23] in three spatial dimensions, $\hat{c} = \frac{1}{C_{3L}(1+\log(\frac{\hat{H}}{H}))^2}$ and $\hat{C} = 1$. Here, \hat{H} is the maximal diameter of a subregion and of course, depending on the problem and dimension, sufficient primal constraints on the second level have to be chosen; see [24, 23]. Let us note that the results in [24, 23] are only proven for scalar diffusion problems. To the best of our knowledge an extension to linear elasticity has not been published so far and is still an open problem. Using Lemma 5.2 and Theorem 5.1, we obtain the condition number bound

$$(5.11) \quad \kappa(\hat{M}_{\text{BDDC,3L}}^{-1}K_g) \leq \frac{\tilde{C}}{\tilde{c}}\Phi(H, h) = C_{3L} \left(1 + \log\left(\frac{\hat{H}}{H}\right)\right)^2 \Phi(H, h);$$

see also [24, 23].

For the vertex-based BDDC preconditioner $\hat{M}_{\text{BDDC,VB}}^{-1}$ we obtain, with Theorem 3 in [9] for edge-based or face-based coarse spaces and quasi-monotone face-connected paths, $\hat{c} \geq \frac{1}{C_1}$, $\max(\hat{C}, 1) \leq C_C$ and $\Phi(H, h) = C(1 + \log(\frac{H}{h}))^2$; see [9, Theorem 3]. Here, C_C is obtained by a coloring argument and therefore usually $C_C \geq 1$. The constant C_1 depends on geometric constants, e.g., the maximum number of subdomains connected by an edge (see [9, Lemma 2]), the maximum number of neighbors of a subdomain (see [9, (4.3)]), or typical subdomain sizes (see [9, Assumption 3]). Additionally, C_1 depends on a tolerance for the lowest coefficient along an acceptable path; see [9, Assumption 1 and 2]; cf. also [12]. The results in [9] are proven for scalar diffusion and linear elasticity problems. All together, with another constant C_{VB} , we obtain

$$\frac{\max(\hat{C}, 1)}{\min(\hat{c}, 1)} \leq C_{VB};$$

see also [9, Theorem 1 and 3] where $\hat{c} = \beta_1$ and $\hat{C} = \beta_2$ for the constants β_1 and β_2 used in [9]. Typically, we have $C_1 \geq 1$, and we then can define $C_{VB} = C_1 \cdot C_C$. Using Theorem 1, we thus obtain the condition number bound

$$(5.12) \quad \kappa(\hat{M}_{\text{BDDC,VB}}^{-1}K_g) \leq \frac{\tilde{C}}{\tilde{c}}\Phi(H, h) \leq C_{VB}\Phi(H, h);$$

see also [9, Theorem 3].

5.1. The GM (Global Matrix) Interpolation. Good constants \tilde{c}, \tilde{C} in Assumption 2 or, respectively, \hat{c}, \hat{C} in Assumption 3, are important for a small condition number and therefore a fast convergence of the approximate BDDC method. It is well known that for scalability of multigrid methods the preconditioner should preserve nullspace or near-nullspace vectors of the operator. Therefore, the AMG method should preserve the nullspace of the operator on all levels and these nullspace vectors have to be in the range of the AMG interpolation. While classical AMG guarantees this property only for constant vectors, the global matrix approach (GM), introduced in [3], allows the user to specify certain near-nullspace vectors, which are interpolated exactly from the coarsest to the finest level; details on the method and its scalability can be found in [3, 2]. Since we are interested in linear elasticity problems, we choose the rotations of the body for the exact interpolation. All translations of the body are already interpolated exactly in classical AMG approaches for systems of PDEs since they use classical interpolation applied component-by-component. In $\widehat{M}_{\text{BDDC, AMG}}^{-1}$ AMG is applied to \tilde{S}_{III} and thus we need the three rotations in the space \tilde{W}_{II} , which is the restriction of \tilde{W} to the primal constraints. Therefore, we first assemble the rotations of the subdomains Ω_i locally, extract the primal components, and finally insert them into three global vectors in \tilde{W}_{II} . In our implementation, we always use BoomerAMG from the hypre package [11], where a highly scalable implementation of the GM2 approach is integrated; see [2]. Let us remark that GM2 is one of two variants to choose the interpolation implemented in BoomerAMG and is recommended to use instead of GM1. In [2] it also showed a better scalability than GM1. We will compare the use of the GM2 approach with a hybrid AMG approach for systems of PDEs. By hybrid AMG approaches, we refer to methods, where the coarsening is based on the physical nodes (nodal coarsening) but the interpolation is based on the degrees of freedoms. In general, a nodal coarsening approach is beneficial for the solution of systems of PDEs, and all degrees of freedom belonging to the same physical node are either all coarse or fine on a certain level. The latter fact is also mandatory for the GM2 approach. Therefore, GM2 is based on the same nodal coarsening and can also be considered as a hybrid approach.

