# PHYSICS-BASED BALANCING DOMAIN DECOMPOSITION BY CONSTRAINTS FOR MULTI-MATERIAL PROBLEMS

SANTIAGO BADIA, ALBERTO F. MARTÍN, AND HIEU NGUYEN

ABSTRACT. In this work, we present a novel balancing domain decomposition by constraints preconditioner that is robust for multi-material problems. We start with a well-balanced subdomain partition, and based on an aggregation of elements according to their physical coefficients, we end up with a finer *physics-based* (PB) subdomain partition. Next, we define geometrical objects (corners, edges, and faces) for this PB partition, and select some of them to enforce subdomain continuity (primal objects). When the physical coefficient in each PB subdomain is constant and the set of selected primal objects satisfy a mild condition on the existence of acceptable paths, we can show both theoretically and numerically that the condition number does not depend on the contrast of the coefficient. An extensive set of numerical experiments for 2D and 3D Poisson's and linear elasticity problems is provided to support our findings. In particular, we show robustness and weak scalability of the new preconditioner up to 8232 cores when applied to 3D multi-material problems with the contrast of the physical coefficient up to $10^8$ and more than half a billion degrees of freedom. For the scalability analysis, we have exploited a highly scalable advanced inter-level overlapped implementation of the preconditioner that deals very efficiently with the coarse problem computation. The proposed preconditioner is compared against a state-of-the-art implementation of an adaptive BDDC method in PETSc for thermal and mechanical multi-material problems.

**Keywords:** BDDC, multi-material problem, adaptive coarse space, parallel solver, parallel preconditioner
**AMS subject classifications:** 65N55, 65N22, 65F08

## CONTENTS

This work has partially been funded by the European Research Council under the FP7 Program Ideas through the Starting Grant No. 258443 - COMFUS: Computational Methods for Fusion Technology and the FP7 NUMEXAS project under grant agreement 611636. Financial support from the EC - International Cooperation in Aeronautics with China (Horizon 2020) under the project: Efficient Manufacturing for Aerospace Components USing Additive Manufacturing, Net Shape HIP and Investment Casting (EMUSIC) and the H2020-FoF-2015 under the project: Computer Aided Technologies for Additive Manufacturing (CAxMan) are also acknowledged. S. Badia gratefully acknowledges the support received from the Catalan Government through the ICREA Acadèmia Research Program. H. Nguyen thanks Vietnam National Foundation for Science and Technology Development (NAFOSTED) for the financial support through Grant No. 101.99-2017.13. Finally, the authors thankfully acknowledge the computer resources at Marenostrum III and IV and the technical support provided by BSC under the RES (Spanish Supercomputing Network), and the North-German Supercomputing Alliance (HLRN) for providing HPC resources that have contributed to the research results reported in this paper.

# 1. Introduction

Many realistic simulations in science and engineering involve multiple materials, e.g., composite materials. The linear systems resulting from the discretization of these problems are hard to solve. The use of direct solvers at a sufficiently fine scale can be prohibitively expensive, even with modern supercomputers, due to their high complexity and scalability issues. In addition, the high contrast of the physical properties significantly increases the condition number of the resulting linear systems, posing great challenges for iterative solvers. In this work, we will focus on developing a domain decomposition (DD) preconditioner that is robust for multi-material thermal and solid mechanics problems with a high variation of the coefficients in the corresponding partial differential equations (PDEs).

DD is one of the most popular approaches to solve large-scale problems on parallel supercomputers. It splits a problem into weakly coupled subproblems on smaller subdomains and use parallel local solutions on these subdomains to form a parallel preconditioner for the original problem [54, 44]. In DD, the coarse space plays an important role in achieving scalability as well as robustness with respect to variations in the coefficient. Many early DD methods, such as those in [14, 21, 20, 35, 58], are robust for heterogeneous problems when the subdomain partition is a geometric coarse grid that resolves the discontinuities in the properties of the media. This is a strong requirement, since such a decomposition does not lead in general to load-balanced partitions with a reduced interface.

For heterogeneous problems, there have been recent works on coarse grids that do not resolve the heterogeneity in the media [28, 48, 28, 46, 47], and especially automatic coarse spaces that adapt to the variation in the properties of the media [25, 26, 45, 51, 18, 53, 52, 31, 30, 32, 39, 27]. In the latter, the coarse spaces are constructed from eigenfunctions associated with small eigenvalues (low-frequency modes) of appropriate generalised eigenvalue problems. This approach is backed up by rigorous mathematical theory and has been numerically shown to be robust for general heterogeneous problems. However, solving eigenvalue problems is expensive and extra implementation effort is required as coarse spaces in DD methods are not naturally formulated as eigenfunctions. Another approach is to use the deluxe scaling technique where local auxiliary Dirichlet problems are solved to compute efficient averaging operators [36, 17, 56]. The approach yields robust DD methods, but it incurs extra implementation and computation cost due to the auxiliary problems and spectral solvers. Even though these methods can be required for some very complex heterogeneous problems, in this work we analyze ways to eliminate the need of spectral solvers and auxiliary problems in some situations. We formulate a *balancing DD by constraints (BDDC) preconditioner that requires no eigenvalue or auxiliary problem and is very robust with the contrast of the coefficients for thermal and mechanical multi-material simulations.* The main motivation behind this work is to achieve such goal while *maintaining the simplicity* of the BDDC preconditioner.

The BDDC method was introduced by Dohrmann in 2003 [15]. It is an improved version of the balancing DD (BDD) method by Mandel [40] and has a close connection with the FETI-DP method [23, 22]. In fact, it can be shown that the eigenvalues of the preconditioned operators associated with BDDC and FETI-DP are almost identical [42, 37, 12]. The BDDC method is particularly well suited for extreme scale simulations, since it allows for a very aggressive coarsening, the computations at different levels can be computed in parallel, the subdomain problems can be solved inexactly [16, 38] by, e.g., one AMG cycle, and it can straightforwardly be extended to multiple levels [55, 43]. All of these properties have been carefully exploited in the series of articles [3, 4, 5, 6] where an extremely scalable implementation of these algorithms has been proposed, leading to excellent weak scalability on nearly half a million cores in its multilevel version. These algorithms are available in the `FEMPAR` library [7, 1]. A freely available implementation of BDDC methods with deluxe scaling can be found in PETSc [57].

Our new BDDC method is motivated from the fact that non-overlapping DD methods, such as BDDC and FETI-DP, are robust with the variation and contrast of the coefficient if it is constant (or varies mildly) in each subdomain [35, 34, 54]. This implies that, in order to have robustness for BDDC methods, one could use a physics-based (PB) partition obtained by aggregating elements of the same coefficient value.

However, using this type of partition is impractical as the number of subdomains might be too large and can lead to a poor load balancing among subdomains and large interfaces. In order to solve this dilemma, we propose to use a well-balanced partition, e.g., one obtained from an automatic graph partitioner like METIS [29], to distribute the workload among processors, as usual. Then, we consider a sub-partition of subdomains based on the physical coefficients, leading to a PB partition. Continuity constraints among subdomains will be defined through the definition of objects based on the PB partition. Consequently, the interface objects are adaptively defined according to the variation of the coefficient. The resulting BDDC preconditioner with constraints imposed on subfaces, subedges, and vertices, is coined PB-BDDC. These ideas can readily be applied to FETI-DP preconditioners.

For multi-material problems, e.g., problems with isolated channels or inclusions, it is computationally feasible to require the physical coefficient in each PB subdomain to be constant. In this situation, we are able to prove that *the resulting BDDC method is scalable and its convergence is independent of the contrast of the coefficient.* An important feature is that not all PB objects are necessary and any selected set of PB objects satisfying a mild condition on the so-called acceptable paths is enough to guarantee robustness [34].

By definition, the PB-BDDC method considers an enlarged coarse space to make the preconditioner robust with respect to heterogeneities. In order to alleviate the effect of potentially larger coarse spaces compared to homogeneous problems, we propose two remedies:

(1) In section 4, taking advantage of the fact that not all PB objects are required for robustness, we study how to keep the size of the coarse problem close to minimal. For the sake of illustration, we present some examples with different distribution of the coefficient and the corresponding minimal set of PB objects required for robustness. The number of required PB objects is the same as the number of extra eigenfunctions needed in other approaches for adaptive coarse spaces based on eigenvalue problems [52, 31, 27]. For general cases, a simple procedure to construct a close to minimal coarse space is also provided. The procedure is based on a mathematically supported definition of *acceptable paths.*

(2) For highly complex problems with many different regions, optimal robust coarse spaces can still be large. In order to cope larger coarse spaces and keep good scalability properties, we have considered an implementation of the PB-BDDC method in the extremely scalable BDDC code in `FEMPAR` [3, 4, 5, 6]. It exploits inter-level overlapping of computations and communications so that coarse tasks run in parallel and can be masked in run time by fine tasks. The implementation also provides a recursive multilevel extension which deals efficiently with large coarse spaces. This is particularly important for the PB-BDDC method presented herein. In section 5.2, we show excellent scalability results up to 8232 cores and more than half a billion of unknowns for a 3D problem.

Summarizing, the PB-BDDC method does not require to solve eigenvalue or auxiliary problems on patches of subdomains, which could incur high computational cost and the formulation and implementation of the PB-BDDC method is very much the same as for the standard BDDC method. The only difference is in identifying and defining BDDC objects to impose constraints. In other words, *the simplicity of the standard BDDC method is maintained.* On the other hand, the PB-BDDC methods involve a richer interface with the application software, e.g., access to the physical properties of the problem, which is in line with the philosophy of the `FEMPAR` library [7, 1], i.e., a tight interaction of discretization and linear solver steps to fully exploit the mathematical structure of the PDE operator.

The rest of the paper is organised as follows. In section 2, we introduce the model problem, the domain partitions and the BDDC object classification. In section 3, we present the formulation of the PB-BDDC method as well as its key ingredients, namely coarse degrees of freedom (coarse DOFs), weighting and harmonic extension operators. The convergence analysis is also provided in this section. An automatic algorithm to determine small and effective coarse spaces is proposed in section 4. In section 5, we provide an extensive set of numerical experiments to demonstrate the robustness and efficiency of the PB-BDDC method, and compare the proposed algorithm with a state-of-the-art implementation of BDDC with deluxe scaling in PETSc. We finally draw some conclusions in section 6.

## 2. Problem setting

Let $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$ being the space dimension, be a bounded polyhedral domain. For a model problem, we study the Poisson's and linear elasticity equations with piece-wise constant physical parameters. Let us consider homogeneous Dirichlet boundary conditions for simplicity in the exposition (the non-homogeneous case only involves an obvious modification of the right-hand side). The Poisson problem in weak form reads as: find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \alpha \nabla u \cdot \nabla v \, dx = \int_\Omega f v dx, \qquad \text{for any } v \in H_0^1(\Omega), \tag{1}$$

with $f \in H^{-1}(\Omega)$. The linear elasticity problem reads as: find $u \in H_0^1(\Omega)^d$ such that

$$\int_\Omega \alpha \epsilon(u) : \epsilon(v) \, dx + \int_\Omega \alpha\beta \operatorname{div} u \operatorname{div} v \, dx = \int_\Omega f \cdot v \, dx \qquad \text{for any } v \in H_0^1(\Omega)^d, \tag{2}$$

with $f \in H^{-1}(\Omega)^d$, where $\epsilon(u) \doteq \frac{1}{2}(\nabla u + \nabla u^T)$ and $(\alpha, \beta)$ can be computed from Young's modulus $E$ and Poisson's ratio $\nu$ as $\alpha = \frac{E}{(1+\nu)}$ and $\beta = \frac{\nu}{(1-2\nu)}$. We consider that $\nu$ is bounded away from $\frac{1}{2}$, i.e., compressible elasticity·

Let $\mathcal{T}$ be a shape-regular quasi-uniform mesh of $\Omega$ with characteristic size $h$. It can consist of tetrahedra or hexahedra for $d = 3$, or triangles or quadrilaterals for $d = 2$. For simplicity of exposition, we assume that $\alpha$ is constant on each element $\tau \in \mathcal{T}$.

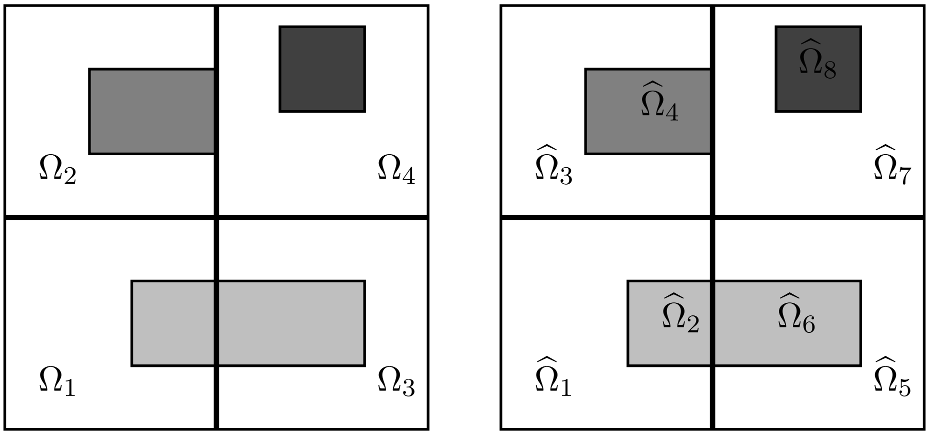


Figure 1. An example of an original partition $\Theta$ (left) and a physics-based partition $\Theta_{\text{pb}}$(right) of a square domain where different colors represent different values of $\alpha$. On the left, we have a $\Theta = \{\Omega_1, \Omega_2, \Omega_3, \Omega_4\}$. On the right, we show the corresponding PB-partition for every subdomain in $\Theta$: $\Theta_{\text{pb}}(\Omega_1) = \{\widehat{\Omega}_1, \widehat{\Omega}_2\}$, $\Theta_{\text{pb}}(\Omega_2) = \{\widehat{\Omega}_3, \widehat{\Omega}_4\}$, $\Theta_{\text{pb}}(\Omega_3) = \{\widehat{\Omega}_5, \widehat{\Omega}_6\}$, and $\Theta_{\text{pb}}(\Omega_4) = \{\widehat{\Omega}_7, \widehat{\Omega}_8\}$. The complete PB-partition is $\Theta_{\text{pb}} = \{\widehat{\Omega}_1, \ldots, \widehat{\Omega}_8\}$. Further, we have $\omega(\widehat{\Omega}_1) = \omega(\widehat{\Omega}_2) = \Omega_1$, $\omega(\widehat{\Omega}_3) = \omega(\widehat{\Omega}_4) = \Omega_2$, $\omega(\widehat{\Omega}_5) = \omega(\widehat{\Omega}_6) = \Omega_3$, $\omega(\widehat{\Omega}_7) = \omega(\widehat{\Omega}_8) = \Omega_4$.

2.1. **Domain partitions.** We first consider a partition $\Theta$ of the domain $\Omega$ into non-overlapping open subdomains. This partition must be driven by computational efficiency in distributed memory platforms, i.e., it should have a reduced interface size and lead to a well-balanced distribution of workload among processors. In a parallel implementation, each subdomain in $\Theta$ is generally assigned to a processor. We further assume that every $\mathcal{D} \in \Theta$ can be obtained by *aggregation of elements* in $\mathcal{T}$ and is connected. We denote by $\Gamma(\Theta)$ the interface of the partition $\Theta$, i.e., $\Gamma(\Theta) \doteq (\cup_{\mathcal{D}\in\Theta} \partial\mathcal{D}) \setminus \partial\Omega$, and by $\Gamma_h(\Theta)$ the discrete version of the interface.

We also consider a PB subdomain partition. This partition is used latter in the new definition of coarse objects and in the analysis. It is, however, not used for work distribution. Given a subdomain $\mathcal{D} \in \Theta$, we can further consider its partition $\Theta_{\text{pb}}(\mathcal{D})$ into a set of "sub-subdomains" with constant $\alpha$. Clearly, the resulting global PB partitions $\Theta_{\text{pb}} \doteq \{\Theta_{\text{pb}}(\mathcal{D})\}_{\mathcal{D}\in\Theta}$ is also a partition of $\Omega$ (into PB subdomains).

The interface of this partition is $\Gamma(\Theta_{\rm pb}) \doteq (\cup_{\hat{\mathcal{D}}\in\Theta_{\rm pb}} \partial\hat{\mathcal{D}})\backslash\partial\Omega$. For a subdomain $\mathcal{D} \in \Theta$ (analogously for $\hat{\mathcal{D}} \in \Theta_{\rm pb}$), we denote by $\mathcal{T}_{\mathcal{D}}$ the submesh of $\mathcal{T}$ associated with $\mathcal{D}$, $\mathcal{T}_{\mathcal{D}} \doteq \{\tau \in \mathcal{T} : \tau \subset \mathcal{D}\} \subset \mathcal{T}$. For any $\hat{\mathcal{D}} \in \Theta_{\rm pb}$, let $\omega(\hat{\mathcal{D}})$ be the only subdomain in $\Theta$ that contains $\hat{\mathcal{D}}$. In Figure 1, we show an example of the original partition $\Theta$ and the PB partition $\Theta_{\rm pb}$ for a simple problem. The meaning of $\Theta_{\rm pb}(\mathcal{D})$ and $\omega(\mathcal{D})$ is also illustrated.

2.2. **Finite element spaces.** Let us perform a discretization of (1) by a continuous finite element (FE) space $\bar{\mathbb{V}}$ associated with the mesh $\mathcal{T}$. The discontinuous Galerkin (DG) case will not be considered in this work, but we refer the reader to [19] for more information.

For every subdomain $\mathcal{D} \in \Theta$, we consider a FE space $\mathbb{V}_{\mathcal{D}}$ associated with the local mesh $\mathcal{T}_{\mathcal{D}}$. Let $H(\mathcal{D})$ be the characteristic length of the subdomain $\mathcal{D}$ and $h(\mathcal{D})$ be the characteristic length of the FE mesh $\mathcal{T}_{\mathcal{D}}$. We define the Cartesian product of local FE spaces as $\mathbb{V} = \Pi_{\mathcal{D}\in\Theta}\mathbb{V}_{\mathcal{D}}$. We note that functions in this space are allowed to be discontinuous across the interface $\Gamma(\Theta)$. Clearly, $\bar{\mathbb{V}} \subset \mathbb{V}$.

For a subdomain $\mathcal{D} \in \Theta$, we also define the subdomain FE operator $\mathcal{A}_{\mathcal{D}} : \mathbb{V}_{\mathcal{D}} \to \mathbb{V}'_{\mathcal{D}}$ as $\langle \mathcal{A}_{\mathcal{D}} u, v\rangle \doteq \int_{\mathcal{D}} \alpha \nabla u \cdot \nabla v \, dx$, for all $u, v \in \mathbb{V}_{\mathcal{D}}$, and the sub-assembled operator $\mathcal{A}^{\Theta} : \mathbb{V} \to \mathbb{V}'$ as $\langle \mathcal{A}^{\Theta} u, v\rangle \doteq \sum_{\mathcal{D}\in\Theta} \langle \mathcal{A}_{\mathcal{D}} u, v\rangle$, for all $u, v \in \mathbb{V}$.

