PHYSICS-BASED BALANCING DOMAIN DECOMPOSITION BY
CONSTRAINTS FOR HETEROGENEOUS PROBLEMS

SANTIAGO BADIA†‡ AND HIEU NGUYEN‡

Abstract. In this work, we present a balancing domain decomposition by constraints method
based on an aggregation of elements depending on the physical coefficients. Instead of imposing
constraints on purely geometrical objects (faces, edges, and vertices) of the partition interface, we use
interface objects (subfaces, subedges, and vertices) determined by the variation of the coefficients.
The new method is easy to implement and does not require to solve any eigenvalue or auxiliary
problem. When the physical coefficient in each object is constant at every subdomain containing the
object, we can show both theoretically and numerically that the condition number does not depend
on the contrast of the coefficient. The constant coefficient condition is possible for multi-material
problems. However, for heterogeneous problems with coefficient varying across a wide spectrum of
values in a small spatial scale, such restriction might result in too many objects (a large coarse
problem). In this case, we propose a relaxed version of the method where we only require that the
maximal contrast of the physical coefficient in each object is smaller than a predefined threshold.
The threshold can be chosen so that the condition number is reasonably small while the size of the
coarse problem is not too large. An extensive set of numerical experiments is provided to support
our findings.

Key words. BDDC, heterogeneous problem, adaptive coarse space, parallel solver, parallel
preconditioner

AMS subject classifications. 65N55, 65N22, 65F08

1. Introduction. Many realistic simulations in science and engineering, such
as subsurface flow simulations in a nuclear waste repository or in an oil reservoir, or
heat conduction in composites, involve heterogeneous 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 with the variation of the coefficients of the PDEs. For a different but related
approach, to find reasonably accurate heterogeneous solution on a coarse mesh, we
refer the interested readers to [19, 1] and references therein.

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 [50, 41]. In DD, the coarse space plays an
important role in achieving scalability as well as robustness w.r.t variations in the
coefficient. Many early DD methods, such as those in [11, 18, 17, 31, 53], work
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 the properties of the media might have complicated variations

† Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
‡ CIMNE – Centre Internacional de Mètodes Numèrics en Enginyeria, Parc Mediterrani de la
Tecnologia, UPC, Estève Terradas 5, 08860 Castelldefels, Spain ({sbadia,hnguyen}@cimne.upc.edu).
* This work has 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. S. Badia gratefully acknowledges the
support received from the Catalun Government through the ICREA Acadèmia Research Program.
on many scales and be difficult to capture by a geometric coarse grid. Further, it is impractical, since it would not lead to load-balanced partitions with a reduced interface.

Recently, there have been works on coarse grids that do not resolve the heterogeneity in the media [25, 45, 25, 43, 44], and especially automatic coarse spaces that adapt to the variation in the properties of the media [22, 23, 42, 47, 15, 49, 48, 28, 27, 29, 36, 24]. In the latter, the coarse spaces are constructed from eigenfunctions associated with small eigenvalues (low-frequency modes) of appropriated 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 [33, 14, 52]. The approach yields robust DD methods, but extra implementation and computation cost incur due to the auxiliary problems. In this paper, we formulate a new balancing DD by constraints (BDDC) preconditioner that requires no eigenvalue or auxiliary problem and is very robust with the contrast of the coefficient. 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 [12]. It is an improved version of the balancing DD (BDD) method by Mandel [38] and has a close connection with the FETI-DP method [21, 20]. In fact, it can be shown that the eigenvalues of the preconditioned operators associated with BDDC and FETI-DP are almost identical [39, 34, 10]. 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 [13, 35] by, e.g., one AMG cycle, and it can straightforwardly be extended to multiple levels [51, 40]. 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.

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 [31, 30, 50]. This implies that in order to have robustness for BDDC methods one could use a physics-based partition obtained by aggregating elements of the same coefficient value. However, using this type of partition is impractical as the number of the 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 METIS [26] an automatic graph partitioner, to distribute the work load among processors. Then, we consider a sub-partition of subdomains based on the physical coefficients, leading to a physics-based (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 will be called PB-BDDC. These ideas can readily be applied to FETI-DP preconditioners.