6. Implementation and Model Problems. Our parallel implementation uses C/C++ and PETSc version 3.9.2 [6]. All matrices are completely local to the computational cores. All assemblies and prolongations are performed using PETSc *VecScatter* and *VecGather* operations. A more detailed description of the parallel data structures of our implementation of the linear BDDC preconditioner can be found in [14], where different nonlinear BDDC methods are applied to hyperelasticity and elasto-plasticity problems.

Since the preconditioners for the coarse problem are in the focus of this paper, we include some details on the implementation of the different variants. In general, the coarse problem \tilde{S}_{III} is assembled on a subset of the available cores. The number of cores can be chosen arbitrarily and should depend on the size of the coarse problem to obtain a good performance. While BoomerAMG and BDDC itself can be applied to \tilde{S}_{III} in parallel, for exact BDDC (M_{BDDC}^{-1}) a sequential copy of \tilde{S}_{III} is sent to each computational core and a sparse direct solver is applied. This is, of course, not scalable in parallel and one would prefer to avoid it. Using a sequential Gauss-Seidel implementation in $\widehat{M}_{\text{BDDC, VB}}^{-1}$ also requires a sequential copy of \tilde{S}_{III} and additionally sequential copies of $\tilde{S}_{\text{III}, r} := \Psi^T \tilde{S}_{\text{III}} \Psi$, and eventually, depending on the implementation, also of Ψ . This can be avoided using a parallel implementation of

the Gauss-Seidel preconditioner. Therefore, as an alternative to the sequential approach, we also use the parallel PETSc implementation of SOR/Gauss-Seidel, which is in fact a block Jacobi preconditioner in between the local blocks associated with the different MPI ranks and an SOR/Gauss-Seidel preconditioner on the local blocks themselves. This can obviously deteriorate the convergence of the method, but we only have to build a local copy of $\tilde{S}_{\text{III},r}$, which is much smaller compared to \tilde{S}_{III} . All other matrices can be stored in a distributed fashion. Let us finally remark that we can apply the Gauss-Seidel preconditioner additively as described in (4.10) as well as multiplicatively, which is of course more robust.

7. Numerical Results. In this paper, we restrict ourselves to homogeneous linear elasticity problems. For heterogeneous examples or different model problems we refer to [14] for $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ or [9] for $\widehat{M}_{\text{BDDC,VB}}^{-1}$. All computations are performed on magnetUDE supercomputer (University of Duisburg-Essen) or JUWELS (FZ Juelich).

7.1. Three-Level BDDC and BDDC with AMG coarse preconditioner.

We first concentrate on a comparison between $\widehat{M}_{\text{BDDC,3L}}^{-1}$ and $\widehat{M}_{\text{BDDC,AMG}}^{-1}$, which clearly have the largest parallel potential, especially due to the larger coarsening ratio from the second to the coarsest level. Also $\widehat{M}_{\text{BDDC,3L}}^{-1}$ can be easily extended to a multilevel preconditioner while $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ already consists of several levels. The alternative $\widehat{M}_{\text{BDDC,VB}}^{-1}$ is limited in scalability by construction, since the vertex-based coarse space is always solved by a sparse direct solver in our implementation. We therefore analyze and compare $\widehat{M}_{\text{BDDC,VB}}^{-1}$ separately in subsection 7.2.