A function $u \in \mathbb{V}_{\mathcal{D}}$ is said to be discrete $\alpha$-harmonic in $\mathcal{D}$ if

$$\langle \mathcal{A}_{\mathcal{D}} u, v\rangle = 0, \quad \text{for any } v \in \mathbb{V}_{0,\mathcal{D}},$$

where $\mathbb{V}_{0,\mathcal{D}} \doteq \{v \in \mathbb{V}_{\mathcal{D}} : v = 0 \text{ on } \partial\mathcal{D}\}$. It should be noted that if $u$ is discrete $\alpha$-harmonic in $\mathcal{D}$ then it satisfies the energy minimising property, namely

$$\langle \mathcal{A}_{\mathcal{D}} u, u\rangle \leq \langle \mathcal{A}_{\mathcal{D}} v, v\rangle, \quad \forall v \in \mathbb{V}_{\mathcal{D}}, \ v|_{\partial\mathcal{D}} = u|_{\partial\mathcal{D}}.$$

In addition, we consider the assembled operator $\mathcal{A} : \bar{\mathbb{V}} \to \bar{\mathbb{V}}'$, defined by $\langle \mathcal{A} u, v\rangle = \int_{\Omega} \alpha \nabla u \cdot \nabla v \, dx$, for all $u, v \in \bar{\mathbb{V}}$. This operator is the Galerkin projection of $\mathcal{A}^{\Theta}$ onto $\bar{\mathbb{V}}$. We want to compute a FE approximation $u \in \bar{\mathbb{V}}$ of $u^{\star}$ in (1) such that

$$\langle \mathcal{A} u, v\rangle = \langle f, v\rangle, \quad \text{for any } v \in \bar{\mathbb{V}}. \tag{3}$$

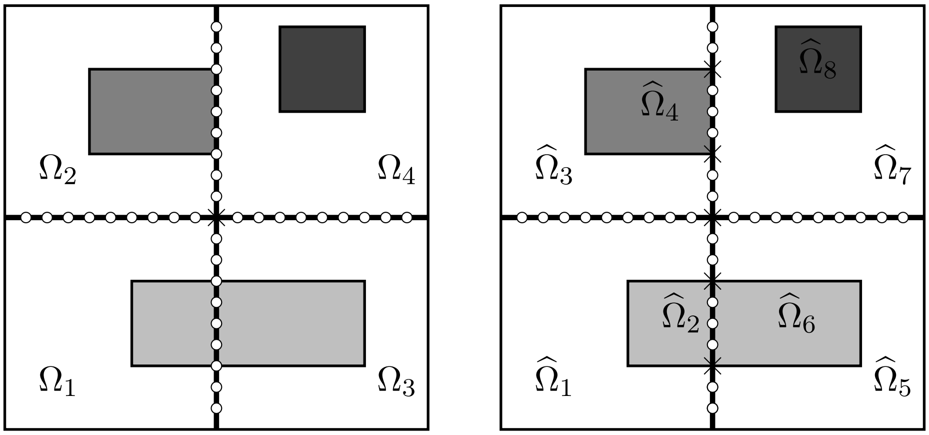


FIGURE 2. An example of how FE nodes (on the interface of the original partition $\Theta$ in Figure 1) are classified in the standard way (left) using $\text{neigh}_{\Theta}$, and in the physics-based way (right) using $\text{neigh}_{\Theta_{\rm pb}}$. Corner nodes are marked with crosses while nodes in edges are marked with small circles. Using the standard classification, on the left, we obtain $\Lambda(\Theta)$ with one corner and four edges. With the new classification, on the right, we have $\Lambda_{\rm pb}(\Theta)$ with five corners and six edges (eight edges if we only consider connected objects).

2.3. **Object classification.** This subsection concerns with objects on subdomain interfaces and their classification. It provides foundations for the definition of coarse DOFs in BDDC methods later on.

Given a subdomain partition $\Theta$, and a point $\xi \in \Gamma(\Theta)$, let us denote by $\text{neigh}_{\Theta}(\xi)$ the set of subdomains in $\Theta$ that contain $\xi$. We can introduce the concept of objects as a classification of points in $\Gamma(\Theta)$. A *geometrical object* is a maximal set $\lambda$ of points in $\Gamma(\Theta)$ with identical subdomain set. We denote by $\mathrm{neigh}_{\Theta}(\lambda)$ the set of subdomains in $\Theta$ containing $\lambda$. It should be noted that the set of all geometrical objects, denoted by $\Lambda(\Theta)$, is a partition of $\Gamma(\Theta)$.

**Remark 2.1.** *Since the set of points in the interface is infinite, the previous classification of* $\Gamma(\Theta)$ *into geometrical objects is performed in practice by the classification of vertices, edges, and faces of elements in the mesh* $\mathcal{T}$ *based on their subdomain set.*

Denote by $\mathrm{ndof}(\lambda)$ the number of DOFs belonging to $\lambda$. We further consider the following standard classification of geometrical objects. In the three-dimensional case, $\lambda \in \Lambda(\Theta)$ is a *face* if $|\mathrm{neigh}_{\Theta}(\lambda)| = 2$ and $\mathrm{ndof}(\lambda) > 1$, is an *edge* if $|\mathrm{neigh}_{\Theta}(\lambda)| > 2$ and $\mathrm{ndof}(\lambda) > 1$, and is a *corner* if $\mathrm{ndof}(\lambda) = 1$. In the two-dimensional case, $\lambda \in \Lambda(\Theta)$ is an *edge* if $|\mathrm{neigh}_{\Theta}(\lambda)| = 2$ and $\mathrm{ndof}(\lambda) > 1$, and is a *corner* if $\mathrm{ndof}(\lambda) = 1$. In the literature, e.g., [35, 54], corners are also referred to as vertices. Analogous definitions are also used frequently for FETI-DP methods (see [54]). In Figure 2 (left), an illustration of this classification is shown for a simple example.

In the next step, we define PB objects, which is the main ingredient of the PB-BDDC methods proposed herein. We consider the set of objects $\Lambda_{\mathrm{pb}}(\Theta)$ obtained by applying the previous classification of $\Gamma(\Theta)$ into corners/edges/faces but with $\mathrm{neigh}_{\Theta}(\cdot)$ replaced by $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\cdot)$. In other words, we use sets of subdomains in $\Theta_{\mathrm{pb}}$ to classify geometrical objects on $\Gamma(\Theta)$. Figure 2 (right) shows objects in $\Lambda_{\mathrm{pb}}(\Theta)$ for a simple example.

**Lemma 2.2.** $\Lambda_{\mathrm{pb}}(\Theta)$ *is a refinement of* $\Lambda(\Theta)$.

*Proof.* The statement holds if for every object $\lambda_{\mathrm{pb}} \in \Lambda_{\mathrm{pb}}(\Theta)$ there exists one and only one object $\lambda \in \Lambda(\Theta)$ containing it. Since all points in $\lambda_{\mathrm{pb}}$ belong to the same set of PB subdomains, $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\lambda)$, they are in the same set of subdomains in $\Theta$, namely $\{\omega(\hat{\mathcal{D}})\}_{\hat{\mathcal{D}}\in\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\lambda)}$. As a result, all these points belong to the same object in $\Lambda(\Theta)$. □

**Remark 2.3.** *In some cases, the DOF-based classification into corners, edges, and faces might need some modification in order to ensure well-posedness of the BDDC method with corner constraints only. This usually involves the use of a kernel detection mechanism (see, e.g., [49]). A new approach based on perturbations has recently been proposed in [9, 8], where the method is well-posed in all cases.*

## 3. PB-BDDC preconditioning

In this section, we present the PB-BDDC preconditioner. The basic idea behind BDDC methods is first to define a sub-assembled operator (no assembling among subdomains), and the global space of functions that are fully independent ("discontinuous") among subdomains. Secondly, we have to define the under-assembled space (the BDDC space) of functions for which continuity among subdomains is enforced only on a set of coarse DOFs. In order to be robust for heterogeneous problems, the PB-BDDC method utilises new definitions of the BDDC space (i.e., new coarse DOF continuity among subdomains) and a new weighting operator.

### 3.1. Coarse degrees of freedom.

Similarly to other BDDC methods, in the PB-BDDC method, some (or all) of the objects in $\Lambda_{\mathrm{pb}}(\Theta)$ are associated with a *coarse* DOF. We denote this set of objects by $\Lambda_O$ and call it the set of *primal* or coarse objects. Obviously, $\Lambda_O \subseteq \Lambda_{\mathrm{pb}}(\Theta)$. Typical choices of $\Lambda_O$ are $\Lambda_O \doteq \Lambda_C$, when only corners are considered, $\Lambda_O \doteq \Lambda_C \cup \Lambda_E$, when corners and edges are considered, or $\Lambda_O \doteq \Lambda_{\mathrm{pb}}(\Theta)$, when corners, edges, and faces are considered. These choices lead to three variants of the PB-BDDC method, referred to as PB-BDDC(c), PB-BDDC(ce) and PB-BDDC(cef), respectively. Figure 2 (right) actually shows the coarse objects of PB-BDDC(ce) for a simple 2D problem.

For a general coercive problem, given an object $\lambda \in \Lambda_O$, we denote by $n_\lambda$ the dimension of the space $\ker(\mathcal{A})|_\lambda$. We also consider a set of functions $\{r_l^\lambda\}_{l=1}^{n_\lambda} \subset \ker(\mathcal{A})$ such that their restrictions to $\lambda$, i.e., $\{r_l^\lambda|_\lambda\}_{l=1}^{n_\lambda}$, span $\ker(\mathcal{A})|_\lambda$. For Poisson's problem, $n_\lambda = 1$ and the constant function $r_1^\lambda = 1$ must be considered for all objects. For 3D linear elasticity, one can consider 6 rigid-body modes (see, for example, [54, 34]) as $\{r_l^\lambda|_\lambda\}_{l=1}^{n_\lambda}$. These rigid-body modes are linearly independent when restricted to faces. However, only 5 and 3 of them are linearly independent when restricted to straight edges and corners, respectively. The dimension $n_\lambda$, therefore, should be decided based on whether $\lambda$ is a face, an edge, or a corner.

We consider the following set of coarse DOFs:

$$c_\lambda^{\mathcal{D},l}(u_{\mathcal{D}}) \doteq \frac{\int_\lambda u_{\mathcal{D}} \cdot r_l^\lambda \, ds}{\int_\lambda r_l^\lambda \cdot r_l^\lambda \, ds}, \quad \text{for} \quad u_{\mathcal{D}} \in \mathbb{V}_{\mathcal{D}}, \quad l = 1, \ldots, n_\lambda.$$

When $\lambda$ is a corner, $c_\lambda^{\mathcal{D}}$ is simply the value of $u_{\mathcal{D}} \cdot r_l^\lambda$ at that corner. The set of all the constraints for a given subdomain $\mathcal{D}$ and object $\lambda \in \Lambda_O$ span a subspace of functionals $\mathcal{C}_\lambda^{\mathcal{D}} \doteq \operatorname{span}\{c_\lambda^{\mathcal{D},l}, \, l = 1, \ldots, n_\lambda\} \subset \mathbb{V}_{\mathcal{D}}'$.

Once we have defined the coarse DOFs, we can define the BDDC space as follows

$$\widetilde{\mathbb{V}} \doteq \{v \in \mathbb{V} : c_\lambda^{\mathcal{D},l}(v) = c_\lambda^{\mathcal{D}',l}(v), \, \forall \lambda \in \Lambda_O, \, \forall \mathcal{D}, \mathcal{D}' \in \operatorname{neigh}_\Theta(\lambda), \, l = 1, \ldots, n_\lambda\}, \tag{4}$$

i.e., the subspace of functions in $\mathbb{V}$ having continuous coarse DOFs values. Clearly, $\bar{\mathbb{V}} \subset \widetilde{\mathbb{V}} \subset \mathbb{V}$.

For BDDC methods, solving the coarse problem is usually the bottleneck (cf. [2, 3, 4, 9]). Therefore, it is of great interest to find a minimal set of coarse objects (the number of the coarse objects is the number of the coarse DOFs and also is the size of the coarse problem), so that BDDC methods can achieve their potential of fast convergence and perfect weak scalability. According to [35, 54], in the case where the physical coefficient in each subdomain is constant, the set of coarse objects only needs to guarantee the existence of the so-called *acceptable paths*. We need a similar concept here for the PB-BDDC method.

The definitions below follows [34]. For the rest of the subsection, we consider the 3D case. The 2D case follows a straightforward restriction where faces become edges and edges become corners.

**Definition 3.1** (Fully primal face)**.** *A face $\lambda \in \Lambda_{\text{pb}}(\Theta)$ is* fully primal *if there exists a set of functionals $\phi_m \in \mathcal{C}_\lambda$, for $m = 1, \ldots, n_\lambda$, such that, for any $\hat{\mathcal{D}} \in \operatorname{neigh}_{\Theta_{\text{pb}}}(\lambda)$ and $u_{\hat{\mathcal{D}}} \in \mathbb{V}_{\hat{\mathcal{D}}}$, the following properties are satisfied:*

$$|\phi_m(u_{\hat{\mathcal{D}}})|^2 \leq C \frac{1}{H(\hat{\mathcal{D}})} \left(1 + \log\left(\frac{H(\hat{\mathcal{D}})}{h(\hat{\mathcal{D}})}\right)\right) (|u_{\hat{\mathcal{D}}}|^2_{H^{\frac{1}{2}}(\lambda)} + \|u_{\hat{\mathcal{D}}}\|^2_{L^2(\lambda)}),$$

$$\phi_m(r_l^\lambda) = \delta_{ml} \quad \forall m, l = 1, \ldots, n_\lambda.$$

It is obvious to check that a primal face is fully primal. Now, we are in position to define the concept of acceptable path.

**Definition 3.2** (Acceptable path)**.** *Let $\Theta_{\text{pb}}^\partial$ be the set of PB subdomains $\hat{\mathcal{D}} \in \Theta_{\text{pb}}$ touching the interface $\Gamma(\Theta)$, i.e., $\partial\hat{\mathcal{D}} \cap \Gamma(\Theta) \neq \emptyset$. For two subdomains $\hat{\mathcal{D}}_a, \hat{\mathcal{D}}_b \in \Theta_{\text{pb}}^\partial$ that share at least one object $\lambda \in \Lambda_{\text{pb}}(\Theta)$, an* acceptable path *connecting them is a sequence $\{\hat{\mathcal{D}}_a = \hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \ldots, \hat{\mathcal{D}}_n = \hat{\mathcal{D}}_b\}$ of PB subdomains in $\Theta_{\text{pb}}$ such that*

$$\text{TOL} \, \alpha_k \geq R(k) \, \min(\alpha_a, \alpha_b), \quad 1 \leq k \leq n \tag{5}$$

*where* TOL *is some predefined tolerance and $R(k) = h(\hat{\mathcal{D}}_k)/H(\hat{\mathcal{D}}_k)$ if $\lambda$ is a corner and $R(k) = 1$ otherwise. The path can only pass from one subdomain to the next one through a primal face.*

**Assumption 3.3.** *Given a predefined tolerance* TOL *for the definition of acceptable paths, we assume that the set of PB-BDDC objects $\Lambda_O$ are such that for any pair of subdomains $\hat{\mathcal{D}}_a, \hat{\mathcal{D}}_b \in \Theta_{\text{pb}}^\partial$ such that $\omega(\hat{\mathcal{D}}_a) \neq \omega(\hat{\mathcal{D}}_b)$ sharing an object, then the object is primal or there is an acceptable path between the subdomains.*

**Remark 3.4.** *For $\hat{\mathcal{D}}_a$ and $\hat{\mathcal{D}}_b$ sharing a primal object in $\Lambda_O$, there is a trivial path with* TOL $= 1$*. If $\omega(\hat{\mathcal{D}}_a) = \omega(\hat{\mathcal{D}}_b)$, pointwise continuity is enforced between them or between subdomains connecting them to form a path. This is much stronger than having an acceptable path. In other words, we can say that there is at least one acceptable path with* TOL $= 1$ *between any pair of PB subdomains belonging to the same original subdomain.*

3.2. **Restriction operator.** Let us define the projection $\mathcal{Q} : \mathbb{V} \to \bar{\mathbb{V}}$ as some weighted average of interface values together with an $\alpha$-harmonic extension to subdomain interiors (see, e.g., [41]). We define these ingredients as follows.

For $u \in \mathbb{V}$, $\xi \in \Gamma(\Theta)$ and an associated PB partition $\Theta_{\rm pb}$ of $\Theta$, the weighting operator $\mathcal{W}$ is defined as

$$\mathcal{W}u(\xi) \doteq \sum_{\mathcal{D}\in \mathrm{neigh}_{\Theta}(\xi)} \delta^{\dagger}_{\mathcal{D}}(\xi)\, u_{\mathcal{D}}(\xi), \quad \text{with} \quad \delta^{\dagger}_{\mathcal{D}}(\xi) \doteq \frac{\sum_{\widehat{\mathcal{D}}\in \mathrm{neigh}_{\Theta_{\rm pb}}(\xi)\cap\Theta_{\rm pb}(\mathcal{D})} \alpha_{\widehat{\mathcal{D}}}}{\sum_{\widehat{\mathcal{D}}\in \mathrm{neigh}_{\Theta_{\rm pb}}(\xi)} \alpha_{\widehat{\mathcal{D}}}}, \tag{6}$$

where $\alpha_{\widehat{\mathcal{D}}}$ denotes the constant coefficient in the PB subdomain $\widehat{\mathcal{D}}$.

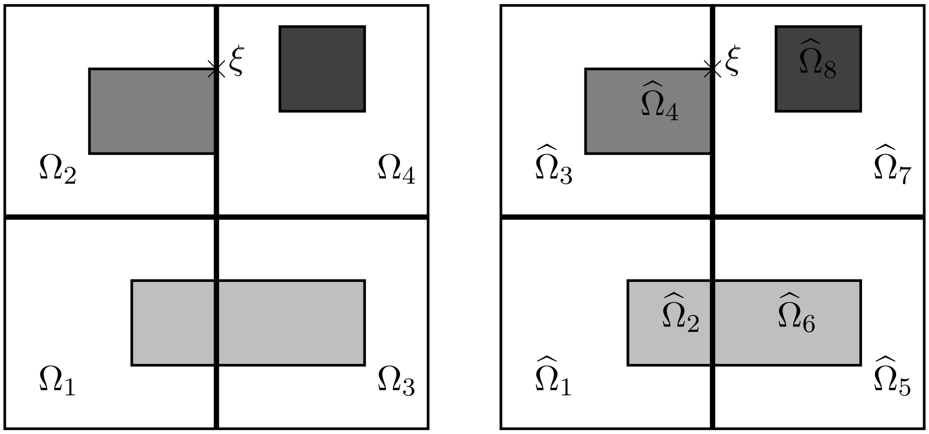


FIGURE 3. Illustration for the weighting $\mathcal{W}$ at $\xi$ on the interface $\Gamma_h(\Theta)$ of the original partition (left) with an associated PB partition (right).

We note that (6) defines the weighting for points on the original interface $\Gamma(\Theta)$. However, information of the associated PB partition $\Theta_{\rm pb}$ is also incorporated. For illustration, the action of $\mathcal{W}$ on $u \in \mathbb{V}$ at $\xi$ shown in Figure 3 is

$$\mathcal{W}u(\xi) = \frac{\hat\alpha_3 + \hat\alpha_4}{\hat\alpha_3 + \hat\alpha_4 + \hat\alpha_7}\, u_2(\xi) + \frac{\hat\alpha_7}{\hat\alpha_3 + \hat\alpha_4 + \hat\alpha_7}\, u_4(\xi),$$

where $u_i$ denotes the local "component" of $u$ associated with subdomain $\Omega_i$.

The $\alpha$-harmonic extension operator $\mathcal{E}$ taking data on the interface $\Gamma_h(\Theta)$ and $\alpha$-harmonically extending it to each subdomain $\mathcal{D} \in \Theta$ is formally defined as

$$\mathcal{E}u \doteq (1 - \mathcal{A}_0^{-1}\mathcal{A})u,$$

where $\mathcal{A}_0$ is the Galerkin projection of $\mathcal{A}$ onto the bubble space $\mathbb{V}_0 \doteq \{v \in \mathbb{V} : v = 0 \text{ on } \Gamma(\Theta)\}$.