We emphasise that the PB-BDDC method does not require to solve any eigenvalue or auxiliary problems. Its formulation and implementation are very much the same.
For simplicity of exposition, we assume that the problem at hand is: find $u^* \in H^1_\partial(\Omega)$ such that $-\alpha \Delta u^* = f$ in $H^{-1}(\Omega)$ sense, with $f \in H^{-1}(\Omega)$ and $\alpha \in L^\infty(\Omega)$ strictly positive. The weak form of the problem reads

$$\int_\Omega \alpha \nabla u^* \cdot \nabla v \, dx = \int_\Omega f v \, dx, \quad \text{for any } v \in H^1_\partial(\Omega).$$

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 $\tau \in \mathcal{T}$.

### 2. Problem setting.

Let $\Omega \subset \mathbb{R}^d$, with $d$ being the space dimension, be a bounded polyhedral domain. For a model problem, we study the Poisson’s equation with non-constant diffusion and homogeneous Dirichlet conditions (the non-homogeneous case only involves an obvious modification of the right-hand side). Thus, the problem at hand is: find $u \in H^1(\Omega)$ such that $-\alpha \Delta u = f$ in $H^{-1}(\Omega)$ sense, with $f \in H^{-1}(\Omega)$ and $\alpha \in L^\infty(\Omega)$ such that

$$\int_\Omega \alpha \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx, \quad \text{for any } v \in H^1(\Omega).$$

### 2.1. Domain partitions.

We first consider a partition $\Theta$ of $\Omega$ 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 work load 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., $\Gamma(\Theta) = (\bigcup_{\mathcal{D} \in \Theta} \partial \mathcal{D}) \setminus \partial \Omega$.

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 $\Theta_{pb}(\mathcal{D})$ into a set of “sub-subdomains” with constant $\alpha$. The minimal set is preferred for efficiency (it will potentially lead to a smaller coarse space) but it is not a requirement (see Remark 5). Clearly, the resulting global PB partitions $\Theta_{pb} \equiv \{\Theta_{pb}(\mathcal{D})\}_{\mathcal{D} \in \Theta}$
is also a partition of $\Omega$ (into PB subdomains). The interface of this partition is $\Gamma(\Theta_{pb}) = (\bigcup_{D \in \Theta_{pb}} \partial D \setminus \partial \Omega$. For a subdomain $D \in \Theta$ (analogously for $\hat{D} \in \Theta_{pb}$), we denote by $T_D$ the submesh of $T$ associated with $\hat{D}$, $T_D = \{ \tau \in T : \tau \subset D \} \subset T$. For any $\hat{D} \in \Theta_{pb}$, let $\omega(\hat{D})$ be the only subdomain in $\Theta$ that contains $\hat{D}$. In Figure 1, we show an example of the original partition $\Theta$ and the PB partition $\Theta_{pb}$ for a simple problem. The meaning of $\Theta_{pb}(D)$ and $\omega(\hat{D})$ is also illustrated.

2.2. Finite element spaces. Let us perform a discretization of (1) by a continuous finite element (FE) space $\overline{V}$ associated with the mesh $T$. The discontinuous Galerkin (DG) case will not be considered in this work, but we refer the reader to [16] for more information.

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

For a subdomain $D \in \Theta$, we also define the subdomain FE operator $A_D : V_D \rightarrow V'_D$ as $\langle A_D u, v \rangle = \int_D \alpha \nabla u \cdot \nabla v \, dx$, for all $u, v \in V_D$, and the sub-assembled operator $A^\Theta : \overline{V} \rightarrow \overline{V}'$ as $\langle A^\Theta u, v \rangle = \sum_{D \in \Theta} \langle A_D u, v \rangle$, for all $u, v \in \overline{V}$.