To have a theoretical baseline, we always include the exact BDDC preconditioner M_{BDDC}^{-1} in all figures. To verify the quadratic dependence of the condition number on the logarithm of H/h , which can be seen as a measure of the subdomain size, we provide Figure 2. There, we consider a linear elastic cube decomposed into 512 subdomains with Young modulus $E = 210 \text{ GPa}$ and different Poisson ratios. As a coarse space we enforce continuity in all vertices and in all edge averages. With a Poisson ratio of 0.3 (Figure 2 (top)), all methods show a similar behavior and the condition numbers are comparable to the exact BDDC preconditioner. For $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ it is useful to include the GM approach, while for $\widehat{M}_{\text{BDDC,3L}}^{-1}$ both tested setups, i.e., 8 or 64 subdomains per subregion, show a similar behavior. Choosing a larger Poisson ratio of 0.49 (Figure 2 (bottom)), $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ has higher condition numbers, especially for small subdomain sizes. But for larger subdomain sizes and using GM, $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ again shows a similar behavior. Let us remark that we always use a highly scalable AMG setup, i.e., aggressive HMIS coarsening, *ext + i* long range interpolation, nodal coarsening, a threshold of 0.3, and a maximum of three entries per row in the AMG interpolation matrices. Less aggressive strategies might show lower condition numbers, but we explicitly optimized the parameters to obtain good parallel scalability; see [2].

For the same setup with a Poisson ratio of 0.3 but fixed $H/h = 24$, we perform a weak scaling study in Figure 3 up to 4096 cores. Considering the number of cg iterations until convergence (Figure 3 (top)), the GM approach is necessary in $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ to obtain results of similar quality as $\widehat{M}_{\text{BDDC,3L}}^{-1}$. The same can be observed considering the time to solution; see Figure 3 (bottom). The time to solution is always the complete runtime measured from the program start until it finishes. This especially includes the assembly of the linear system, the setup of the preconditioner, and the iteration/solution. Of course, the exact BDDC preconditioner does not scale due to