We finally define $\mathcal{Q} = \mathcal{E}\mathcal{W}$.

3.3. **Preconditioner statement.** In this subsection, we present the PB-BDDC preconditioner, and describe its set-up and formulation. The PB-BDDC preconditioner is a BDDC preconditioner in which the set of coarse DOFs enforce continuity on a set of PB coarse objects, thus modifying the BDDC space being used. Once one has defined the set of PB coarse objects $\Lambda_O$, the rest of ingredients of the PB-BDDC preconditioner are identical to the ones of a standard BDDC preconditioner. In any case, the definition of the weighting operator introduced in (6) is new.

The BDDC preconditioner is a Schwarz-type preconditioner that combines interior corrections with corrections in the BDDC space (see, e.g., [11, 54]). In case of the PB-BDDC preconditioner, the BDDC correction is expressed as $\mathcal{Q}(\tilde{\mathcal{A}}^{\Theta})^{-1}\mathcal{Q}^T$, where $\tilde{\mathcal{A}}^{\Theta}$ is the Galerkin projection of $\mathcal{A}^{\Theta}$ onto $\widetilde{\mathbb{V}}$. More specifically, the PB-BDDC preconditioner reads as follows:

$$\mathcal{B} = \mathcal{A}_0^{-1} + \mathcal{Q}(\tilde{\mathcal{A}}^{\Theta})^{-1}\mathcal{Q}^T.$$

Apart from the task of identifying and defining coarse objects, the implementation of the PB-BDDC method is identical to that of the standard BDDC method. We refer the interested reader to [15, 16, 43, 11] for more details on the formulation of BDDC methods and to [2, 4, 6] for an efficient implementation of BDDC methods on distributed memory machines, which requires much further elaboration.

3.4. **Condition number estimates.** In order to prove condition number estimates for the PB-BDDC preconditioner, we first need to introduce $\widehat{\mathcal{B}}$, an auxiliary BDDC preconditioner. The definition of this preconditioner follows verbatim that of the PB-BDDC preconditioner. There are only changes in the partition, the weighting, and the coarse objects being used.

$\widehat{\mathcal{B}}$ is defined on the PB partition. Given the FE mesh $\mathcal{T}$, the FE space type, and the PB subdomain partition $\Theta_{\mathrm{pb}}$, one can similarly build the FE spaces and operators as in section 2.2, leading to the sub-assembled space $\mathbb{V}^{\mathrm{pb}}$ and operator $\mathcal{A}^{\Theta_{\mathrm{pb}}}$.

As for the weighting, $\widehat{\mathcal{W}}$, its rigorous formulation is given as follows. For $\hat{u} \in \mathbb{V}^{\mathrm{pb}}$, and $\xi \in \Gamma(\Theta_{\mathrm{pb}})$, the weighting operator $\widehat{\mathcal{W}}$ is defined as

$$\widehat{\mathcal{W}}\hat{u}(\xi) \doteq \sum_{\widehat{\mathcal{D}} \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi)} \delta^{\dagger}_{\widehat{\mathcal{D}}}(\xi)\, \hat{u}_{\widehat{\mathcal{D}}}(\xi), \quad \text{where } \delta^{\dagger}_{\widehat{\mathcal{D}}}(\xi) \doteq \frac{\alpha_{\widehat{\mathcal{D}}}}{\sum_{\widehat{\mathcal{D}} \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi)} \alpha_{\widehat{\mathcal{D}}}}. \tag{7}$$

For $\xi$ illustrated in Figure 3, this definition yields

$$\widehat{\mathcal{W}}\hat{u}(\xi) = \frac{\hat{\alpha}_3}{\hat{\alpha}_3 + \hat{\alpha}_4 + \hat{\alpha}_7}\, \hat{u}_3(\xi) + \frac{\hat{\alpha}_4}{\hat{\alpha}_3 + \hat{\alpha}_4 + \hat{\alpha}_7}\, \hat{u}_4(\xi) + \frac{\hat{\alpha}_7}{\hat{\alpha}_3 + \hat{\alpha}_4 + \hat{\alpha}_7}\, \hat{u}_7(\xi),$$

where $\hat{u}_i$ denotes the local "component" of $\hat{u}$ associated with subdomain $\widehat{\Omega}_i$. In other words, $\widehat{\mathcal{W}}$ is nothing but the pseudoinverse of a classical weighted counting function introduced in [35] (see also [54, 6.2.1]).

It is well known that $\delta^{\dagger}_{\widehat{\mathcal{D}}}$ is constant in each (PB) coarse object associated with $\widehat{\mathcal{D}}$ and the following important inequality, cf. [54, (6.19)], holds

$$\alpha_{\widehat{\mathcal{D}}_a} \left(\delta^{\dagger}_{\widehat{\mathcal{D}}_b}(\xi)\right)^2 \leq \min\left(\alpha_{\widehat{\mathcal{D}}_a}, \alpha_{\widehat{\mathcal{D}}_b}\right), \quad \forall \widehat{\mathcal{D}}_a, \widehat{\mathcal{D}}_b \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi). \tag{8}$$

**Lemma 3.5.** *The weighting* $\widehat{\mathcal{W}}$ *and* $\mathcal{W}$ *satisfy*

$$\widehat{\mathcal{W}}u(\xi) = \mathcal{W}u(\xi) \quad \textit{for all } u \in \mathbb{V},\ \xi \in \Gamma(\Theta).$$

*Proof.* The proof comes directly from (6), (7), and the fact that $v_{\widehat{\mathcal{D}}_a}(\xi) = v_{\widehat{\mathcal{D}}_b}(\xi) = v_{\mathcal{D}}(\xi)$ for any $\xi \in \Gamma(\Theta)$, $v \in \mathbb{V} \subset \mathbb{V}^{\mathrm{pb}}$, and $\widehat{\mathcal{D}}_a, \widehat{\mathcal{D}}_b \in \Theta_{\mathrm{pb}}(\mathcal{D})$ such that $\xi \in \widehat{\mathcal{D}}_a, \widehat{\mathcal{D}}_b$. Using these facts, we find that

$$\begin{aligned} \mathcal{W}v(\xi) &= \sum_{\mathcal{D} \in \mathrm{neigh}_{\Theta}(\xi)} \delta^{\dagger}_{\mathcal{D}}(\xi)\, v_{\mathcal{D}}(\xi) = \sum_{\mathcal{D} \in \mathrm{neigh}_{\Theta}(\xi)} \sum_{\widehat{\mathcal{D}} \in \Theta_{\mathrm{pb}},\, \widehat{\mathcal{D}} \ni \xi} \delta^{\dagger}_{\widehat{\mathcal{D}}}(\xi)\, v_{\mathcal{D}}(\xi) \\ &= \sum_{\hat{\mathcal{D}} \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi)} \delta^{\dagger}_{\hat{\mathcal{D}}}(\xi)\, v_{\hat{\mathcal{D}}}(\xi) = \widehat{\mathcal{W}}v(\xi). \end{aligned}$$

It ends the proof. □

With the definition of $\widehat{\mathcal{W}}$ in place, we can define $\widehat{\mathcal{Q}} = \widehat{\mathcal{E}}\widehat{\mathcal{W}}$, where $\widehat{\mathcal{E}}$ is the extension operator taking data on $\Gamma(\Theta_{\mathrm{pb}})$ and $\alpha$-harmonically extending to the interiors of each PB subdomain (cf. section 3.2).

The definition of the set of coarse objects of $\widehat{\mathcal{B}}$ requires further elaboration. The set of objects $\Lambda(\Theta_{\mathrm{pb}})$ obtained by applying the classification in section 2.3 for the PB subdomain partition $\Theta_{\mathrm{pb}}$ provides a classification of $\Gamma(\Theta_{\mathrm{pb}}) \supset \Gamma(\Theta)$. We have the following relation between the PB objects $\Lambda_{\mathrm{pb}}(\Theta)$ and the (standard) objects of the PB partition $\Lambda(\Theta_{\mathrm{pb}})$.

**Lemma 3.6.** *All the objects in* $\Lambda_{\mathrm{pb}}(\Theta)$ *are also in* $\Lambda(\Theta_{\mathrm{pb}})$*, i.e.,* $\Lambda_{\mathrm{pb}}(\Theta) \subset \Lambda(\Theta_{\mathrm{pb}})$.

*Proof.* Let us consider an object $\lambda_{\mathrm{pb}} \in \Lambda_{\mathrm{pb}}(\Theta)$. In both object partitions $\Lambda_{\mathrm{pb}}(\Theta)$ and $\Lambda(\Theta_{\mathrm{pb}})$, we are using the same criteria, i.e., $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\cdot)$, to classify points. The difference is that $\Lambda_{\mathrm{pb}}(\Theta)$ is the result of a classification of points in $\Gamma(\Theta)$ whereas $\Lambda(\Theta_{\mathrm{pb}})$ is obtained from a classification of points in $\Gamma(\Theta_{\mathrm{pb}})$. Since $\Gamma(\Theta) \subset \Gamma(\Theta_{\mathrm{pb}})$, all points in $\lambda_{\mathrm{pb}}$ belong to the same object $\lambda' \in \Lambda(\Theta_{\mathrm{pb}})$. Since $\lambda_{\mathrm{pb}}$ is on the interface $\Gamma(\Theta)$, there exist at least two subdomains $\hat{\mathcal{D}}, \hat{\mathcal{D}}' \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\lambda_{\mathrm{pb}})$ such that $\omega(\hat{\mathcal{D}}) \neq \omega(\hat{\mathcal{D}}')$. Let us assume there is a point $\xi \in \lambda'$ such that $\xi \notin \lambda_{\mathrm{pb}}$. Then, $\xi \in \Gamma(\Theta_{\mathrm{pb}}) \backslash \Gamma(\Theta)$, i.e., it only belongs to one subdomain in $\Theta$. As a result, $\omega(\hat{\mathcal{D}})$ is the same for all $\hat{\mathcal{D}} \in \mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi)$. Thus, we have a contradiction, since $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\xi)$ cannot be the same as $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(\lambda_{\mathrm{pb}})$. □

With the theoretical support from Lemma 3.6, we can define the set of coarse objects $\widehat{\Lambda}_O$ of $\widehat{\mathcal{B}}$ as a classification of $\Gamma(\Theta_{\mathrm{pb}})$ as follows. On $\Gamma(\Theta)$, we consider the same set of objects $\Lambda_O$ used in the PB-BDDC preconditioner, i.e., $\Lambda_C$, or $\Lambda_C \cup \Lambda_E$, or $\Lambda_{\mathrm{pb}}(\Theta)$. For the rest of the interface $\Gamma(\Theta_{\mathrm{pb}})\backslash\Gamma(\Theta)$, we enforce *full continuity* among PB subdomains. It can be understood as treating all FE nodes on $\Gamma(\Theta_{\mathrm{pb}})\backslash\Gamma(\Theta)$ as corners. Denote this set of objects by $\widehat{\Lambda}^*$, we have $\widehat{\Lambda}_O = \Lambda_O \cup \widehat{\Lambda}^*$. Figure 4 illustrates the partitions and coarse objects of $\mathcal{B}$ and $\widehat{\mathcal{B}}$ when $\Lambda_O = \Lambda_C \cup \Lambda_E$.

**Remark 3.7.** *By construction, the BDDC space $\widetilde{\mathbb{V}}^{\mathrm{pb}}$ of the auxiliary BDDC preconditioner $\widehat{\mathcal{B}}$ is the same as the BDDC space $\widetilde{\mathbb{V}}$, defined in* (4), *of the PB-BDDC preconditioner.*

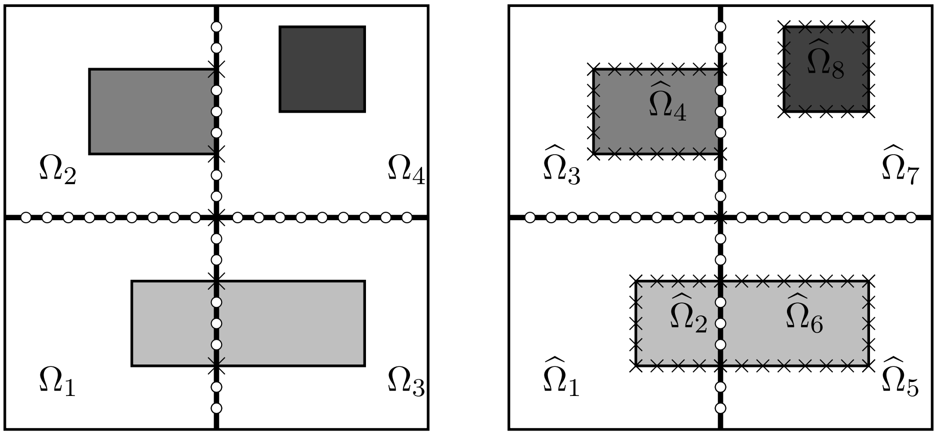


FIGURE 4. Partitions and coarse objects of the PB-BDDC preconditioner $\mathcal{B}$ (left) and the auxiliary BDDC preconditioner $\widehat{\mathcal{B}}$ (right) when $\Lambda_O = \Lambda_C \cup \Lambda_E$: corner objects are labeled with crosses while nodes of other objects are labeled with circles.

**Lemma 3.8.** *The condition number $\kappa(\mathcal{BA})$ of the PB-BDDC preconditioned operator is bounded by*

$$\kappa(\mathcal{BA}) \leq \max_{v\in\widetilde{\mathbb{V}}^{\mathrm{pb}}} \frac{\langle \mathcal{A}^{\Theta_{\mathrm{pb}}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle}{\langle \mathcal{A}^{\Theta_{\mathrm{pb}}}v, v\rangle}. \tag{9}$$

*Proof.* According to [43, Theorem 15] (see also [41]), $\kappa(\mathcal{BA})$ is bounded by

$$\kappa(\mathcal{BA}) \leq \max_{v\in\widetilde{\mathbb{V}}} \frac{\langle \mathcal{A}^{\Theta}\mathcal{Q}v, \mathcal{Q}v\rangle}{\langle \mathcal{A}^{\Theta}v, v\rangle}. \tag{10}$$

Now we only need to bound the right-hand-side in (10) by the one in (9).

On the one hand, using the fact that $\widetilde{\mathbb{V}} = \widetilde{\mathbb{V}}^{\mathrm{pb}}$ (cf. Remark 3.7) , we have $\langle \mathcal{A}^{\Theta}v, v\rangle = \langle \mathcal{A}^{\Theta_{\mathrm{pb}}}v, v\rangle$ for all $v \in \widetilde{\mathbb{V}}$ because any $v \in \widetilde{\mathbb{V}}$ is continuous in each subdomain of $\Theta$. On the other hand, let us prove that the weighting operator $\widehat{\mathcal{W}}$ defined by (6) for $\Theta_{\mathrm{pb}}$ restricted to $\mathbb{V}$ is identical to the weighting operator $\mathcal{W}$ defined by (6) for $\Theta$. Let us consider a subdomain $\mathcal{D} \in \Theta$ and its PB partition $\Theta_{\mathrm{pb}}(\mathcal{D})$. We have

$$\delta^{\dagger}_{\mathcal{D}}(\xi) = \sum_{\hat{\mathcal{D}}\in\Theta_{\mathrm{pb}}(\mathcal{D}),\, \hat{\mathcal{D}}\ni\xi} \delta^{\dagger}_{\hat{\mathcal{D}}}(\xi),$$

by the definition in (6). Using Lemma 3.5, we get that $\widehat{\mathcal{Q}}v$ and $\mathcal{Q}v$ are identical on $\Gamma(\Theta)$ and $\widehat{\mathcal{Q}}v$ is continuous across $\Gamma(\Theta_{\mathrm{pb}})$. In addition, $\mathcal{Q}v$ is discrete $\alpha$-harmonic in each $\mathcal{D} \in \Theta$ and has minimal energy norm with respect to $\mathcal{A}^{\Theta}$. As a consequence,

$$\begin{aligned}\langle \mathcal{A}^{\Theta}\mathcal{Q}v, \mathcal{Q}v\rangle &= \sum_{\mathcal{D}\in\Theta} \langle \mathcal{A}^{\Theta}_{\mathcal{D}}\mathcal{Q}v, \mathcal{Q}v\rangle \\ &\leq \sum_{\mathcal{D}\in\Theta} \langle \mathcal{A}^{\Theta}_{\mathcal{D}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle = \sum_{\hat{\mathcal{D}}\in\Theta_{\mathrm{pb}}} \langle \mathcal{A}^{\Theta_{\mathrm{pb}}}_{\hat{\mathcal{D}}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle = \langle \mathcal{A}^{\Theta_{\mathrm{pb}}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle.\end{aligned}$$

This finishes the proof. □

We could stop here and derive the estimate for $\kappa(\mathcal{B}\mathcal{A})$ knowing that the condition number of the auxiliary BDDC preconditioned operator $\widehat{\mathcal{B}\mathcal{A}}$ is estimated by an upper bound of the last quantity on the right of (10). However, we will go a bit further to obtain a stronger result.