A function $u \in V_D$ is said to be discrete $\alpha$-harmonic in $D$ if

$$\langle A_D u, v \rangle = 0, \quad \text{for any } v \in V_{0,D},$$

where $V_{0,D} = \{ v \in V_D : v = 0 \text{ on } \partial D \}$. It should be noted that if $u$ is discrete $\alpha$-harmonic in $D$ then it satisfies the energy minimizing property, namely

$$\langle A_D u, u \rangle \leq \langle A_D v, v \rangle, \quad \forall v \in V_D, \quad v|_{\partial D} = u|_{\partial D}.$$

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

\[
(2) \quad \langle Au, v \rangle = \langle f, v \rangle, \quad \text{for any } v \in \overline{V}.
\]

Fig. 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 \( \text{neigh}_{\Theta_{\text{pb}}} \). 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 \( \Lambda_{\text{pb}}(\Theta) \) 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 \( \text{neigh}_{\Theta}(\lambda) \) 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 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 \( T \) based on their subdomain set.

Denote by \( \text{ndof}(\lambda) \) 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 \( |\text{neigh}_{\Theta}(\lambda)| = 2 \) and \( \text{ndof}(\lambda) > 1 \), is an edge if \( |\text{neigh}_{\Theta}(\lambda)| > 2 \) and \( \text{ndof}(\lambda) > 1 \), and is a corner if \( \text{ndof}(\lambda) = 1 \). In the two-dimensional case, \( \lambda \in \Lambda(\Theta) \) is an edge if \( |\text{neigh}_{\Theta}(\lambda)| = 2 \) and \( \text{ndof}(\lambda) > 1 \), and is a corner if \( \text{ndof}(\lambda) = 1 \). In the literature, e.g., [31, 50], corners are also referred to as vertices. Analogous definitions are also used frequently for FETI-DP methods (see [50]). 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 \( \Lambda_{\text{pb}}(\Theta) \) obtained by applying the previous classification of \( \Gamma(\Theta) \) into corners/edges/faces but with \( \text{neigh}_{\Theta}(\cdot) \) replaced by \( \text{neigh}_{\Theta_{\text{pb}}}(\cdot) \). In other words, we use sets of subdomains in
\[ \Theta_{pb}, \text{to classify geometrical objects on } \Gamma(\Theta). \text{ Figure 2 (right) shows objects in } \Lambda_{pb}(\Theta) \text{ for a simple example.} \]

**Lemma 2.** \( \Lambda_{pb}(\Theta) \) is a refinement of \( \Lambda(\Theta) \).

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

**Remark 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., [46]). A new approach based on perturbations has recently been proposed in [8, 7], where the method is well-posed in all cases.

**Remark 4.** The PB aggregation (classification) of the interface \( \Gamma(\Theta) \) into \( \Lambda_{pb}(\Theta) \) can be relaxed. As it is currently stated, the PB partition is unique and have the minimal number of PB subdomains. However, it might introduces disconnected objects. For example, the edge between \( \hat{\Omega}_3 \) and \( \hat{\Omega}_7 \) in Figure 2 (right) is disconnected. Alternatively, one can require that objects must be connected. This leads to two connected edges between \( \hat{\Omega}_3 \) and \( \hat{\Omega}_7 \). We adopt this practice for the numerical experiments in section 4. However, it should be noted that the use of disconnected objects leads to a smaller coarse space and can be beneficial in some cases.

**Remark 5.** In practical implementations, one only needs the set of PB geometrical objects \( \Lambda_{pb}(\Theta) \) to define the PB-BDDC preconditioner. When using the approach with only connected objects (see Remark 4), one does not need to explicitly define the PB partition \( \Theta_{pb} \). Only a partition of objects in \( \Lambda(\Theta) \) (see Figure 2 (left)) into PB objects (see Figure 2 (right)) based on the physical coefficients is required. Therefore, only a \((d - 1)\)-dimensional PB partition (of the interface \( \Gamma(\Theta) \)) is needed. This is what we have actually implemented for our numerical experiments in section 4. We only use the PB partition \( \Theta_{pb} \) in the analysis in subsection 3.4. In any case, the PB partition can easily be implemented if necessary.