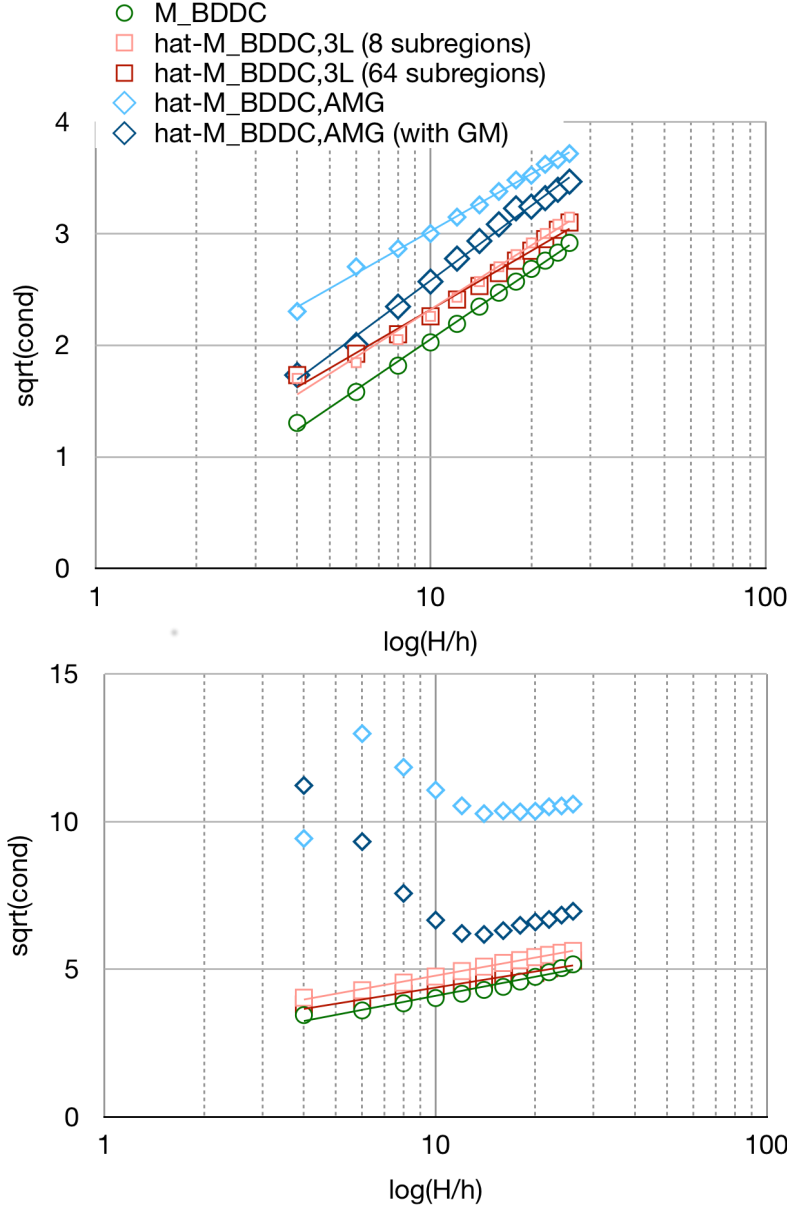


FIG. 2. Homogeneous linear elastic cube decomposed into 512 subdomains with $H/h = 4, 6, \dots, 26$. **Top:** $E = 210.0$ and $\nu = 0.3$; **Bottom:** $E = 210.0$ and $\nu = 0.49$. We vary the number of subdomains per subregion in $\hat{M}_{\text{BDDC},3\text{L}}^{-1}$ and we compare nodal AMG and AMG-GM in $\hat{M}_{\text{BDDC},\text{AMG}}^{-1}$. Computed on the magnitUDE supercomputer.

the sequential coarse solve.

7.2. Vertex-Based BDDC. We provide a weak scaling test up to 5832 cores for $\hat{M}_{\text{BDDC},\text{VB}}^{-1}$ for a similar model problem, i.e., linear elasticity with a Poisson ratio of 0.3 and a Young modulus of 210 GPa. In Figure 4 we provide a comparison with exact BDDC and $\hat{M}_{\text{BDDC},\text{AMG}}^{-1}$ using GM with respect to cg iterations as well as

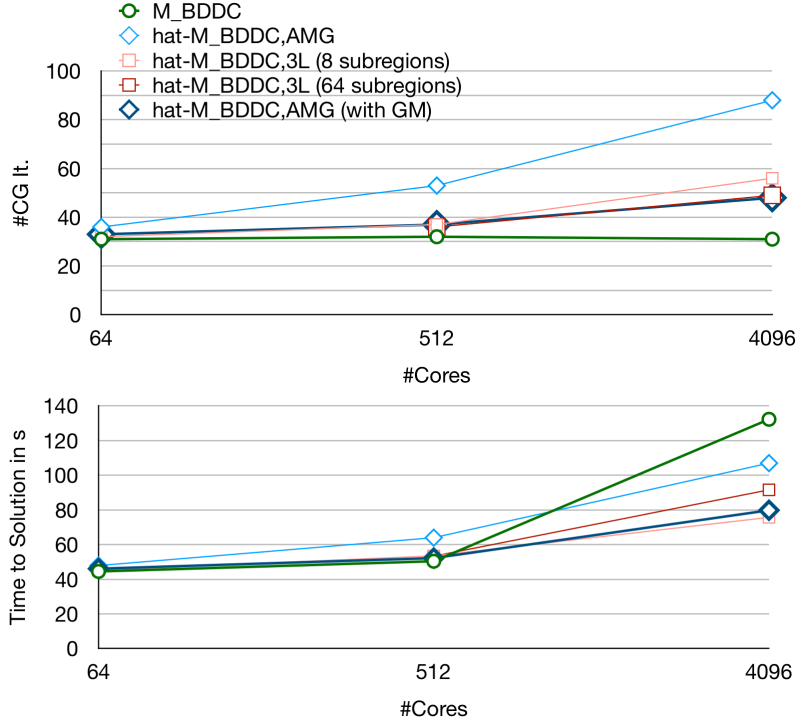


FIG. 3. Comparison of $\widehat{M}_{\text{BDDC}}^{-1}$, $\widehat{M}_{\text{BDDC},3\text{L}}^{-1}$ with 8 or 64 subregions, and $\widehat{M}_{\text{BDDC},\text{AMG}}^{-1}$ with and without GM. Using vertex and edge constraints. Homogeneous linear elastic cube decomposed into 64, 512, and 4096 subdomains with $H/h = 24$. **Top:** Number of CG iterations; **Bottom:** Total time to solution including assembly of stiffness matrices, setup of the preconditioner and solution phase. Computed on JUWELS.

time to solution. Considering $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$, a multiplicative combination of Gauss-Seidel applied to \tilde{S}_{III} and the direct solve of the vertex based coarse problem is always the better choice compared to an additive variant. The parallel Gauss-Seidel method, which - as implemented in PETSc - is in fact a block Jacobi preconditioner in between the processors parts of the matrix, always results in more cg iterations but faster runtimes. With respect to parallel scalability, the best variant of $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$ is competitive with $\widehat{M}_{\text{BDDC},\text{AMG}}^{-1}$, at least up to the moderate core count of 5832. For an increasing number of cores, we expect $\widehat{M}_{\text{BDDC},\text{AMG}}^{-1}$ to outperform $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$ due to its inherent multilevel structure.