**Lemma 3.9.** *Assume that $\Lambda_O$ is such that Assumption 3.3 holds. Then we have the following inequality:*

$$\max_{v\in\widetilde{\mathbb{V}}^{\text{pb}}} \frac{\langle \mathcal{A}^{\Theta_{\text{pb}}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle}{\langle \mathcal{A}^{\Theta_{\text{pb}}}v, v\rangle} \leq C\ \max\{1, \text{TOL}\} \max_{\mathcal{D}\in\Theta^{\partial}_{\text{pb}}} \left(1+\log\left(\frac{H(\mathcal{D})}{h(\mathcal{D})}\right)\right)^2,$$

*where the constant $C$ is independent of the number of subdomains, $H(\hat{\mathcal{D}})$, $h(\hat{\mathcal{D}})$ and the physical coefficient $\alpha$.*

*Proof.* By triangle inequality, we have

$$\max_{v\in\widetilde{\mathbb{V}}^{\text{pb}}} \frac{\langle \mathcal{A}^{\Theta_{\text{pb}}}\widehat{\mathcal{Q}}v, \widehat{\mathcal{Q}}v\rangle}{\langle \mathcal{A}^{\Theta_{\text{pb}}}v, v\rangle} \leq 1 + \max_{v\in\widetilde{\mathbb{V}}^{\text{pb}}} \frac{\langle \mathcal{A}^{\Theta_{\text{pb}}}(\widehat{\mathcal{Q}}v - v), (\widehat{\mathcal{Q}}v - v)\rangle}{\langle \mathcal{A}^{\Theta_{\text{pb}}}v, v\rangle}.$$

Let $w = \widehat{\mathcal{Q}}v - v$. Then its component on $\widehat{\mathcal{D}}_a$, $w_{\widehat{\mathcal{D}}_a}$, can be explicitly written as

$$w_{\widehat{\mathcal{D}}_a}(\xi) = \sum_{\widehat{\mathcal{D}}_b\in\text{neigh}_{\Theta_{\text{pb}}}(\xi)} \delta^{\dagger}_{\widehat{\mathcal{D}}_b}(\xi)\big(w_{\widehat{\mathcal{D}}_a}(\xi) - w_{\widehat{\mathcal{D}}_b}(\xi)\big).$$

Given a FE function $u \in \mathbb{V}_{\hat{\mathcal{D}}}$, we denote by $\theta^{\hat{\mathcal{D}}}_{\lambda}(u) \in \mathbb{V}_{\hat{\mathcal{D}}}$ the FE function that is discrete $\alpha$-harmonic in $\hat{\mathcal{D}}$ and agrees with $u$ at the FE nodes in the object $\lambda$ and vanishes at all the other nodes on $\partial\hat{\mathcal{D}}$. Since $\Lambda(\Theta_{\text{pb}})$ is a partition of $\Gamma(\Theta_{\text{pb}})$, we can split $w$ into object and subdomain contributions as follows:

$$w = \sum_{\lambda\in\Lambda(\Theta_{\text{pb}})} \sum_{\hat{\mathcal{D}}\in\text{neigh}_{\Theta_{\text{pb}}}(\lambda)} \theta^{\hat{\mathcal{D}}}_{\lambda}(w). \tag{11}$$

By construction of the set of objects $\widehat{\Lambda}_O = \Lambda_O \cup \widehat{\Lambda}^*$ and the definition of $\widetilde{\mathbb{V}}^{\text{pb}}$, $w$ vanishes at all coarse objects in $\widehat{\Lambda}^*$, i.e., at all FE nodes in $\Gamma(\Theta_{\text{pb}}) \setminus \Gamma(\Theta)$. Consequently, (11) can be simplified as follows:

$$w = \sum_{\lambda\in\Lambda_{\text{pb}}(\Theta)} \sum_{\hat{\mathcal{D}}\in\Theta^{\partial}_{\text{pb}}} \theta^{\hat{\mathcal{D}}}_{\lambda}(w).$$

When $\Lambda_O$ satisfies Assumption 3.3, the set of objects in $\widehat{\Lambda}_O$ also fulfills [54, Assumption 6.27]. Consequently, using the fact that $\delta^{\dagger}_{\widehat{\mathcal{D}}}$ is constant in each PB coarse object associated with $\widehat{\mathcal{D}}$ and (8), we can use the analysis in [34] for 3D linear elasticity (see also the proof in [54, Lemma 6.36] and in [35, Lemma 10] for the Poisson's problem) to obtain

$$\begin{aligned}
&\langle \mathcal{A}^{\Theta_{\text{pb}}}_{\hat{\mathcal{D}}}\, \theta^{\hat{\mathcal{D}}}_{\lambda}(w), \theta^{\hat{\mathcal{D}}}_{\lambda}(w)\rangle \\
&\leq C\ \max\{1, \text{TOL}\} \left(1+\log\left(\frac{H(\hat{\mathcal{D}})}{h(\hat{\mathcal{D}})}\right)\right)^2 \sum_{\hat{\mathcal{D}}\in\text{neigh}_{\Theta_{\text{pb}}}(\lambda)} \langle \mathcal{A}^{\Theta_{\text{pb}}}_{\hat{\mathcal{D}}} v, v\rangle,
\end{aligned}$$

for any $\hat{\mathcal{D}} \in \Theta^{\partial}_{\text{pb}}$ and $\lambda \in \Lambda(\Theta_{\text{pb}})$. Here the constant $C$ is independent of $H(\hat{\mathcal{D}})$, $h(\hat{\mathcal{D}})$ and the physical coefficient $\alpha$.

Adding up the estimate for all subdomains $\hat{\mathcal{D}} \in \Theta_{\text{pb}}$, we find that

$$\langle \mathcal{A}^{\Theta_{\text{pb}}} w, w\rangle \leq C\ \max\{1, \text{TOL}\} \max_{\hat{\mathcal{D}}\in\Theta^{\partial}_{\text{pb}}} \left(1+\log\left(\frac{H(\hat{\mathcal{D}})}{h(\hat{\mathcal{D}})}\right)\right)^2 \langle \mathcal{A}^{\Theta_{\text{pb}}} v, v\rangle.$$

This finishes the proof. □

Combining results in Lemma 3.8 and Lemma 3.9, we have the final bound for the PB-BDDC preconditioner, which is both weakly scalable and independent of the coefficient $\alpha$.

**Theorem 3.10.** *If the set of PB-BDDC coarse objects $\Lambda_O$ satisfies Assumption 3.3, then the condition number of the PB-BDDC preconditioned operator $\kappa(\mathcal{BA})$ is bounded by*

$$\kappa(\mathcal{BA}) \leq C \ \max\{1, \mathrm{TOL}\} \max_{\hat{\mathcal{D}} \in \Theta_{\mathrm{pb}}^{\partial}} \left(1 + \log\left(\frac{H(\hat{\mathcal{D}})}{h(\hat{\mathcal{D}})}\right)\right)^2,$$

*where the constant $C$ is independent of the number of subdomains, $H(\hat{\mathcal{D}})$, $h(\hat{\mathcal{D}})$ and the physical coefficient $\alpha$.*

**Remark 3.11.** *As seen in Lemma 3.9 and Theorem 3.10, the condition number associated with the PB-BDDC method depends only on the characteristic size and mesh size of PB subdomains touching the original interface $\Gamma(\Theta)$. Further, the convergence of PB-BDDC is independent of variations of the coefficient. The main target of this work is achieved.*

**Remark 3.12.** *As PB partition and PB subdomains are defined mainly based on the variation of the coefficient, a natural question that arises is whether the convergence of the PB-BDDC preconditioner depends on the shapes of PB subdomains. It is well-known that subdomains with bad aspect ratio [24] and irregular subdomains with ragged boundary do have bad effect on domain decomposition preconditioners [33]. For PB-BDDC, such effect is negligible. The explanation is two-fold. First, the PB partition is not unique, i.e., we could split any PB subdomains into smaller PB subdomains as soon as $\Lambda_{\mathrm{pb}}(\Theta)$ remains the same, leading to the* same *preconditioner. Therefore, PB subdomains with bad aspect ratio or irregular boundary can be broken into smaller PB subdomains with good aspect ratio and more regular boundary and ideally consider the one that minimizes the constants in Theorem 3.10. Second, in the auxiliary preconditioner $\widehat{\mathcal{B}}$ for which the classical domain decomposition theory in [54] is applied, all of the boundary FE nodes on the PB interface but not on the original interface are treated as corners. Therefore, constraints on the aspect ratio as well as the regularity of the boundary of these PB subdomains do not affect the performance of the preconditioner. This will be later confirmed by the numerical experiment in section 5.1.4.*

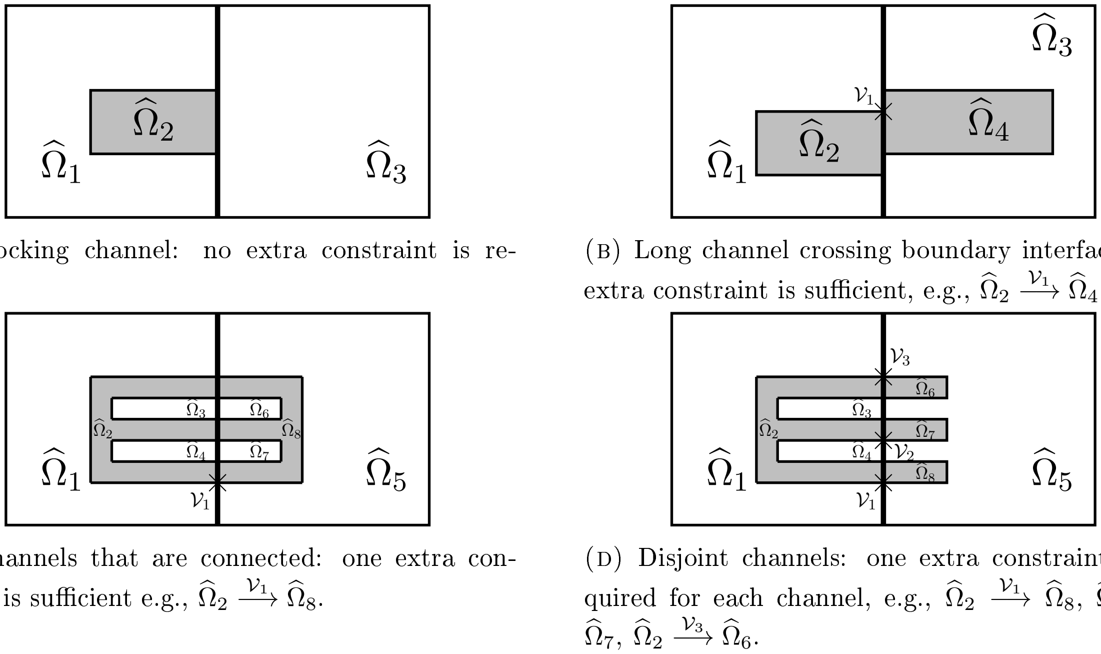
## 4. Reducing the size of the coarse space in PB-BDDC

4.1. **Some explanatory examples.** If all the constraints associated with the PB partition are used, we might end up using too many constraints and the size of the coarse space can be quite large. This could affect the overall performance of the PB-BDDC method since the coarse problem is usually the bottleneck in large scale simulations (cf. [2, 3, 4, 9]). However, according to Theorem 3.10, *not all the constraints associated with the PB partition are required to guarantee robustness.* We only need to choose the set of PB-BDDC coarse objects $\Lambda_O$ so that Assumption 3.3 holds. A simple option is to use, e.g., PB-BDDC(f), i.e., considering all faces as primal, since Assumption 3.3 readily holds. However, care must be taken as Assumption 3.3 might hold, but only for TOL proportional to the constrast of physical values (see [54, 6.4.5]). In the following, we will discuss a more elaborated approach.

Roughly speaking, for any two PB subdomains that share at least one point on the original subdomain interface, we need at least one acceptable path between them. From Definition 3.2, it implies that for any pair of PB subdomains, if one or both of them have a small coefficient, then the other neighboring PB subdomains with larger or equal coefficients can be included in the acceptable path without increasing the tolerance TOL. In other words, one only needs to worry about pairs of connected PB subdomains with large coefficients. There can be as many inclusions and channels as wanted inside a subdomain and not a single extra PB object is required. We note that these inclusions and channels can touch but not cross the original interface.

In Fig. 5, we illustrate some common scenarios of the coefficient and indicate the number of extra constraints, apart from the ones needed for well-posedness, required to guarantee the existence of acceptable paths. For simplicity of exposition, for the rest of this section, we only consider Poisson's problem. Linear elasticity will be considered in the next section. In addition, we only use channels for the clarity of the illustration even though the procedure can be applied to general scenarios. For docking channels that do not cross the subdomain interface as in Figure 5a, no extra constraint is required provided that the original subdomains are *connected*, i.e., there is at least an acceptable path that connects them. For a long channel crossing the subdomain interface (see Figure 5b), one extra constraint is required. This constraint can be a vertex, an edge, or a face constraint. For channels that are connected inside subdomains, as illustrated in Figure 5c, they all belong to the same PB subdomain and only one extra vertex, edge or face constraint is sufficient (due to the weaker restriction on face-connected subdomains in Figure 3.3). When channels are disjoint even in one of the subdomains only, we need at least one extra constraint for each non-connected channel, because every channel of this type leads to a different PB subdomain. This is illustrated in Figure 5d.

In summary, we need one extra PB constraint for every non-connected channel of high coefficient crossing the original interface. This is similar to spectral-based approaches of adaptive coarse spaces, e.g., [52, 31, 27], where the number of eigenfunctions needed is the same as the number of high coefficient channels crossing the interface. The conclusion will be numerically verified in section 5.1.1 and section 5.1.3.



(A) Docking channel: no extra constraint is required.

(B) Long channel crossing boundary interface: one extra constraint is sufficient, e.g., $\widehat{\Omega}_2 \xrightarrow{\mathcal{V}_1} \widehat{\Omega}_4$.

(C) Channels that are connected: one extra constraint is sufficient e.g., $\widehat{\Omega}_2 \xrightarrow{\mathcal{V}_1} \widehat{\Omega}_8$.

(D) Disjoint channels: one extra constraint is required for each channel, e.g., $\widehat{\Omega}_2 \xrightarrow{\mathcal{V}_1} \widehat{\Omega}_8$, $\widehat{\Omega}_2 \xrightarrow{\mathcal{V}_2} \widehat{\Omega}_7$, $\widehat{\Omega}_2 \xrightarrow{\mathcal{V}_3} \widehat{\Omega}_6$.

Figure 5. Some scenarios of coefficient distribution and the minimal number of extra constraints required.

In practice, it can be complicated to select *the minimal* set of constraints. However, it is not difficult to choose a reasonably small set of constraints that is sufficient for PB-BDDC to be $\alpha$-robust. For example, we can consider only PB subdomains touching the original interface and having coefficient relatively larger than the smallest coefficient of the neighboring PB subdomains in the same original subdomain. For two PB subdomains that are of this type and belong to two different original subdomains, we will select a vertex, an edge, or a face constraint to "connect" them if they share at least a point. For simplicity, vertex constraints only can be used in 2D. This approach is utilized in our numerical experiments with Poisson's problem in section 5.

4.2. **PB-BDDC(f-min): an automatic face-based algorithm.** In order to make the PB-BDDC algorithm competitive, especially for general 3D linear elasticity problems, one needs to define an effective automatic algorithm that will generate a *small* coarse space, i.e., with a dimension close to the minimum number of coarse DOFs needed to satisfy Assumption 3.3. Since the computation of the optimal is impractical and can involve long acceptable paths, our main target is to define a heuristic but scalable and effective algorithm.

We consider herein an algorithm that *relies on face constraints.* This choice is motivated by the fact that the acceptable path relies on fully primal faces and that primal faces are automatically fully primal.[1] Thus, the BDDC problem is well-posed without the need to deal with complex kernel detection mechanisms.

[1] We note that the definition of constraint sets in 3D linear elasticity for FETI-DP methods in the seminal work [34] rely on edge constraints.

However, for heterogeneous problems, face-only algorithms are not robust in all cases, e.g., when inclusions with high coefficients only share an edge (see Fig. 5.2 in [34] for an illustration).

We provide the following algorithm to automatically determine a small face-based coarse space satisfying Assumption 3.3. In the following algorithm, we work with an undirected graph $G$ (to be built during the course of the algorithm) where the vertices are the PB subdomains in $\Theta_{\mathrm{pb}}$ and an edge connecting two vertices means that there is an acceptable path between the PB subdomains.

*Part 1: Determine subdomain interior connections.* Every subdomain $\mathcal{D} \in \Theta$ is composed by a set of PB subdomains $\Theta_{\mathrm{pb}}(\mathcal{D})$. We consider that two PB subdomains in $\Theta_{\mathrm{pb}}(\mathcal{D})$ are connected if they share a face, i.e., $\hat{\mathcal{D}}_a \cap \hat{\mathcal{D}}_b$ is of dimension $d-1$. We note that full continuity is enforced on all these faces in PB-BDDC, and thus these faces are fully primal and the PB subdomains have an acceptable path. In a first step, we introduce all these connections in the graph $G$ for all subdomains in $\Theta$. This work is embarrassingly parallel.

*Part 2: Determine face connections.* For a face $F \in \Lambda(\Theta)$ shared by subdomains $\mathcal{D}_1, \mathcal{D}_2 \in \Theta$, we denote by $\Lambda_{\mathrm{pb}}(F)$ all the PB objects $\lambda \in \Lambda_{\mathrm{pb}}(\Theta)$ such that $\lambda \subseteq F$. For every PB face $f \in \Lambda_{\mathrm{pb}}(\lambda)$ and the respective PB subdomains $\hat{\mathcal{D}}_1$ and $\hat{\mathcal{D}}_2$ sharing this face, we compute the value $\min(\alpha_{\hat{\mathcal{D}}_1}, \alpha_{\hat{\mathcal{D}}_2})$ and $\max(\alpha_{\hat{\mathcal{D}}_1}, \alpha_{\hat{\mathcal{D}}_2})$. We sort these faces first with respect to the first value and next to the second one in decreasing order.[2] (If there are still ties, one can consider, e.g., the face with maximum area and finally the minimum global Id.) Thus, the set of faces in $\Lambda_{\mathrm{pb}}(F)$ are ordered and can be represented with $f_1, \ldots, f_{n_F}$. We define $f_1$ as coarse DOF, i.e., it is included in $\Lambda_O$, and mark the PB subdomains sharing this face as connected in $G$, since there is a trivial acceptable path between them. We take $i = 2$, and check whether these two subdomains have an acceptable path *restricted* to PB subdomains in $\Theta_{\mathrm{pb}}(\mathcal{D}_1) \cup \Theta_{\mathrm{pb}}(\mathcal{D}_2)$, i.e., two consecutive subdomains are connected in $G$ and their physical values satisfy (5).[3] If there is no acceptable path, we consider the face as coarse DOF. We mark as connected these two subdomains in $G$ in any case. We proceed for $i = i + 1$, till traversing all PB faces. We proceed for all faces in $\Lambda(\Theta)$.

To have at every processor all the connections of $G$ for PB subdomains in $\Theta_{\mathrm{pb}}(\mathcal{D}_1) \cup \Theta_{\mathrm{pb}}(\mathcal{D}_2)$, nearest-neighbor communications are performed right before Part 1. We note that every processor only needs to perform this work for its faces to have all the connections of its PB subdomains. With this information, the seek of acceptable paths is embarrassingly parallel.

*Part 3: Adding coarse edges/corners constraints.* Now, for every PB edges/corners $e \in \Lambda_{\mathrm{pb}}(\Theta)$, we check whether there is a trivial acceptable path, *restricted* to PB subdomains in $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(e)$. If not, the edge/corner is added to the set of coarse DOFs $\Lambda_O$. As before, this step involves a nearest-neighbor communication to have at every processor all the connections of $G$ (in the status corresponding to the end of Part 2) for PB subdomains in $\mathrm{neigh}_{\Theta_{\mathrm{pb}}}(e)$, whereas the seek of acceptable paths is embarrassingly parallel.

**Proposition 4.1.** *The coarse DOFs $\Lambda_O$ obtained with the previous algorithm satisfies Assumption 3.3.*

*Proof.* First, full continuity is enforced for interior PB objects. By construction, all PB subdomains sharing faces of $\Lambda(\Theta)$ have an acceptable path by construction in Step 2. Finally, PB subdomains only sharing corners/edges also have an acceptable path by construction in Step 3. □

**Remark 4.2.** *Less restrictive algorithms that would look for connections in larger patches could be considered. It would reduce the coarse space but it would involve longer paths, deteriorating convergence. It would also incur in more communications between processors and memory consumption.*

**Remark 4.3.** *It can easily be checked that this automatic algorithm leads to the same number of coarse constraints as for the cases in Fig. 5, the only difference being that face (edge in 2D) constraints instead of vertices, would be considered.*

[2] This choice is motivated by (5) in the definition of the acceptable path.

[3] Clearly, by restricting the connections to PB subdomains in $\Theta_{\mathrm{pb}}(\mathcal{D}_1) \cup \Theta_{\mathrm{pb}}(\mathcal{D}_2)$, we restrict the length of the acceptable paths, not attaining the minimum coarse space. However, it is essential to expose concurrency in the implementation of the algorithm by controlling the length of the acceptable paths.

## 5. Numerical experiments

In this section, we test the robustness and efficiency of the PB-BDDC preconditioner for the system matrix associated with (3) for different types of variation in the coefficient $\alpha$, which are similar but generally harder than the ones in [45, 31, 39].

5.1. **Experiments in 2D.** We consider the problem (1) with homogeneous Dirichlet boundary condition and the forcing term $f = 1$. In most cases, the physical domain is $\Omega = (0,1)^2$. Unless otherwise stated, we use uniform triangular meshes of size $h = 1/72$ and a regular $3 \times 3$ subdomain partition. We report the dimension of the coarse space, denoted by dim, and the number of iterations required for the preconditioned conjugate gradients method to reduce the residual norm by a factor of $10^6$. We also provide the computed (not estimated) condition number $\kappa$ of the preconditioned operator in most examples.