### 3. Physics-based BDDC preconditioning

In this section, we present our new PB-BDDC method. 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 \( \Lambda_{pb}(\Theta) \) are associated with a coarse DOF. We denote this set of objects by \( \Lambda_O \) and call it the set of coarse objects. Obviously, \( \Lambda_O \subset \Lambda_{pb}(\Theta) \). Typical choices of \( \Lambda_O \) are \( \Lambda_O = \Lambda_C \), when only corners are considered, \( \Lambda_O = \Lambda_C \cup \Lambda_E \), when corners and edges are considered, or \( \Lambda_O = \Lambda_{pb}(\Theta) \), 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.
Given an object $\lambda \in \Lambda_O$, we define its coarse DOF as the mean value on $\lambda$. The rigorous definition is as follows. Assume $\lambda \in \Lambda_O$ is associated with a subdomain $D \in \Theta$. We define the coarse DOF $c_\lambda^D$ corresponding to $\lambda$ as

$$
c_\lambda^D(u_D) = \frac{\int_D u_D \, ds}{\lambda}, \quad \text{for } u_D \in \mathbb{V}_D.
$$

Clearly, $c_\lambda^D$ is a functional in $\mathbb{V}_D$. When $\lambda$ is a corner, $c_\lambda^D$ is simply the value at that corner. Once we have defined the coarse DOFs, we can define the BDDC space as follows

$$
\mathbb{V} \triangleq \{ v \in \mathbb{V} : c_\lambda^D(v) = c_\lambda^{D'}(v), \forall \lambda \in \Lambda_O, \forall D, D' \in \text{neigh}_\Theta(\lambda) \},
$$
i.e., the subspace of functions in $\mathbb{V}$ that are continuous “at” coarse DOFs. Clearly, $\mathbb{V} \subset \mathbb{V}_1 \subset \mathbb{V}$.

For BDDC methods, solving the coarse problem is usually the bottleneck (cf. [2, 3, 4, 8]). 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 [31, 50], in the case where the physical coefficient in each subdomain is constant, the set of coarse objects only need to guarantee the existence of the so-called acceptable paths. We need a similar concept here for the PB-BDDC method.

**Definition 6 (Acceptable path).** Let $\Theta_{pb}^D$ be the set of PB subdomains $\hat{D} \in \Theta_{pb}$ touching the interface $\Gamma(\Theta)$, i.e., $\partial \hat{D} \cap \Gamma(\Theta) \neq \emptyset$. For two subdomains $\hat{D}_a, \hat{D}_b \in \Theta_{pb}^D$ that share an edge $\lambda$ but no face in $\Lambda_{pb}(\Theta)$ or share a corner $\lambda$ but no edge in $\Lambda_{pb}(\Theta)$, an acceptable path is a sequence $\{ \hat{D}_a = \hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n = \hat{D}_b \}$ of PB subdomains in $\Theta_{pb}$, which satisfy the following properties:

i) they all share the common object $\lambda \in \Lambda_{pb}(\Theta)$

ii) subdomains $\hat{D}_k$ and $\hat{D}_{k+1}$, $k = 1, \ldots, n-1$, must share, apart from $\lambda$, an object in $\Lambda_O$ and the type of the shared object (face, edge or corner) must be the same for the whole sequence

iii) their (constant) coefficients satisfy

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

where TOL is some predefined tolerance and $R(k, \lambda) = 1$ if $\lambda$ is an edge and $R(k, \lambda) = h(\hat{D}_k)/H(\hat{D}_k)$ if $\lambda$ is a corner.

**Assumption 7.** We assume that the set of BDDC objects $\Lambda_O$ satisfies the following properties:

1. In the three dimensional case, for each face on $\Gamma(\Theta)$, there is at least one edge that is part of its boundary and belongs to $\Lambda_O$.