8. Conclusion. We have presented different approaches to approximate the coarse solve in BDDC and compared them with respect to theory and parallel scalability for the first time. If an appropriate AMG approach is available, e.g., the GM approach in the case of linear elasticity problems, $\widehat{M}_{\text{BDDC},\text{AMG}}^{-1}$ and $\widehat{M}_{\text{BDDC},3\text{L}}^{-1}$ show a very similar behavior and both variants can be recommended. Up to a moderate number of compute cores also $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$ can be an adequate alternative. An advantage of $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$ is the fact that neither a further decomposition into subregions is necessary nor an appropriate AMG method has to be chosen. On the other hand, the parallel potential of $\widehat{M}_{\text{BDDC},\text{VB}}^{-1}$ is limited, since it is not easily extendable to an arbitrary number of levels.

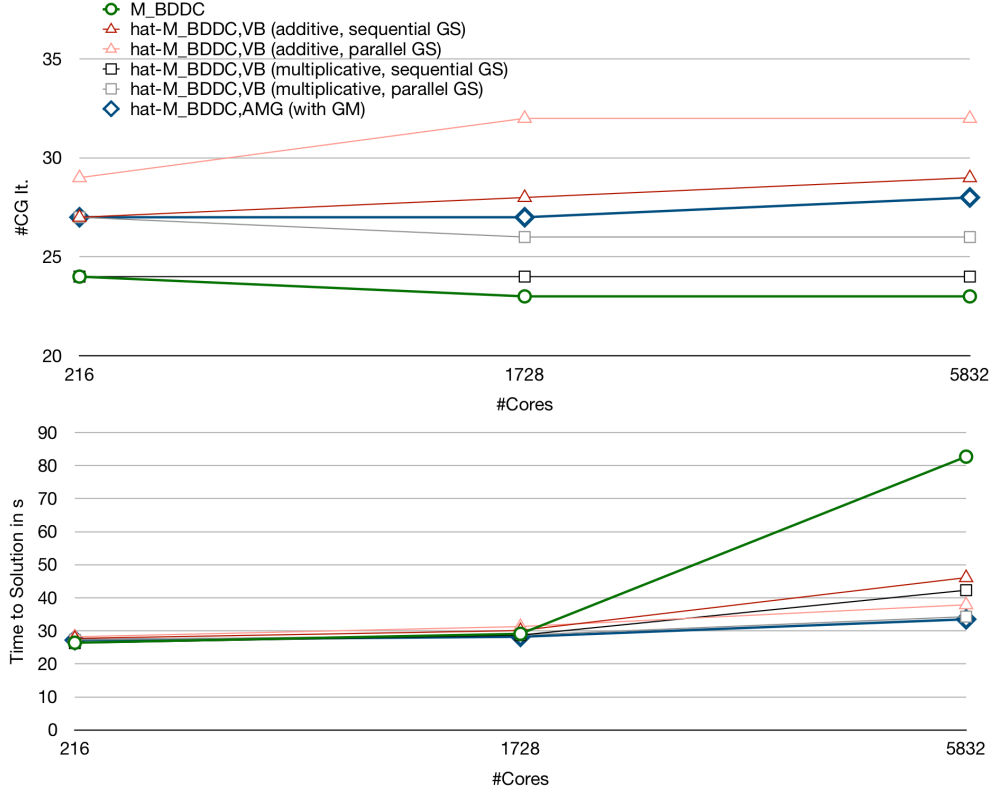


FIG. 4. Comparison of M_{BDDC}^{-1} , $\widehat{M}_{\text{BDDC,VB}}^{-1}$ using additive/multiplicative sequential/parallel Gauss-Seidel, and $\widehat{M}_{\text{BDDC,AMG}}^{-1}$ with GM. Using only edge constraints. Homogeneous linear elastic cube with $H/h = 22$. **Top:** Number of CG iterations; **Bottom:** Total time to solution including assembly of stiffness matrices, setup of the preconditioner and solution phase. Computed on the magnitUDE supercomputer.

Acknowledgments. The authors gratefully acknowledge the computing time granted by the Center for Computational Sciences and Simulation (CCSS) of the Universität of Duisburg-Essen and provided on the supercomputer **magnitUDE** (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at the Zentrum für Informations- und Mediendienste (ZIM).

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer **JUWELS** at Jülich Supercomputing Centre (JSC).