We will compare the performance of BDDC(ce), the standard BDDC method with all possible subdomain corner and edge coarse objects, with that of PB-BDDC(ce), PB-BDDC(e) and PB-BDDC([c]), three variants of the PB-BDDC method where all possible physics-based corner and edge or edge coarse objects, or selected physics-based corner coarse objects are used. No corner detection mechanism (see, e.g., [49]) has been needed. Alternatively, one might want to consider the perturbed formulation introduced in [8, 9].

5.1.1. *Two channels.* In this test case, we consider two channels of high $\alpha$ cutting through vertical subdomain edges (see Figure 6). The coefficient in the channels $\alpha_{\max}$ takes the values $\{10^2, 10^4, 10^6, 10^8\}$, while the coefficient in the rest of the domain is equal to 1.

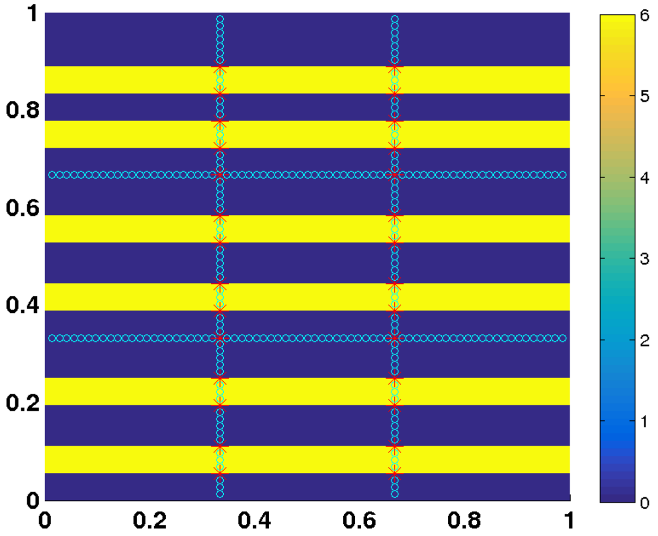


Figure 6. Two channels test case: the coefficient distribution when $\alpha_{\max} = 10^6$. The coarse objects used by PB-BDDC(ce) are shown on the interface with corners labeled by stars and DOFs in edges labeled by circles.

From Table 1, we can see that the condition number and the number of iterations for the standard BDDC preconditioner, BDDC(ce), definitely increase with $\alpha_{\max}$, whereas they remain practically constant for the three variants of the PB-BDDC preconditioner: PB-BDDC(ce), PB-BDDC(e) and PB-BDDC([c]). In other words, the convergence of the PB-BDDC method is independent of the contrast and the PB-BDDC method is perfectly robust for this test case.

In addition, we want to emphasize that PB-BDDC([c]) can deliver perfect robustness with a coarse space of the same dimension as the coarse space of the standard method BDDC(ce). The coarse objects of PB-BDDC(ce) and PB-BDDC([c]) are illustrated in Figure 6 and Figure 7 respectively.

5.1.2. *Channels and inclusions.* In this test case, we consider both channels and inclusions of high coefficient. First, the three channels include all the elements whose centroids are less than $2 \cdot 10^{-2}$ from one of

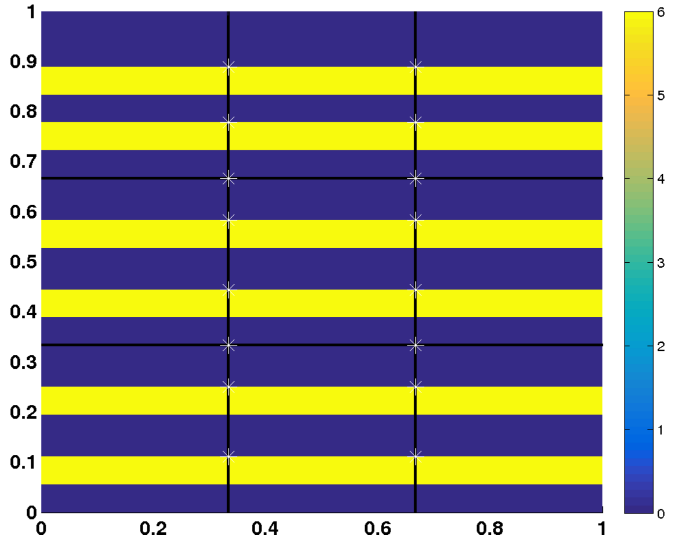


FIGURE 7. Two channels test case: only a small number of critical corners, and subdomain corners are required by PB-BDDC([c]) to have perfect robustness. These corners are shown on the interface as stars.

TABLE 1. Comparison of the iteration count and condition number in the two channels test case.

| | BDDC(ce) | PB-BDDC(ce) | PB-BDDC(e) | PB-BDDC([c]) |
|---|---|---|---|---|
| dim | 16 | 64 | 36 | 16 |
| $\alpha_{\max}$ | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) |
| $10^2$ | 9(8.68e2) | 4 (1.38e0) | 5 (3.01e0) | 6 (3.77e0) |
| $10^4$ | 13(1.83e5) | 4 (1.36e0) | 5 (2.92e0) | 5 (2.25e0) |
| $10^6$ | 17(3.61e5) | 4 (1.36e0) | 5 (2.92e0) | 5 (2.24e0) |
| $10^8$ | 22(4.05e7) | 4 (1.36e0) | 5 (2.92e0) | 5 (2.24e0) |

the following three lines:

$$\begin{aligned} L1 &: x_1 - x_2 - 0.2 = 0, \\ L2 &: x_1 + x_2 - 0.7 = 0, \\ L3 &: x_1 - 0.7x_2 - 0.7 = 0. \end{aligned}$$

The coefficient $\alpha_{\max}$ in these channels takes the values $\{10^2, 10^4, 10^6, 10^8\}$. Secondly, the inclusions are defined as the regions of elements whose all vertices $x$ satisfy

$$\text{mod}\big(\text{floor}(10\,x_i), 2\big) = 1, \text{ for } i = 1, 2.$$

For an element $\tau$ that belongs to one of the inclusions and is not in the channels, its coefficient is defined as

$$\alpha|_\tau = (\alpha_{\max}/10)^{1/5*\text{floor}(0.5*\text{floor}(10\,x_1(c_\tau))+1}, \quad \text{where } c_\tau \text{ is the centroid of } \tau. \tag{12}$$

The coefficient in (12) is: a) constant in each inclusion; b) increasing from left to right; c) increasing as $\alpha_{\max}$ increases; and d) always belongs to $(1, \alpha_{\max})$. For the rest of the domain, we set $\alpha = 1$. The maximal contrast ratio in this experiment is $10^8$. The coefficient distribution when $\alpha = 10^6$ is shown in Figure 8.

We can see from Table 2 that as $\alpha_{\max}$ becomes larger the condition number and the number of iterations associated with the standard BDDC(ce) method increases significantly. In contrast, both variants of the PB-BDDC method, PB-BDDC(ce) and PB-BDDC(e), are perfectly robust with respect to the changes of the coefficient in the channels and in the inclusions. Especially, PB-BDDC(e) maintains its robustness with a reasonably small coarse space (see also Figure 9).

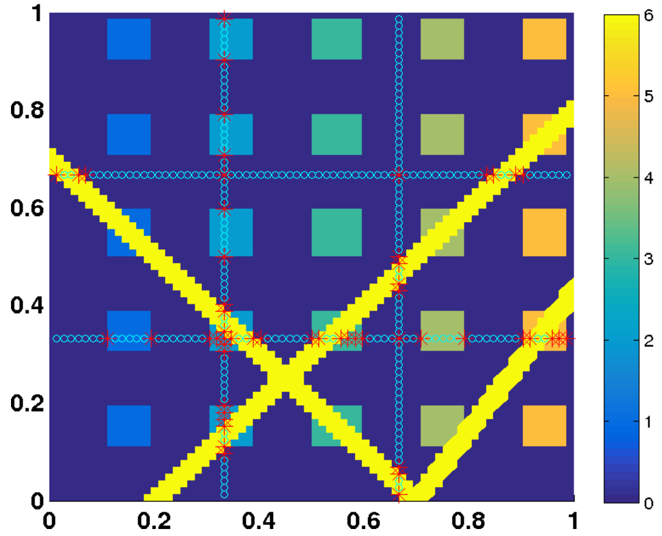


Figure 8. Channels and inclusions test case: the coefficient distribution when $\alpha_{\max} = 10^6$. The coarse objects used by PB-BDDC(ce) are shown on the interface with corners labeled by stars and DOFs in edges labeled by circles.

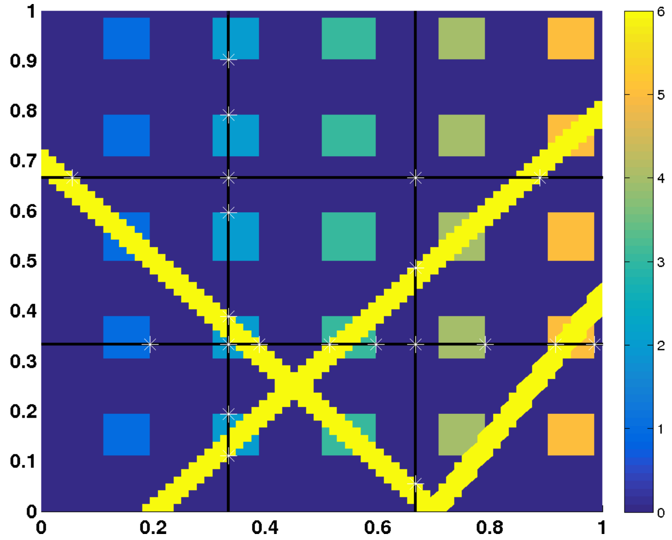


Figure 9. Channels and inclusions test case: only a small number of critical corners, and subdomain corners are required by PB-BDDC([c]) to have $\alpha$-robustness. These corners are shown on the interface as stars.

Table 2. Comparison of the iteration count and condition number in the channels and inclusions test case.

|  | BDDC(ce) | PB-BDDC(ce) | PB-BDDC(e) | PB-BDDC([c]) |
|---|---|---|---|---|
| dim | 16 | 89 | 39 | 19 |
| $\alpha_{\max}$ | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) |
| $10^2$ | 10(2.57e5 ) | 6 (1.91e0) | 10 (4.84e1) | 12 (2.25e2) |
| $10^4$ | 24(1.79e7 ) | 6 (1.99e0) | 10 (7.00e1) | 13 (3.44e2) |
| $10^6$ | 36(1.54e9 ) | 6 (2.04e0) | 11 (7.03e1) | 14 (3.49e2) |
| $10^8$ | 61(1.47e11) | 6 (2.04e0) | 11 (7.03e1) | 14 (3.50e2) |

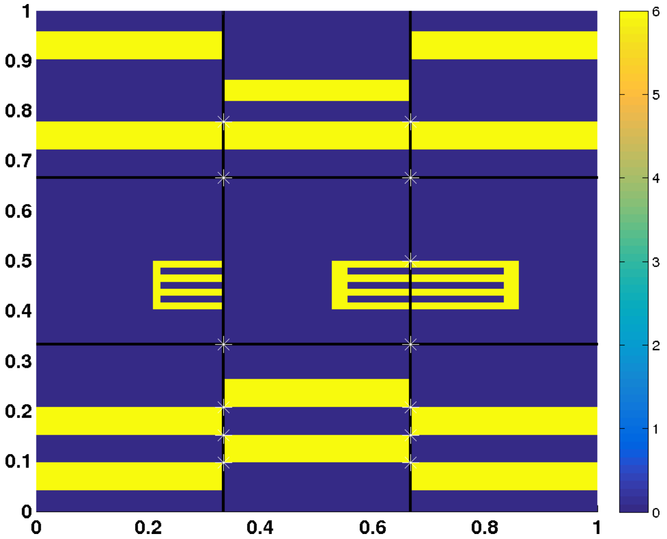


FIGURE 10. Complex channels test case: the coefficient distribution when $\alpha_{\max} = 10^6$. Only few corners are required by PB-BDDC([c]) to guarantee perfect robustness (to have acceptable paths). These corners are shown on the interface as stars.

5.1.3. *Complex channels.* In this test case, we demonstrate the importance of having acceptable paths. We consider a distribution with multiple channels of high coefficient $\alpha_{\max}$ taking values in $\{10^2, 10^4, 10^6, 10^8\}$ (see Figure 10 for the case when $\alpha_{\max} = 10^6$).

From Table 3, we can see that PB-BDDC(ce) is perfectly robust. On the other hand, the condition number and number of iterations of the PB-BDDC(e) preconditioner increase significantly as $\alpha_{\max}$ increases. The reason is that there are some pairs of channels sharing a corner but not an edge. In PB-BDDC(e), none of these corners are selected as a coarse object. Consequently, there is no acceptable path with TOL independent of the contrast between the associated paired of channels (PB subdomains) and Assumption 3.3 does not hold. By using a small number of these critical corners in order to satisfy Assumption 3.3 and the subdomain corners to guarantee invertibility, the resulting preconditioner, PB-BDDC([c]), is perfectly robust with respect to changes in the contrast of the coefficient (see Table 3).

TABLE 3. Comparison of the iteration count and condition number in the complex channels test case.

| | BDDC(ce) | PB-BDDC(ce) | PB-BDDC(e) | PB-BDDC([c]) |
|---|---|---|---|---|
| dim | 16 | 78 | 46 | 13 |
| $\alpha_{\max}$ | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) |
| $10^2$ | 16 (2.40e3) | 6 (2.75e0) | 14 (1.42e3) | 10 (8.79e1) |
| $10^4$ | 26 (4.22e5) | 6 (2.82e0) | 19 (1.96e5) | 10 (9.83e1) |
| $10^6$ | 38 (4.25e7) | 6 (2.82e0) | 25 (1.97e7) | 10 (9.84e1) |
| $10^8$ | 55 (4.93e9) | 6 (2.82e0) | 29 (1.97e9) | 10 (9.84e1) |

5.1.4. *Special PB subdomains with bad aspect ratio or irregular boundary.* In this test case, we check whether PB subdomains with bad aspect ratios or irregular boundary could jeopardize the convergence of the PB-BDDC preconditioner. We consider several regions of high coefficient $\alpha$ including long and thin channels thickness $h$ and length $H$, and regions with ragged boundary. (see Figure 11). The coefficient $\alpha_{\max}$ of these regions takes the values $\{10^2, 10^4, 10^6, 10^8\}$, while the coefficient in the rest of the domain is equal to 1.

From Table 4, we can see that the condition number and the number of iterations for BDDC(ce) and PB-BDDC(e) definitely increase with $\alpha_{\max}$, whereas they remain practically constant for PB-BDDC(ce) and PB-BDDC([c]). So as long as Assumption 3.3 on the existence of acceptable paths holds, PB-BDDC

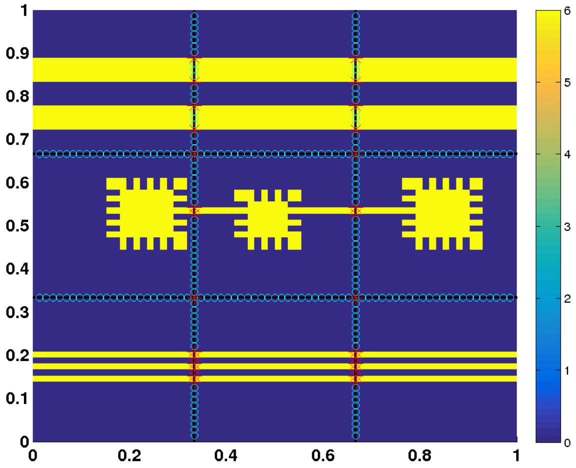


Figure 11. Special PB subdomains test case: the coefficient distribution when $\alpha_{\max} = 10^6$. The coarse objects used by PB-BDDC(ce) are shown on the interface with corners labeled by stars and DOFs in edges labeled by circles.

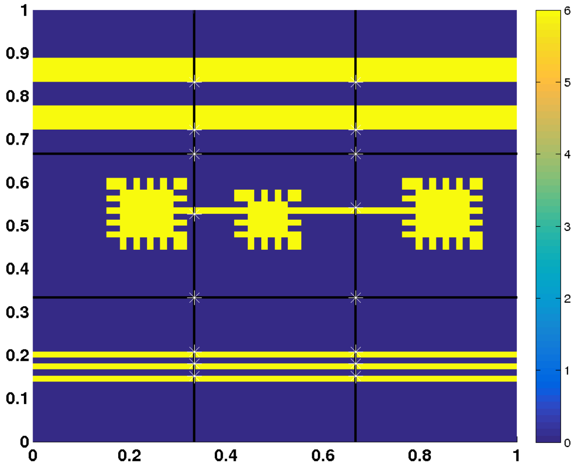


Figure 12. Special PB subdomains test case: only a small number of critical corners, and subdomain corners are required by PB-BDDC([c]) to have perfect robustness. These corners are shown on the interface as stars.

preconditioner is robust. More importantly, compared with results in experiments in section 5.1.1 and section 5.1.3, we can conclude that PB subdomains with bad aspect ratio or irregular boundary do not or have a very mild influence on the convergence of PB-BDDC preconditioner.

5.2. **Experiments in 3D.** For the set of experiments presented in this section, an HPC implementation of the PB-BDDC preconditioner was developed using the tools provided by FEMPAR [1, 7]. FEMPAR is a parallel hybrid MPI+OpenMP, object-oriented (OO) software package for the massively parallel Finite Element (FE) simulation of multiphysics problems governed by PDEs, developed by the members of the LSSC team at CIMNE. FEMPAR is released under the GNU GPL v3 license, and its first public release has almost 300K lines of (mostly) Fortran 95/200X code.

A major and unique (compared to other packages available on the Internet) cornerstone of FEMPAR is an abstract OO framework for the implementation of widely applicable highly scalable multilevel DD

Table 4. Comparison of the iteration count and condition number in the special PB subdomains test case.

| | BDDC(ce) | PB-BDDC(ce) | PB-BDDC(e) | PB-BDDC([c]) |
|---|---|---|---|---|
| dim | 16 | 52 | 24 | 16 |
| $\alpha_{\max}$ | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) | # it. ($\kappa$) |
| $10^2$ | 7 (6.56e0) | 6 (1.78e0) | 10 (3.13e01) | 8 (5.68e1) |
| $10^4$ | 11 (3.17e3) | 6 (2.09e0) | 16 (1.46e04) | 8 (1.17e2) |
| $10^6$ | 17 (3.61e5) | 6 (2.11e0) | 26 (1.53e06) | 8 (1.19e2) |
| $10^8$ | 22 (4.05e7) | 6 (2.11e0) | 41 (7.50e08) | 8 (1.19e2) |

solvers.[4] By letting this framework to be highly coupled with the numerical integration data structures of the application, on the one hand, and to be highly customizable, on the other, one can derive optimal preconditioners for the particular structure of the discrete operator at hand, and tackle new problems and challenges, while leveraging the distributed-memory implementation ideas [3, 4, 6] on which the framework is grounded on. In a nutshell, the global set of MPI tasks is split into those that have fine-grid duties and those that have coarse-grid duties. In addition, the different computations and communications arising in the BDDC-PCG algorithm are scheduled and mapped in such a way that the maximum degree of duty overlapping is achieved. Customizable building blocks in the framework include the fine-grid to coarse-grid DOFs aggregation, the constraint matrix underlying the imposition of continuity of coarse DOFs functionals across coarse objects, the weighting operator underlying the injection among the continuous and discontinuous spaces, and the kind of solvers to be used for the Dirichlet, Neumann constrained local problems, and the coarsest-grid global problem. In particular, the parallel codes in FEMPAR heavily use standard computational kernels provided by highly-efficient vendor implementations of BLAS and LAPACK. Besides, through proper interfaces to several third party libraries, the aforementioned problems can be either solved via sparse direct or approximate (e.g., AMG) solvers. For example, the use of inexact solvers within this framework has been presented in [5].