2. For all pairs of subdomains $\hat{D}_a, \hat{D}_b \in \Theta_{pb}^D$, which have an edge but not a face in $\Lambda_{pb}(\Theta)$ in common, or a corner but not an edge in $\Lambda_{pb}(\Theta)$ in common, there exists an acceptable path for a predefined tolerance TOL.

**Remark 8.** In **Definition 6**, if the shared object $\lambda$ belongs to the set of BDDC objects $\Lambda_O$, then there exists a trivial acceptable path $\{ \hat{D}_a, \hat{D}_b \}$ with TOL = 1 and $n = 2$. Thus, BDDC(ce) and BDDC(cef) always satisfy **Assumption 7** for TOL = 1.
3.2. Injection operators. Let us define the projection $Q : V \to \bar{V}$ as some weighted average of interface values together with an $\alpha$-harmonic extension to subdomain interiors (see, e.g., [39]). We define these ingredients as follows.

For $u \in V$ and $\xi \in \Gamma(\Theta)$, the weighting operator is defined as

$$Wu(\xi) = \sum_{D \in \text{neigh}_b(\xi)} \delta^I_D(\xi) u_D(\xi), \quad \text{with} \quad \delta^I_D(\xi) = \frac{\sum_{\tau \in T_D, \tau \ni \xi} \alpha_T |\tau|}{\sum_{\tau \in T, \tau \ni \xi} \alpha_T |\tau|},$$

where $|\tau|$ denotes the volume (area if in 2D) of the element $\tau$.

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

$$\mathcal{E}u = (1 - A_0^{-1}A)u,$$

where $A_0$ is the Galerkin projection of $A$ onto the bubble space $V_0 = \{ v \in V : v = 0 \text{ on } \Gamma(\Theta) \}$.

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

3.3. PB-BDDC preconditioner. 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 $\Lambda_0$, 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 (5) is new.

The BDDC preconditioner is a Schwarz-type preconditioner that combines interior corrections with corrections in the BDDC space (see, e.g., [9, 50]). In case of the PB-BDDC preconditioner, the BDDC correction is expressed as $Q(\tilde{A}^\Theta)^{-1}Q^T$, where $\tilde{A}^\Theta$ is the Galerkin projection of $A^\Theta$ onto $\bar{V}$. More specifically, the PB-BDDC preconditioner reads as follows:

$$B = A_0^{-1} + Q(\tilde{A}^\Theta)^{-1}Q^T.$$

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 [12, 13, 40, 9] 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 $\bar{B}$, an auxiliary BDDC preconditioner. The definition of this preconditioner follows verbatim that of the PB-BDDC preconditioner above but the PB subdomain partition $\Theta_{pb}$ is used instead of $\Theta$.

Given the FE mesh $T$, the FE space type, and the subdomain partition $\Theta_{pb}$, one can similarly build the FE spaces and operators as in subsection 2.2, leading to the sub-assembled space $V^\text{pb}$ and operator $A^\text{pb}$. Further, we can define the injection operator $\tilde{Q}$ using the definitions in subsection 3.2 with $\Theta$ is replaced by $\Theta_{pb}$ for the weighting $\bar{W}$ and harmonic extension $\bar{E}$ operators.

**Lemma 9.** For any PB subdomain $\bar{D} \in \Theta_{pb}$, the function $\delta^I_{\bar{D}}(\cdot)$ is constant on each PB object $\lambda$ associated with it, i.e.,

$$\delta^I_{\bar{D}}(\xi) = \delta^I_{\bar{D}}(\xi'), \quad \forall \xi, \xi' \in \lambda.$$
In addition, the following important inequality, cf. [50, 6.19], holds

\[ (7) \quad \alpha_{\hat{D}_a} \left( \frac{\delta_{\hat{D}_b}(\xi)}{\delta_{\hat{D}_h}(\xi)} \right)^2 \leq C_{\delta1} \min \left( \alpha_{\hat{D}_a}, \alpha_{\hat{D}_h} \right), \quad \forall \hat{D}_a, \hat{D}_h \in \text{neigh}_{\Theta_{pb}}(\xi), \]

where \( C_{\delta1} \leq \max \left( 1, \frac{N_{\max} - 1}{4} \right) \) with \( N_{\max} \) is the maximal number of elements in \( \mathcal{T} \) sharing at least one point.