REFERENCES

- [1] Santiago Badia, Alberto F. Martín, and Javier Principe. Multilevel Balancing Domain Decomposition at Extreme Scales. *SIAM J. Sci. Comput.*, 38(1):C22–C52, 2016.
- [2] Allison H. Baker, Axel Klawonn, Tzanio Kolev, Martin Lanser, Oliver Rheinbach, and Ulrike Meier Yang. Scalability of Classical Algebraic Multigrid for Elasticity to half a Million Parallel Tasks. In *Software for Exascale Computing—SPPEXA 2013–2015*, volume 113 of *Lect. Notes Comput. Sci. Eng.*, pages 113–140. Springer, [Cham], 2016.
- [3] Allison H. Baker, Tzanio. V. Kolev, and Ulrike M. Yang. Improving algebraic multigrid in-

- terpolation operators for linear elasticity problems. Numer. Linear Algebra Appl., 17(2-3):495–517, 2010.
- [4] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.10, Argonne National Laboratory, 2018.
 - [5] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2018.
 - [6] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997.
 - [7] L. Beirão da Veiga, L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. Parallel sum primal spaces for isogeometric deluxe BDDC preconditioners. In Domain decomposition methods in science and engineering XXIII, volume 116 of Lect. Notes Comput. Sci. Eng., pages 17–29. Springer, Cham, 2017.
 - [8] Clark R. Dohrmann. An approximate BDDC preconditioner. Numer. Linear Algebra Appl., 14(2):149–168, 2007.
 - [9] Clark R. Dohrmann, Kendall H. Pierson, and Olof B. Widlund. Vertex-Based Preconditioners for the Coarse Problem of BDDC. Technical Report TR2019-993, Courant Institute, New York, 2019.
 - [10] Charbel Farhat, Michael Lesoinne, and Kendall Pierson. A scalable dual-primal domain decomposition method. Numer. Linear Algebra Appl., 7(7-8):687–714, 2000. Preconditioning techniques for large sparse matrix problems in industrial applications (Minneapolis, MN, 1999).
 - [11] Van E. Henson and Ulrike Meier Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Appl. Numer. Math., 41:155–177, 2002.
 - [12] A. Klawonn and O.B. Widlund. Dual-primal feti methods for linear elasticity. Comm. Pure Appl. Math., 59:1523–1572, 2006.
 - [13] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Toward Extremely Scalable Nonlinear Domain Decomposition Methods for Elliptic Partial Differential Equations. SIAM J. Sci. Comput., 37(6):C667–C696, 2015.
 - [14] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Nonlinear BDDC Methods with Approximate Solvers. ETNA, 49:pp. 244–273, 2018.
 - [15] Axel Klawonn, Martin Lanser, Oliver Rheinbach, and Matthias Uran. Nonlinear FETI-DP and BDDC methods: A Unified Framework and Parallel Results. SIAM J. Sci. Comput., 39(6):C417–C451, 2017.
 - [16] Axel Klawonn and Oliver Rheinbach. Inexact FETI-DP methods. Internat. J. Numer. Methods Engrg., 69(2):284–307, 2007.
 - [17] Axel Klawonn and Oliver Rheinbach. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. Comput. Methods Appl. Mech. Engrg., 196(8):1400–1414, 2007.
 - [18] Axel Klawonn and Oliver Rheinbach. Highly scalable parallel domain decomposition methods with an application to biomechanics. ZAMM Z. Angew. Math. Mech., 90(1):5–32, 2010.
 - [19] Jing Li and Olof B. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. Comput. Methods Appl. Mech. Engrg., 196(8):1415–1428, 2007.
 - [20] Jan Mandel, Bedřich Sousedík, and Clark R. Dohrmann. Multispace and multilevel BDDC. Computing, 83(2-3):55–85, 2008.
 - [21] Bedřich Sousedík, Jakub Šístek, and Jan Mandel. Adaptive-multilevel BDDC and its parallel implementation. Computing, 95(12):1087–1119, 2013.
 - [22] Andrea Toselli and Olof Widlund. Domain Decomposition Methods—Algorithms and Theory, volume 34 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2005.
 - [23] Xuemin Tu. Three-level BDDC in three dimensions. SIAM J. Sci. Comput., 29(4):1759–1780, 2007.
 - [24] Xuemin Tu. Three-level BDDC in two dimensions. Internat. J. Numer. Methods Engrg., 69(1):33–59, 2007.