The experiments in this section have been performed on a large-scale multicore-based massively parallel supercomputer, Gottfried, hosted by the Leibniz Universität IT Services (LUIS) at Hannover (Germany). Gottfried is a Cray XC30 equipped with 744 compute nodes connected together by Cray's proprietary "Aries" high speed interconnect (dragonfly topology). Each node is equipped with 2x Intel Xeon IvyBridge (E5-2695v2) multi-core CPUs, with 12 cores each (i.e., 24 cores per node), and 64 GBytes of DDR3 RAM. Compute nodes run a lightweight kernel and run-time environment based on the SuSE Linux Enterprise Server (SLES) version 11.

FEMPAR was compiled with the Intel Fortran compiler (16.0.3) using system-recommended optimization flags, and linked against the Intel MPI Library (5.1.3) for message-passing, and the BLAS/LAPACK and PARDISO available on the Intel MKL library (11.3.3) for optimized dense linear algebra kernels, and sparse direct solvers, respectively. All floating-point calculations were performed in IEEE double precision.

The performance and scalability of the HPC implementation of PB-BDDC available at FEMPAR will be compared against those of PCBDDC (see [57] and references therein), a highly scalable and robust implementation of BDDC available from PETSc library [10]. PCBDDC combines adaptive constraint selection with the optimal deluxe scaling [57], making it suitable for elliptic PDEs with arbitrary jumps in the coefficients. It belongs to the category of DD preconditioners for heterogeneous PDEs which require auxiliary problems and spectral solvers (see Section 1). This comparison is designed in order to provide experimental evidence on our claim that PB-BDDC can still very robust with the contrast of the coefficients for thermal and mechanical multi-material simulations, while consuming significantly less computational and memory resources. Unassembled local stiffness matrices (and associated data describing distributed-memory layout) are exported from FEMPAR into data files and imported into PCBDDC by an *ad-hoc* designed driver program. This driver program was linked against the PETSc library (3.7.0) compilation available as one of the Cray Programming Environment supported software packages on Gottfried. For a

[4] Indeed, since 2014, the multilevel DD solvers within FEMPAR have been in the High-Q club of the most scalable European codes, maintained by the Jülich supercomputing center [13].

fair comparison among PB-BDDC and PCBDDC, only the aggregated time spent in setting up the BDDC preconditioner and solving the global linear system by PCG will be reported.

5.2.1. *3D Poisson's problem with multiple channels.* In this experiment, we will study a series of problems modeled by (1) in the unit cube $\Omega = [0, 1]^3$. We consider homogeneous Dirichlet boundary conditions on the whole boundary, and the forcing term $f = 1$. First, we build a set of uniform meshes with $4mk \times 3mk \times 2mk$ hexahedra, where $k = 10, 20, 30, 40$, and $m = 1, 2, 3, 4, 5, 6, 7$. The meshes are split into $4m \times 3m \times 2m$ cubic subdomains, with $k \times k \times k$ hexahedra each. In other words, the parameter $m$ decides the partition of the global mesh into subdomains, while $k$ is the size of the local problems ($\frac{H}{h}$) on each subdomain. We use Q1 scalar-valued Lagrangian FEs for the discretization. The largest problem considered is thus discretized using a mesh composed of 526,848,000 cells, and solved on 8,232 subdomains. For all the experiments in Section 5.2, we consider a one-to-one mapping among subdomains, MPI tasks and CPU cores.

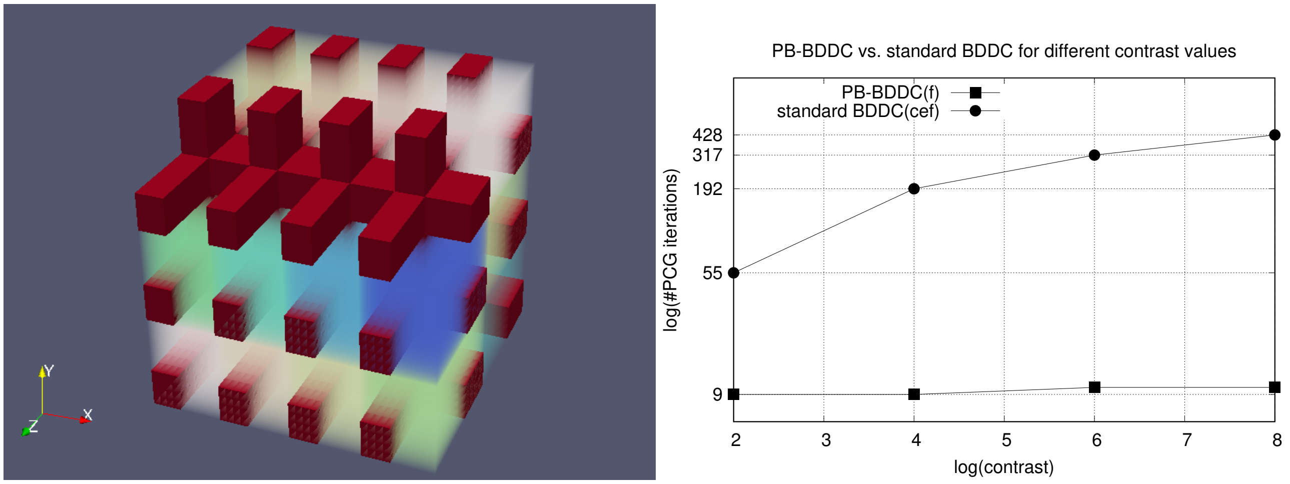
For this experiment, we construct the physical coefficient $\alpha$ in (1) using two different materials, the background with $\alpha = 1.0$ and a set of channels (see Figure 13a) with $\alpha = 10^{\ell}$ (the particular values chosen for $\ell$ will be given when describing each experiment at hand). The number of channels as well as their structure depends on the partition in such a way that we have one channel per direction per subdomain. In addition, the channels are positioned at the lower corner (i.e., at the smallest $x$ and $y$ coordinates of each subdomain). The fact that these channels not only cut through subdomain faces with their smaller cross sections but also touching two other subdomain faces by two of their longer cross sections, makes the problem even more difficult. We also note that the number of channels scales with the number of subdomains and thus the problems get harder and harder, i.e., the number of inclusions scales with the number of subdomains, and the number of additional coarse functions to make the preconditioner robust with respect to the contrast of physical parameters increase in the same way.

In Figure 13b, we compare the number of PCG iterations required by the standard BDDC(cef) and PB-BDDC(f) preconditioners for the smallest problem of the series, i.e., the one obtained with $k = 10$ and $m = 1$, as a function of $\alpha$ within the channels, with $\alpha = 10^{\ell}$ and $\ell = 2, 4, 6, 8$. We set the initial guess $x_0 = 0$, and the PCG iteration is stopped whenever the residual $r_k$ at a given iteration $k$ satisfies $\|r_k\|_2 \leq 10^{-6}\|r_0\|_2$. *This set-up also applies to the rest of experiments in Section 5.2.* From Figure 13b, it can be observed that the PB-BDDC preconditioner is robust with respect to the jump of the diffusion coefficient between materials. It needs only 9 or 10 iterations to converge regardless of the contrast of the coefficient. On the other hand, the number of iterations required by the standard BDDC becomes larger as the contrast increases. When $\ell = 8$, it needs more than 400 iterations to converge for this simple test problem.

In Figures 14, 15 and 16, we respectively report the weak scalability in the computational time, in the number PCG iterations, and in the size of the coarse-grid problem of the three variants of BDDC, namely PCBDDC(c), PB-BDDC(f) and PB-BDDC(f-min), for the case $l = 8$. Here, PCBDDC(c) is the HPC implementation of BDDC in PETSc equipped with deluxe scaling and adaptive selection of constraints started with only corner constraints. In addition, PB-BDDC(f) and PB-BDDC(f-min) correspond to the HPC implementation of the PB-BDDC preconditioner in `FEMPAR`. PB-BDDC(f) uses all PB face coarse objects, while PB-BDDC(f-min) use only the ones associated with high diffusion coefficient, following the mathematical analysis-driven algorithm in section 4.2.

For PCBDDC(c), there are two parameters needed to be tuned. On the one hand, the so-called adaptive threshold [57], for selecting eigenfunctions to enrich the coarse space, and the number of MPI tasks devoted to solve the coarse-grid problem by the parallel distributed-memory sparse direct solver in the MUMPS library. We performed preliminary tests, where the threshold equals 2, 5, and 10 (the smallest and largest values were used in the PCBDDC paper [57]). Based on these preliminary tests, the threshold 5 is selected for all of the scalability tests because PCBDDC(c), since it was the one that provided better CPU times. On the other hand, we selected the number of MPI tasks from the set $2, 4, 8, 12, 24$ that led to less computational time for each number of subdomains. Thus, we note that, in Figure 14, the number of MPI tasks devoted for the coarse-grid problem may increase as we go to a larger number of subdomains.[5]

[5] For reproducibility purposes, the exact command-line argument values that we used to invoke the PETSc PCBDDC driver program were [57]: `-pc_bddc_use_vertices true -pc_bddc_use_edges false -pc_bddc_use_faces`



(A) High diffusivity channels are shown in solid color for the partition $4 \times 3 \times 2$. Subdomains are shown as transparent volume with different colors. The subdomains in the top front are clipped to reveal the inner structure of the channels. The number of channels is always proportional to the number of subdomains.

(B) PCG iterations vs. $\log(\alpha_{\max})$.

Figure 13. 3D Poisson's problem

The HPC implementation of the PB-BDDC preconditioner follows the ideas described in detail in [4]. A separated MPI task is devoted to the coarse-grid problem, and mapped to a separated compute node, where it can exploit up to 24 cores and 64 GBytes of RAM memory. The multi-threaded PARDISO solver in Intel MKL was used to solve the coarse-grid problem within this node. The number of threads was set to 24 in all experiments, as this number leads to the lowest computational time.

In Figures 14, 15, and 16, the results for $\frac{H}{h} = 40$ are not reported for PCBDDC(c). For such local problem size, PCBDDC(c) did not fit into memory *for all number of subdomains considered.* A memory profile of PCBDDC(c) revealed that there are two main sources of extra memory consumption over PB-BDDC. First, the PETSc implementation of PCBDDC is such that it does not exploit the symmetry of the problem at hand, i.e., both the lower and upper triangles of matrices and their respective LU factors are stored. Second, adaptive selection of constraints and deluxe scaling need to explicitly compute the Schur complement matrix related to subdomain interface DOFs, and to set up auxiliary eigenvalue problems. The second source was confirmed as PCBDDC(c) for $H/h = 40$ without deluxe scaling and adaptive selection of constraints did fit into memory.

As observed in Figure 16, the coarse-grid problem size for PB-BDDC(f) and PB-BDDC(f-min) does not depend on the value of $\frac{H}{h}$, but solely on the partition of mesh cells into PB subdomains. On the other hand, some of the coarse DoFs of PCBDDC(c) are judiciously selected based on numerical eigenvalue analysis, and this may depend on the spectrum of the operators at hand. This in particular justifies why in Figure 16 the coarse-grid problem size depends on $\frac{H}{h}$.

It can be concluded by the results of Figure 15 that the three methods are robust and scalable, as they all lead to an asymptotically constant number of iterations for fixed problem size, and increasing number of subdomains, regardless of the high contrast of the $\alpha$ coefficient, and ill-distribution of the channels. PB-BDDC(f-min) and PCBDDC(c) require a very similar number of iterations, while PB-BDDC(f) slightly need less at the price of a much larger coarse-grid problem size, see Figure 16. For all three methods, a slight increase in the number of iterations is observed as $\frac{H}{h}$ becomes larger. This is a well-known behavior mathematically justified by the analysis of these methods.

However, significant differences in the performance and scalability in computational time of the three methods are observed in Figure 14. For simplicity, let us first focus on the top left subfigure with the

```
false -pc_bddc_adaptive_threshold {2,5,10} -pc_bddc_use_deluxe_scaling true -pc_bddc_coarse_redistribute
{2,4,8,12,24} -pc_bddc_coarse_pc_type lu -pc_bddc_coarse_pc_factor_mat_solver_package mumps.
```

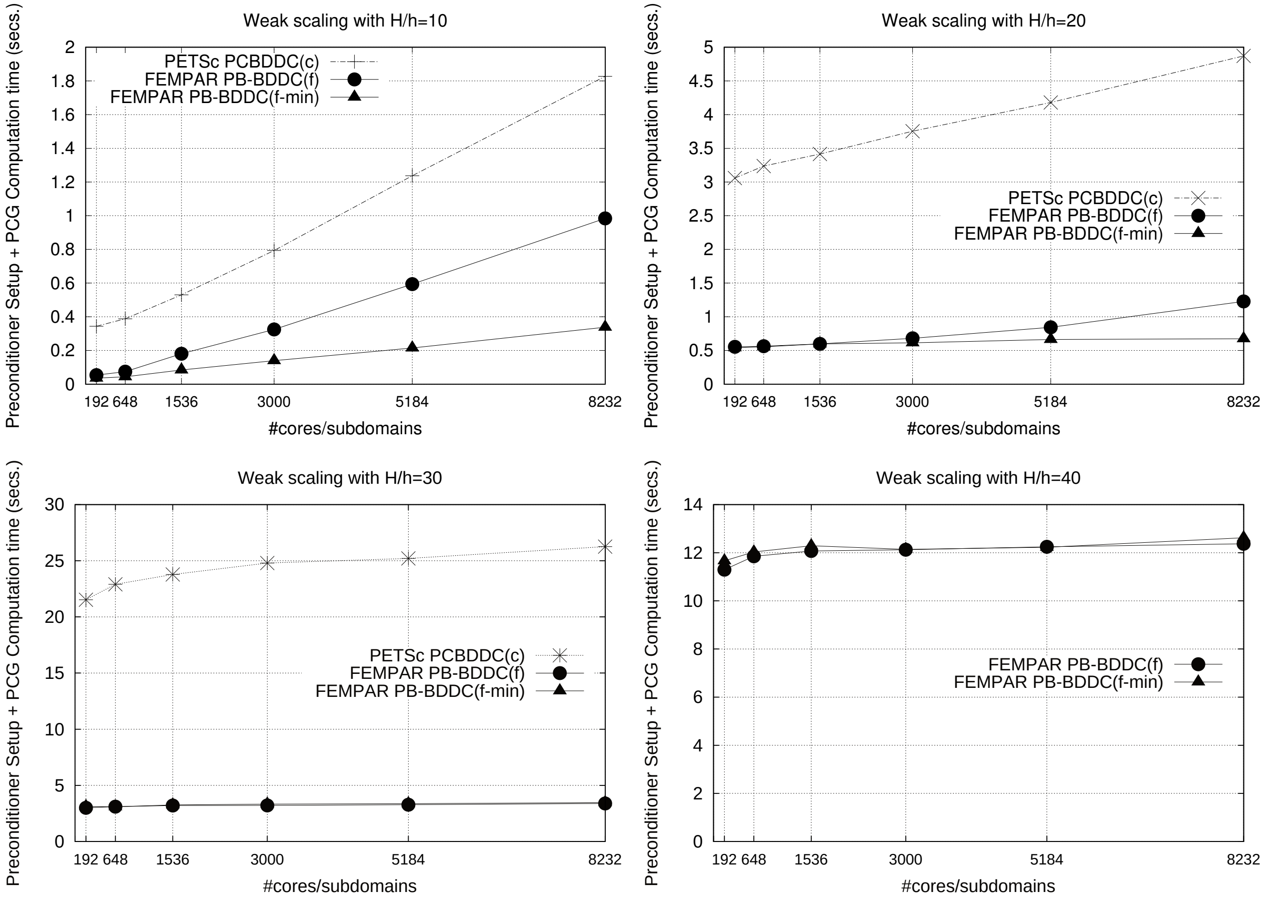


FIGURE 14. Weak scalability for the total computational time of the PCBDDC(c), PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Poisson problem on Gottfried.

smallest local problem size $\frac{H}{h} = 10$. For such local problem size, the computational time increases with fixed problem size and increasing number of subdomains. The reason is that the coarse-grid problem time-to-solution grows with the number of subdomains and dominates the total computation time. Their computational times grow at a different pace, though. In particular, it can be observed that, despite the coarse-grid problem size of PB-BDDC(f-min) and PCBDDC(c) being almost the same (see Figure 16), the HPC implementation of the former turns out to be much more effective in tackling the overhead associated with the coarse-grid problem. This can be explained by the positive effect that overlapping of fine-grid and coarse-grid duties at all stages of the BDDC implementation has on its parallel scalability [4]. In any case, *we can observe that PB-BDDC is faster than PCBDDC*. This is justified by the extra computational time that PCBDDC requires for deluxe scaling and adaptive selection of constraints, i.e., explicit computation of Schur complement associated to interface DoFs, set up and solution of auxiliary eigenvalue problems. In addition, the larger $\frac{H}{h}$ becomes, the larger the gap between the two algorithms becomes. For example, for $\frac{H}{h} = 30$, PB-BDDC(*) is roughly 7.8x faster than PCBDDC when larger number of subdomains is considered, and even larger factors are foreseen for larger $\frac{H}{h}$. Another interesting observation is that the three methods become more and more effective in tackling the bottleneck associated to the solution of the coarse-grid problem as $\frac{H}{h}$ increases, with PB-BDDC being more effective than PCBDDC. Indeed, for $\frac{H}{h} = 30, 40$, no difference is observed for the computational time of PB-BDDC(f) and PB-BDDC(f-min), and the weak scalability of both is excellent, despite the much larger coarse-grid problem size of the latter compared to the former.

5.2.2. *3D linear elasticity problem with multiple channels.* In this experiment, we consider the linear elasticity problem (2) in the unit cube $\Omega = [0,1]^3$. We let the forcing term $f = [0\ 0\ 0]^T$, and impose

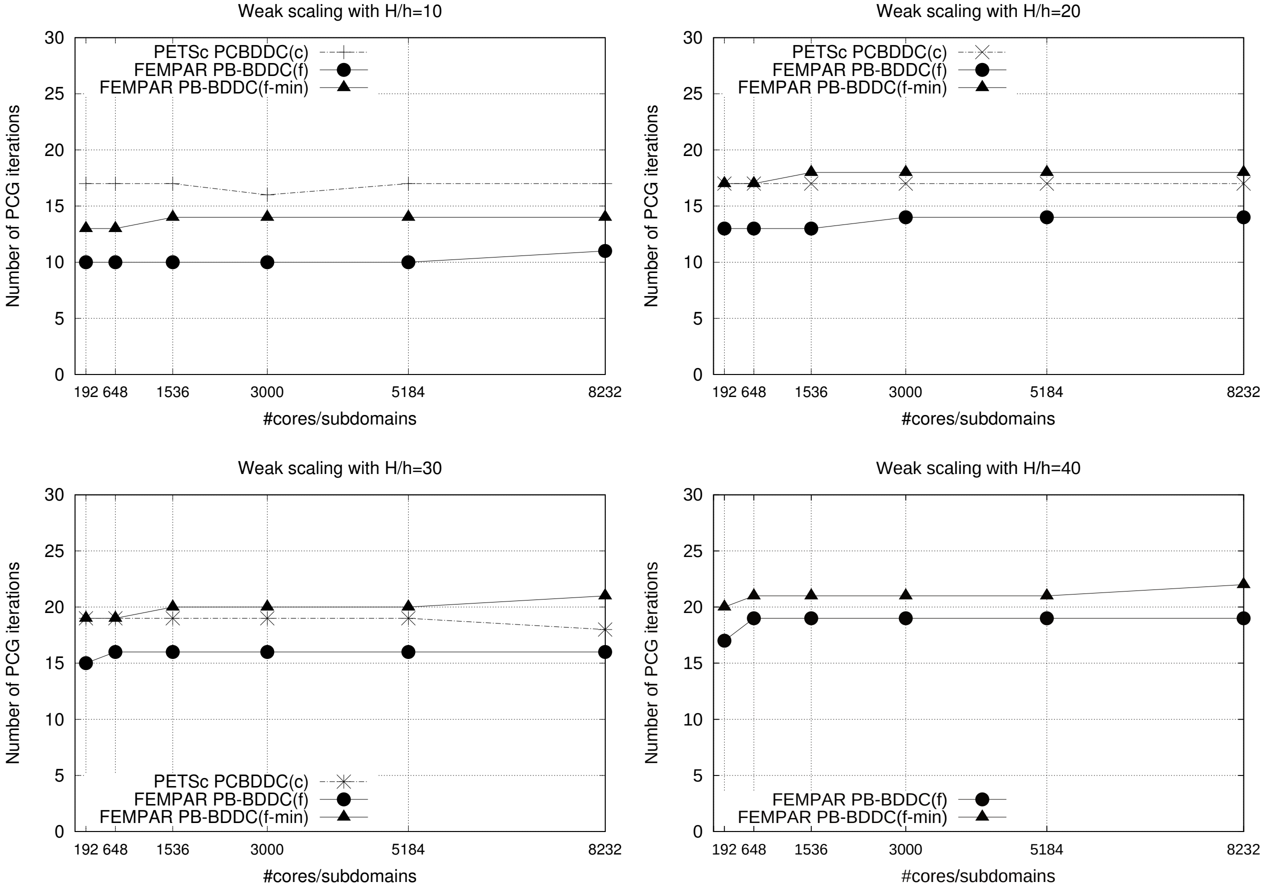


Figure 15. Weak scalability for the number of PCG iterations of the PCBDDC(c), PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Poisson problem.