**Proof.** The identity (6) follows from (5), the fact that \( \alpha \) is constant in each subdomain in \( \Theta_{pb} \) and \( \text{neigh}_{\Theta_{pb}}(\xi) = \text{neigh}_{\Theta_{pb}}(\xi') = \text{neigh}_{\Theta_{pb}}(\lambda) \).

Now we need to verify (7). Since \( \alpha \) is constant in each PB subdomain, we can rewrite and bound \( \delta_{\hat{D}_b}^1(\xi) \) as follows

\[ \delta_{\hat{D}_b}^1(\xi) = \frac{\alpha_{\hat{D}_b} A_{\hat{D}_b}}{\sum_{\hat{D} \in \text{neigh}_{\Theta_{pb}}(\xi)} \alpha_{\hat{D}} A_{\hat{D}}} < \frac{\alpha_{\hat{D}_a} A_{\hat{D}_a}}{\alpha_{\hat{D}_a} A_{\hat{D}_a} + \alpha_{\hat{D}_h} A_{\hat{D}_h}}, \]

where \( A_{\hat{D}} > 0 \) denotes the volume (area) of the patch of elements in \( \hat{D} \) containing \( \xi \).

Clearly, \( \delta_{\hat{D}_b}^1(\xi) \leq 1 \). Therefore, \( \alpha_{\hat{D}_a} (\delta_{\hat{D}_b}^1(\xi))^2 \leq \alpha_{\hat{D}_a} \). Now we need to prove that

\[ (9) \quad \alpha_{\hat{D}_a} \left( \delta_{\hat{D}_b}^1(\xi) \right)^2 \leq C_{\delta1} \alpha_{\hat{D}_a}. \]

Using (8), it is sufficient to show

\[ (10) \quad \alpha_{\hat{D}_a} \alpha_{\hat{D}_h} A_{\hat{D}_b}^2 \leq C_{\delta1} \left( \alpha_{\hat{D}_a} A_{\hat{D}_a} + \alpha_{\hat{D}_h} A_{\hat{D}_h} \right)^2. \]

Since the mesh \( \mathcal{T} \) is quasi-uniform, elements sharing at least a point have roughly the same volume (area). Consequently, \( A_{\hat{D}_b} \lesssim (N_{\max} - 1) A_{\hat{D}_a} \) (the worst case scenario is when \( \hat{D}_b \) has \( N_{\max} - 1 \) elements and \( \hat{D}_a \) has 1 element). Using this, we have

\[ \alpha_{\hat{D}_a} \alpha_{\hat{D}_h} A_{\hat{D}_b}^2 \lesssim (N_{\max} - 1) \alpha_{\hat{D}_a} \alpha_{\hat{D}_h} A_{\hat{D}_a} A_{\hat{D}_h} \leq \frac{N_{\max} - 1}{4} \left( \alpha_{\hat{D}_a} A_{\hat{D}_a} + \alpha_{\hat{D}_h} A_{\hat{D}_h} \right)^2. \]

This implies (10) and we finish the proof.

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

**Lemma 10.** All the objects in \( \Lambda_{pb}(\Theta) \) are also in \( \Lambda(\Theta_{pb}) \), i.e., \( \Lambda_{pb}(\Theta) \subset \Lambda(\Theta_{pb}) \).

**Proof.** Let us consider an object \( \lambda_{pb} \in \Lambda_{pb}(\Theta) \). In both object partitions \( \Lambda_{pb}(\Theta) \) and \( \Lambda(\Theta_{pb}) \), we are using the same criteria, i.e., \( \text{neigh}_{\Theta_{pb}}(\cdot) \), to classify points. The difference is that \( \Lambda_{pb}(\Theta) \) is the result of a classification of points in \( \Gamma(\Theta) \) whereas \( \Lambda(\Theta_{pb}) \) is obtained from a classification of points in \( \Gamma(\Theta_{pb}) \). Since \( \Gamma(\Theta) \subset \Gamma(\Theta_{pb}) \), all points in \( \lambda_{pb} \) belong to the same object \( \lambda' \in \Lambda(\Theta_{pb}) \). Since \( \lambda_{pb} \) is on the interface \( \Gamma(\Theta) \), there exist at least two subdomains \( \hat{D}, \hat{D'} \in \text{neigh}_{\Theta_{pb}}(\lambda_{pb}) \) such that \( \omega(\hat{D}) \neq \omega(\hat{D'}) \). Let us assume there is a point \( \xi \in \lambda' \) such that \( \xi \notin \lambda_{pb} \). Then, \( \xi \in \Gamma(\Theta_{pb}) \cap \Gamma(\Theta) \), i.e., it only belongs to one subdomain in \( \Theta \). As a result, \( \omega(\hat{D}) \) is the same for all \( \hat{D} \in \text{neigh}_{\Theta_{pb}}(\xi) \). Thus, we have a contradiction, since \( \text{neigh}_{\Theta_{pb}}(\xi) \) cannot be the same as \( \text{neigh}_{\Theta_{pb}}(\lambda_{pb}) \).
With the theoretical support from Lemma 10, we can define the set of coarse objects \( \hat{\Lambda}_O \) of \( \hat{B} \) as a classification of \( \Gamma(\Theta_{pb}) \) 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 \( \Lambda_{pb}(\Theta) \). For the rest of the interface \( \Gamma(\Theta_{pb}) \setminus \Gamma(\Theta) \), we enforce full continuity among PB subdomains. It can be understood as treating all FE nodes on \( \Gamma(\Theta_{pb}) \setminus \Gamma(\Theta) \) as corners. Denote this set of objects by \( \hat{\Lambda}^* \), we have \( \hat{\Lambda}_O = \Lambda_O \cup \hat{\Lambda}^* \).

Remark 11. By construction, the BDDC space \( \tilde{V}^{pb} \) of the auxiliary BDDC preconditioner \( \hat{B} \) is identical to the BDDC space \( \tilde{V} \), defined in (4), of the PB-BDDC preconditioner.

**Lemma 12.** The condition number \( \kappa(BA) \) of the PB-BDDC preconditioned operator is bounded by

\[
\kappa(BA) \leq \max_{v \in \tilde{V}^{pb}} \frac{\langle A^{\Theta_{pb}} \hat{Q}v, \hat{Q}v \rangle}{\langle A^{\Theta_{pb}} v, v \rangle}.
\]

Proof. According to [39, Theorem 15], \( \kappa(BA) \) is bounded by

\[
\kappa(BA) \leq \max_{v \in \tilde{V}} \frac{\langle A^{\Theta} Qv, Qv \rangle}{\langle A^{\Theta} v, v \rangle}.
\]

Now we only need to bound the right-hand-side in (12) by the one in (11).

On the one hand, using the fact that \( \tilde{V} = \tilde{V}^{pb} \), we have \( \langle A^{\Theta} v, v \rangle = \langle A^{\Theta_{pb}} v, v \rangle \) for all \( v \in \tilde{V} \) because any \( v \in \tilde{V} \) is continuous in each subdomain of \( \Theta \). On the other hand, let us prove that the weighting operator \( \hat{W} \) defined by (5) for \( \Theta_{pb} \) restricted to \( \tilde{V} \) is identical to the weighting operator \( W \) defined by (5) for \( \Theta \). Let us consider a subdomain \( D \in \Theta \) and its PB partition \( \Theta_{pb}(D) \). We have

\[
\delta^T_D(\xi) = \sum_{D \in \Theta_{pb}(D), D \ni \xi} \delta^T_D(\xi),
\]
by the definition in (5). For an arbitrary function \( v \in V \subset \mathbb{V}^{pb} \), we find that
\[
Wv(\xi) = \sum_{D \in \text{neigh}_h(\xi)} \delta_D^v(\xi) v_D(\xi) = \sum_{D \in \text{neigh}_h(\xi)} \sum_{D \in \Theta_{pb}, \partial D \ni \xi} \delta_D^v(\xi) v_D(\xi)
= \sum_{D \in \text{neigh}^{pb}_h(\xi)} \delta_D^v(\xi) v_D(\xi) = \hat{W}v(\xi).
\]