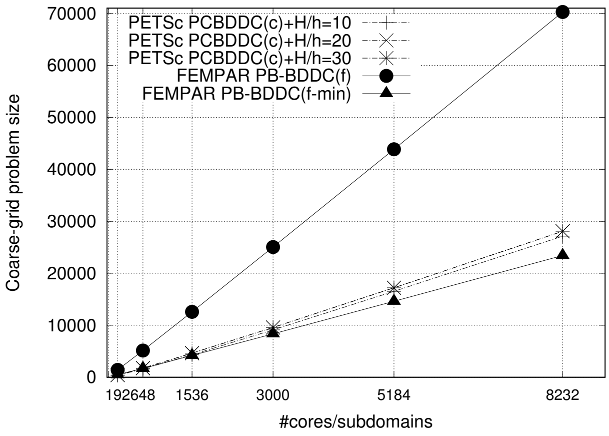


Figure 16. Weak scalability for the coarse-grid problem size of the PCBDDC(c) with different local problem size $\frac{H}{h}$, PB-BDDC(f), and PB-BDDC(f-min) for the 3D Poisson's problem.

inhomogeneous Dirichlet boundary conditions $u(x) = x_1 + x_2 + x_3$ on the whole boundary. Similar to the 3D Poisson's problem, we also use meshes with $4mk \times 3mk \times 2mk$ hexahedra, where $m = 2, 3, 4, 5, 6, 7$, but with $k = 10, 15, 20, 30$. Regular partitions are used to divide this set of meshes into $4m \times 3m \times 2m$ subdomains. Then we consider a material with Poisson's ratio $\nu = 0.2$ and two different values for the

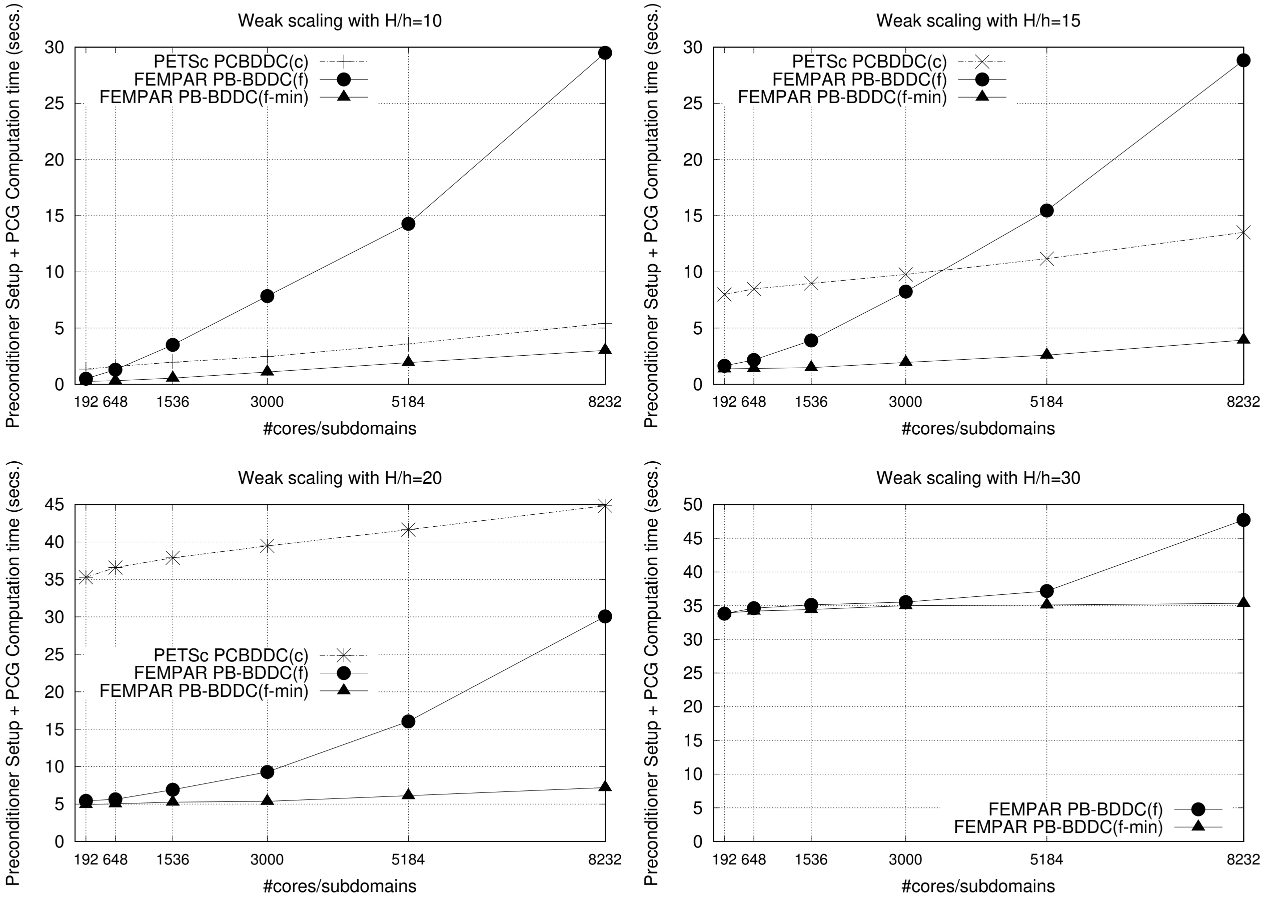


FIGURE 17. Weak scalability for the total computational time of the PCBDDC(c), PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Linear Elasticity problem on Gottfried.

Young's modulus: the background with $E = 1$ and a system of channels with $E = 10^{\ell}$, where $\ell = 2, 4, 6, 8$. The configuration of these channels is exactly as described in section 5.2.1 for the 3D Poisson's problem. Q1 vector-valued Lagrangian FEs are used for the discretization of this problem.

In Figure 17, we respectively report the weak scalability in computational time, in the number of PCG iterations and in the coarse grid problem size of the PCBDDC(c), PB-BDDC(f) and PB-BDDC(f-min). The Young modulus $E = 10^6$ was used in this test case for the coefficient within the channels. The results of PCBDDC(c) for $\frac{H}{h} = 30$ are not reported as this test did not fit into available memory.

Overall, similar conclusions to the ones in Section 5.2.1 can be drawn from the results in Figures 17, 18, and 19. For the linear elasticity problem, though, the balance achieved among the factors determining performance and scalability of the solvers at hand was different. The linear elasticity problem is a vector-valued problem with 3 unknowns per vertex. This implies that, for a given FE mesh, the size of the discrete operator is 3 times larger to that of the Poisson problem, and has 9 times more nonzero entries. Besides, the PB-BDDC(*) solvers require 6 coarse DoFs per PB coarse face, corresponding to the six rigid body modes. Thus, a different balance is necessarily struck among the time spent in local solvers and the coarse-grid problem.

We would like to remark the following observations. Firstly, between two variants of PB-BDDC, PB-BDDC(f-min) is the only one that achieves perfect weak scalability in time for this test case. Indeed, the HPC implementation of PB-BDDC(f) is not able to effectively tackle the bottleneck associated to the coarse-grid problem in the whole number of subdomains range, even for the most favorable scenario when $\frac{H}{h} = 30$. For this local problem size, time weak scalability starts degrading beyond 3K subdomains for PB-BDDC(f), see Figure 17. Even though PB-BDDC(f) is faster than PCBDDC(c) in this case, the

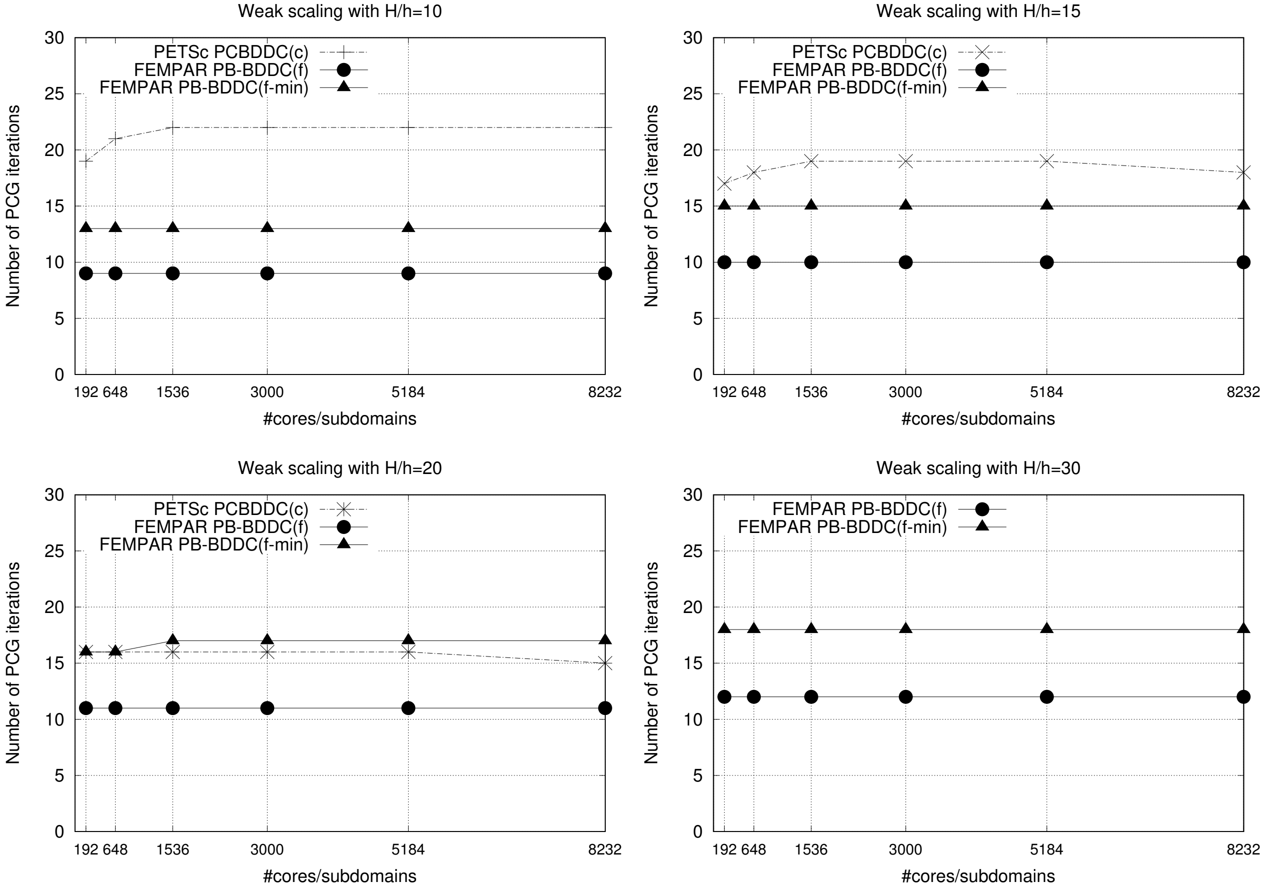


FIGURE 18. Weak scalability for number of PCG iterations of the PCBDDC(c), PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Linear Elasticity problem on Gottfried.

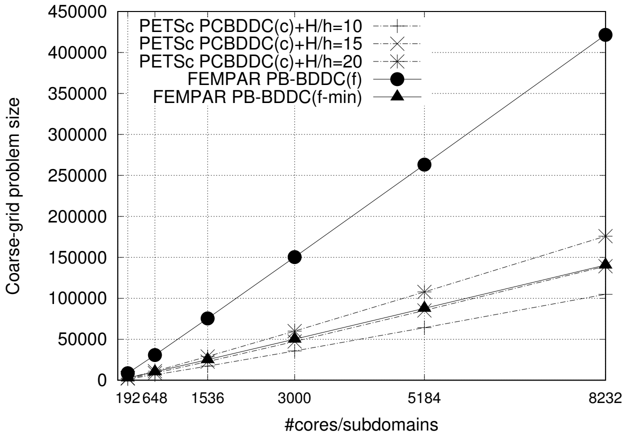
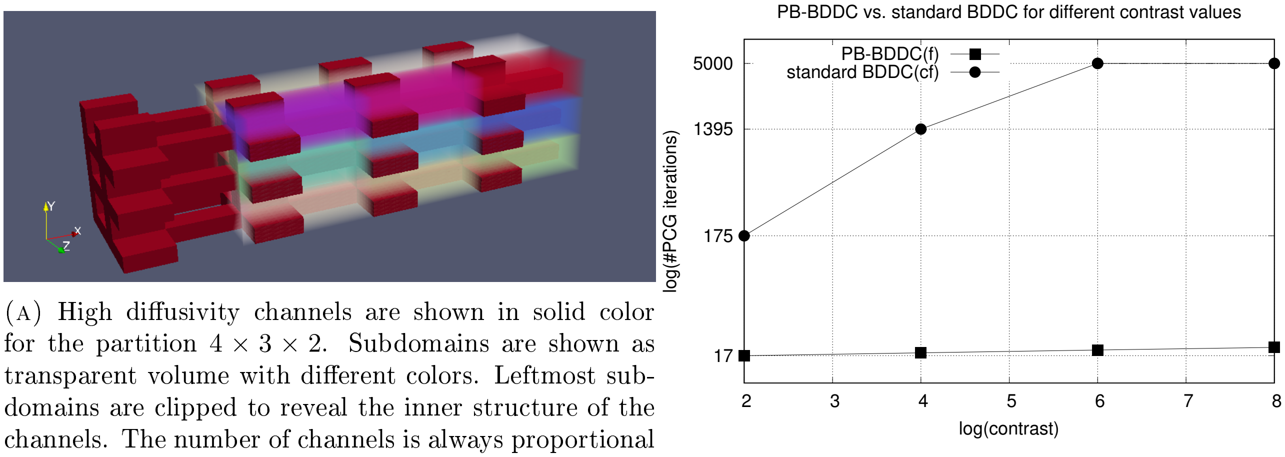


FIGURE 19. Weak scalability for the coarse-grid problem size of the PCBDDC(c), PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Linear Elasticity problem.

trends indicate that PCBDDC(c) will be faster for larger number of subdomains. Secondly, PB-BDDC(f-min) requires more iterations than PB-BDDC(f), but still less or very close to the ones of PCBDDC(c), depending on the local problem size. This extra number of iterations can be compensated by the benefit associated to the reduction of the coarse-grid problem size. As a result, PB-BDDC(f-min) is faster than



(A) High diffusivity channels are shown in solid color for the partition 4 × 3 × 2. Subdomains are shown as transparent volume with different colors. Leftmost subdomains are clipped to reveal the inner structure of the channels. The number of channels is always proportional to the number of subdomains.

(B) PCG iterations vs. $\log(\alpha_{\max})$.

FIGURE 20. 3D Linear Elasticity of a beam.

PCBDDC(c), most noticeably for the local problem size $\frac{H}{h} = 20$. For example, for $\frac{H}{h} = 20$, and 3K subdomains, PB-BDDC(f-min) is 7.4x faster than PCBDDC(c).

5.2.3. *3D linear elasticity for a beam with multiple channels.* In this experiment, we consider problem (2), where $\Omega$ is the long beam $[0,2] \times [0,0.5] \times [0,0.5]$. We use homogeneous Dirichlet boundary conditions on the plane $x = 0$, and homogeneous Neumann boundary conditions on the rest of the boundary. The force is $f = [0\ \text{-}9.8\ 0]^T$. The rest of experiment set up was equivalent to that described in Section 5.2.2. In Figure 20a, we illustrate the structure of the channels (i.e. the ones where a value of $\alpha = 10^l$ is used) for the partition 4 × 3 × 2. In Figure 20b, we compare the number of PCG iterations required by the standard BDDC(cf) and PB-BDDC(f) preconditioners for the smallest problem of the series, i.e., the one obtained with $k = 10$ and $m = 1$, as a function of the Young's modulus $E$ within the channels, $E = 10^{\ell}$, and $\ell = 2, 4, 6, 8$. From Figure 20b, it can be observed that the PB-BDDC preconditioner is robust with respect to the jump of the diffusion between materials. It needs only 17 or 18 iterations to converge regardless of the contrast of the coefficient. On the other hand, the number of iterations required by the standard BDDC becomes larger as the contrast increases. For $l = 6, 8$, standard BDDC does not converge for this small-scale test problem even after 5000 iterations.

In Figures 21 and 18, we respectively report the weak scalability in the computational time and in the number of PCG iterations of PB-BDDC(f) and PB-BDDC(f-min) for the Linear Elasticity problem of a beam with multiple channels. The sizes of the coarse-grid problem for PB-BDDC(f) and PB-BDDC(f-min) for this experiment are the same as the ones reported in Section 5.2.2, see Figure 19. The Young's modulus $E = 10^6$ was used as the coefficient within the channels. The results for PCBDDC(c) are not reported for this problem. Our implementation of PB-BDDC in FEMPAR can handle natural boundary condition on any part of the boundary without losing robustness or weak scalability. On the other hand, PCBDDC breaks down if there is at least one subdomain associated only with natural boundary conditions. In order for PCBDDC to work in this case, an automatic selection of primal corners that guarantees the well posedness of local Neumann problems is needed, such as the one described in [50], but not implemented yet. In any case, the results are expected to be analogous to the ones in the previous section.

Overall, very similar conclusions to the ones in Section 5.2.2 can be drawn from the results in Figures 21 and 22. Perhaps the most noticeably difference between the two experiments is that PB-BDDC(f-min) roughly requires twice the number of iterations compared to that of PB-BDDC(f) for this experiment, see Figure 22. However, small coarse-grid problem sizes appear to compensate for large number of iteration. Indeed, PB-BDDC(f-min) is faster than PB-BDDC(f) either for small local problem sizes, or for large number of subdomains, see Figure 21.

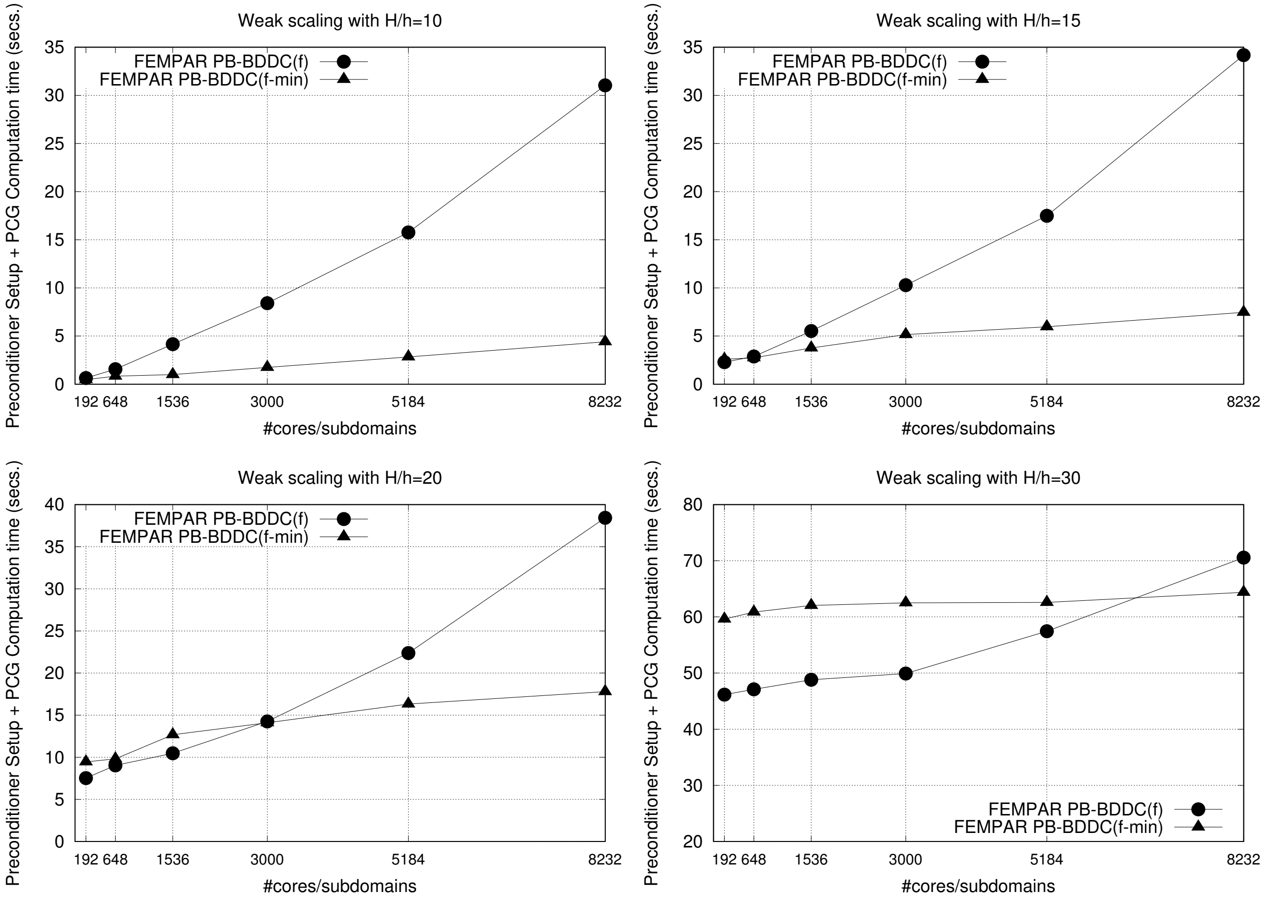


Figure 21. Weak scalability for the total computational time of the PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Linear Elasticity of a beam with multiple channels on Gottfried.

## 6. Conclusions

In this work, we have proposed a novel type of BDDC preconditioners that are robust for multi-material problems with high contrast. The underlying idea is to modify the continuity constraints enforced among subdomains making use of the knowledge about the physical coefficients. In order to do that, we rely on a physically motivated partition of standard coarse objects (corners, edges, and faces) into coarse subobjects. The motivation for that is the well-known robustness of DD methods when there are only jumps of physical coefficients across the interface between subdomains. All these ideas can also be used in the frame of FETI methods.

In cases where the physical coefficient is constant in each PB subdomain, we are able to prove that the associated condition number can be bounded independently of the number of the subdomains and the contrast of the physical coefficient. In other words, the new preconditioner is scalable and robust for multi-material problems. In addition, not all but any set of PB objects satisfying a mild condition on acceptable paths is sufficient to guarantee robustness.

Apart from the new set of coarse objects and a weighting operator, the PB-BDDC preconditioner is very much the same as the standard BDDC preconditioner. As a result, the implementation of the new preconditioners involves a very simple modification of the standard BDDC implementation. In all of our experiments, the new preconditioners delivers fast, robust and contrast-independent convergence while maintaining the simplicity of BDDC methods at a reasonable computational cost. Compared to the other robust DD solvers for heterogeneous problems currently available, such as the ones in [28, 48, 28, 46, 47, 25, 26, 45, 51, 18, 53, 52, 31, 30, 32, 39, 27], our approach does not involve any type of eigenvalue or auxiliary problem.

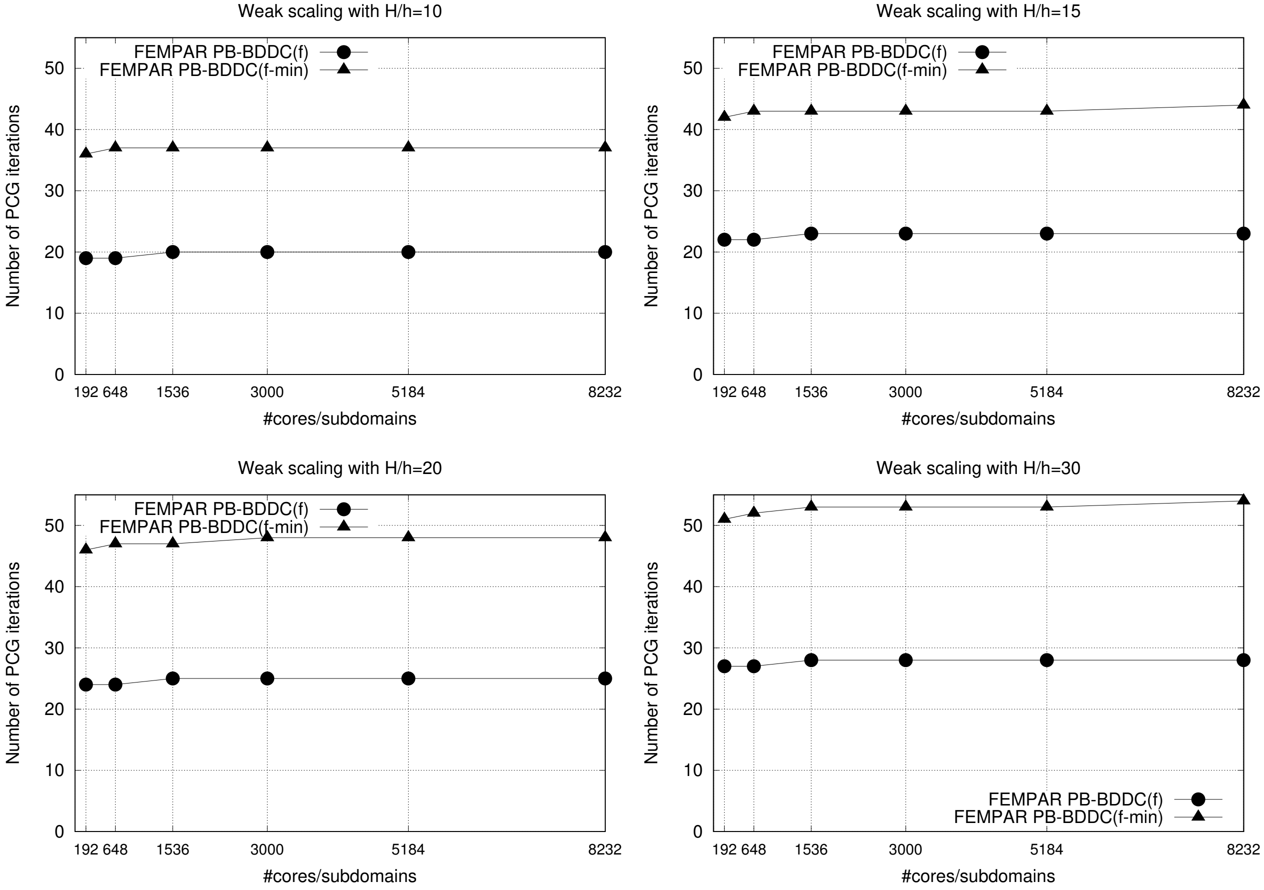


FIGURE 22. Weak scalability for number of PCG iterations of the PB-BDDC(f), and PB-BDDC(f-min) solvers for the 3D Linear Elasticity problem of a beam with multiple channels.

The use of the PB-BDDC method to highly complex multi-material problems can lead to larger coarse problems than in the homogeneous case (as the spectral-based methods listed above). In order to cope this problem, we have considered two remedies. First, we provide a simple procedure to find small and close to minimal coarse spaces that guarantee robustness. Second, we have implemented the algorithms in the extremely scalable BDDC code in FEMPAR [3, 4, 5, 6], which exploits an inter-level task-overlapping implementation to efficiently deal with the coarse space computation in weak scalability analyses. We have shown excellent scalability results up to 8232 cores and more than 500 million unknowns on a supercomputer. The preconditioner has been compared with a state-of-the-art implementation of BDDC with deluxe scaling in PETSc, called PCBDDC. PB-BDDC has been observed to be cheaper and less expensive in terms of memory than PCBDDC for the multi-material problems considered in the numerical experiments. It justifies the motivation of this work, i.e., to define robust and simple BDDC methods for multi-material problems that do not require expensive spectral solvers to define the coarse spaces or weighting operators, but to define these ingredients only looking at the physical properties of the materials at hand.

## References

[1] S. Badia, A. Martín, and J. Principe, *FEMPAR: An object-oriented parallel finite element framework*, Archives of Computational Methods in Engineering, (2017).

[2] S. Badia, A. F. Martín, and J. Principe, *Enhanced balancing Neumann-Neumann preconditioning in computational fluid and solid mechanics*, International Journal for Numerical Methods in Engineering, 96 (2013), pp. 203–230.

[3] ——, *Implementation and scalability analysis of balancing domain decomposition methods*, Archives of Computational Methods in Engineering, 20 (2013), pp. 239–262.

[4] ——, *A highly scalable parallel implementation of balancing domain decomposition by constraints*, SIAM Journal on Scientific Computing, 36 (2014), pp. C190–C218.

[5] ——, *On the scalability of inexact balancing domain decomposition by constraints with overlapped coarse/fine corrections*, Parallel Computing, 50 (2015), pp. 1 – 24.

[6] ——, *Multilevel balancing domain decomposition at extreme scales*, SIAM J. Sci. Comput., 38 (2016), pp. C22–C52.

[7] ——, *FEMPAR Web page.* `http://www.fempar.org`, 2017.

[8] S. Badia and H. Nguyen, *Relaxing the roles of corners in BDDC by perturbed formulation*, 2016.

[9] ——, *Balancing domain decomposition by constraints and perturbation*, SIAM Journal on Numerical Analysis, (accepted).

[10] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.

[11] S. C. Brenner and L. R. Scott, *The mathematical theory of finite element methods*, vol. 15 of Texts in Applied Mathematics, Springer, New York, third ed., 2008.

[12] S. C. Brenner and L.-Y. Sung, *BDDC and FETI-DP without matrices or vectors*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 1429–1435.

[13] D. Brömmel, B. J. N. Wylie, and W. Frings, *JUQUEEN Extreme Scaling Workshop 2015*, Tech. Rep. FZJ-2015-01645, Jülich Supercomputing Center, 2015.

[14] T. F. Chan and T. P. Mathew, *Domain decomposition algorithms*, Acta numerica, 3 (1994), pp. 61–143.

[15] C. R. Dohrmann, *A Preconditioner for Substructuring Based on Constrained Energy Minimization*, SIAM Journal on Scientific Computing, 25 (2003), pp. 246–258.

[16] ——, *An approximate BDDC preconditioner*, Numerical Linear Algebra with Applications, 14 (2007), pp. 149–168.

[17] C. R. Dohrmann and O. B. Widlund, *A BDDC algorithm with deluxe scaling for three-dimensional H(*curl*) problems*, Communications on Pure and Applied Mathematics, 69 (2016), pp. 745–770.

[18] V. Dolean, F. Nataf, R. Scheichl, and N. Spillane, *Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps*, Comput. Methods Appl. Math., 12 (2012), pp. 391–414.

[19] M. Dryja, J. Galvis, and M. Sarkis, *BDDC methods for discontinuous Galerkin discretization of elliptic problems*, J. Complexity, 23 (2007), pp. 715–739.

[20] M. Dryja, M. V. Sarkis, and O. B. Widlund, *Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions*, Numer. Math., 72 (1996), pp. 313–348.

[21] M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal., 31 (1994), pp. 1662–1694.

[22] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen, *FETI-DP: a dual-primal unified FETI method-part I: A faster alternative to the two-level FETI method*, International Journal for Numerical Methods in Engineering, 50 (2001), pp. 1523–1544.

[23] C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numerical Linear Algebra with Applications, 7 (2000), pp. 687–714.

[24] C. Farhat, N. Maman, and G. W. Brown, *Mesh partitioning for implicit computations via iterative domain decomposition: Impact and optimization of the subdomain aspect ratio*, International Journal for Numerical Methods in Engineering, 38 (1995), pp. 989–1000.

[25] J. Galvis and Y. Efendiev, *Domain decomposition preconditioners for multiscale flows in high-contrast media*, Multiscale Model. Simul., 8 (2010), pp. 1461–1483.

[26] ——, *Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces*, Multiscale Model. Simul., 8 (2010), pp. 1621–1644.

[27] M. J. Gander, A. Loneland, and T. Rahman, *Analysis of a new harmonically enriched multiscale coarse space for domain decomposition methods*, Dec. 2015.

[28] I. G. Graham, P. O. Lechner, and R. Scheichl, *Domain decomposition for multiscale PDEs*, Numer. Math., 106 (2007), pp. 589–626.

[29] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1998), pp. 359–392 (electronic).

[30] H. H. Kim and E. T. Chung, *A BDDC algorithm with enriched coarse spaces for two-dimensional elliptic problems with oscillatory and high contrast coefficients*, Multiscale Model. Simul., 13 (2015), pp. 571–593.

[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] A. Klawonn, P. Radtke, and O. Rheinbach, *Adaptive coarse spaces for BDDC with a transformation of basis*, in Domain Decomposition Methods in Science and Engineering XXII, T. Dickopf, J. M. Gander, L. Halpern, R. Krause, and F. L. Pavarino, eds., Springer International Publishing, 2016, pp. 301–309.

[33] A. Klawonn, O. Rheinbach, and O. B. Widlund, *An analysis of a FETI–DP algorithm on irregular subdomains in the plane*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2484–2504.

[34] A. Klawonn and O. B. Widlund, *Dual-primal FETI methods for linear elasticity*, Communications on Pure and Applied Mathematics, 59 (2006), pp. 1523–1572.

[35] A. Klawonn, O. B. Widlund, and M. Dryja, *Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients*, SIAM J. Numer. Anal., 40 (2002), pp. 159–179 (electronic).

[36] J. H. Lee, *A balancing domain decomposition by constraints deluxe method for Reissner-Mindlin plates with Falk-Tu elements*, SIAM J. Numer. Anal., 53 (2015), pp. 63–81.

[37] J. Li and O. B. Widlund, *FETI-DP, BDDC, and block Cholesky methods*, International Journal for Numerical Methods in Engineering, 66 (2006), pp. 250–271.

[38] ——, *On the use of inexact subdomain solvers for BDDC algorithms*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 1415–1428.

[39] S. Loisel, H. Nguyen, and R. Scheichl, *Optimized schwarz and 2-lagrange multiplier methods for multiscale elliptic pdes*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2896–A2923.

[40] J. Mandel, *Balancing domain decomposition*, Communications in Numerical Methods in Engineering, 9 (1993), pp. 233–241.

[41] J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numerical Linear Algebra with Applications, 10 (2003), pp. 639–659.

[42] J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Applied Numerical Mathematics, 54 (2005), pp. 167–193.

[43] J. Mandel, B. Sousedík, and C. Dohrmann, *Multispace and multilevel BDDC*, Computing, 83 (2008), pp. 55–85.

[44] T. P. A. Mathew, *Domain decomposition methods for the numerical solution of partial differential equations*, vol. 61, Springer, 2008.

[45] F. Nataf, H. Xiang, V. Dolean, and N. Spillane, *A coarse space construction based on local Dirichlet-to-Neumann maps*, SIAM J. Sci. Comput., 33 (2011), pp. 1623–1642.

[46] C. Pechstein and R. Scheichl, *Analysis of FETI methods for multiscale PDEs*, Numer. Math., 111 (2008), pp. 293–333.

[47] ——, *Analysis of FETI methods for multiscale PDEs. Part II: interface variation*, Numer. Math., 118 (2011), pp. 485–529.

[48] R. Scheichl and E. Vainikko, *Additive Schwarz with aggregation-based coarsening for elliptic problems with highly variable coefficients*, Computing, 80 (2007), pp. 319–343.

[49] J. Šístek, M. Čertíková, P. Burda, and J. Novotný, *Face-based selection of corners in 3D substructuring*, Math. Comput. Simulation, 82 (2012), pp. 1799–1811.

[50] ——, *Face-based selection of corners in 3D substructuring*, Math. Comput. Simulation, 82 (2012), pp. 1799–1811.

[51] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl, *A robust two-level domain decomposition preconditioner for systems of PDEs*, C. R. Math. Acad. Sci. Paris, 349 (2011), pp. 1255–1259.

[52] 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*, Numer. Math., 126 (2014), pp. 741–770.

[53] N. Spillane, V. Dolean, P. Hauret, F. Nataf, and D. J. Rixen, *Solving generalized eigenvalue problems on the interfaces to build a robust two-level FETI method*, C. R. Math. Acad. Sci. Paris, 351 (2013), pp. 197–201.

[54] A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.

[55] X. Tu, *Three-Level BDDC in Three Dimensions*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1759–1780.

[56] O. B. Widlund and C. R. Dohrmann, *BDDC deluxe domain decomposition*, in Domain Decomposition Methods in Science and Engineering XXII, T. Dickopf, J. M. Gander, L. Halpern, R. Krause, and F. L. Pavarino, eds., Cham, 2016, Springer International Publishing, pp. 93–103.

[57] S. Zampini, *PCBDDC: A Class of Robust Dual-Primal Methods in PETSc*, SIAM Journal on Scientific Computing, 38 (2016), pp. S282–S306.

[58] Y. Zhu, *Domain decomposition preconditioners for elliptic equations with jump coefficients*, Numer. Linear Algebra Appl., 15 (2008), pp. 271–289.

Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034, Barcelona, Spain.
*E-mail address*: sbadia@cimne.upc.edu,amartin@cimne.upc.edu

CIMNE – Centre Internacional de Mètodes Numèrics en Enginyeria, Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain.
*E-mail address*: amartin@cimne.upc.edu,sbadia@cimne.upc.edu

Institute of Research and Development, Duy Tan, University, 3 Quang Trung, Danang, Vietnam.
*E-mail address*: nguyentrunghieu14@dtu.edu.vn