Therefore, \( \hat{Q}v \) and \( Qv \) are identical on \( \Gamma(\Theta) \) and \( \hat{Q}v \) is continuous across \( \Gamma(\Theta_{pb}) \). In addition, \( Qv \) is discrete \( \alpha \)-harmonic in each \( D \in \Theta \) and have minimal energy norm w.r.t \( A^{\Theta} \). As a consequence,
\[
\langle A^{\Theta} Qv, Qv \rangle = \sum_{D \in \Theta} \langle A_D^{\Theta} Qv, Qv \rangle \leq \sum_{D \in \Theta} \langle A_D^{\Theta_{pb}} Qv, Qv \rangle = \sum_{D \in \Theta_{pb}} \langle A_D^{\Theta_{pb}} Qv, Qv \rangle = \langle A^{\Theta_{pb}} Qv, Qv \rangle.
\]

This finishes the proof. \( \square \)

We could stop here and derive the estimate for \( \kappa(BA) \) knowing that the condition number of the auxiliary BDDC preconditioned operator \( \hat{B}A \) is estimated by an upper bound of the last quantity on the right of (12). However, we will go a bit further to obtain a stronger result.

**Lemma 13.** Assume that \( \Lambda_O \) is such that Assumption 7 holds. Then we have the following inequality:
\[
\max_{v \in \mathbb{V}_{pb}} \frac{\langle A^{\Theta_{pb}} \hat{Q}v, \hat{Q}v \rangle}{\langle A^{\Theta_{pb}} v, v \rangle} \leq C \max_{D \in \Theta_{pb}} \left( 1 + \log \left( \frac{H(D)}{h(D)} \right) \right)^2,
\]
where the constant \( C \) is independent of the number of subdomains, \( H(D), h(D) \) and the physical coefficient \( \alpha \).

**Proof.** By triangle inequality, we have
\[
\max_{v \in \mathbb{V}_{pb}} \frac{\langle A^{\Theta_{pb}} \hat{Q}v, \hat{Q}v \rangle}{\langle A^{\Theta_{pb}} v, v \rangle} \leq 1 + \max_{v \in \mathbb{V}_{pb}} \frac{\langle A^{\Theta_{pb}} (\hat{Q}v - v), (\hat{Q}v - v) \rangle}{\langle A^{\Theta_{pb}} v, v \rangle}.
\]

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

By the construction of the set of object \( \hat{\Lambda}_O = \Lambda_O \cup \hat{\Lambda}^\ast \) and the definition of \( \mathbb{V}_{pb}, w \) vanishes at all coarse objects in \( \hat{\Lambda}^\ast \), i.e, at all FE nodes in \( \Gamma(\Theta_{pb}) \setminus \Gamma(\Theta) \). Consequently, (15) can be simplified as follows:
\[
w = \sum_{\lambda \in \Lambda_{pb}(\Theta)} \sum_{D \in \Theta_{pb}^\lambda} \theta_D^\lambda(w).
\]

This manuscript is for review purposes only.
When $\Lambda$ satisfies Assumption 7, the set of objects in $\hat{\Lambda}$ also fulfils [50, Assumption 6.27]. Consequently, using Lemma 9, we can perform an analysis similar to that in the proof [50, Lemma 6.36] (see also [31, Lemma 10]) to obtain

$$
\langle A_{\hat{D}}^{\Theta \theta} \theta_{\hat{D}}^{\hat{D}}(w), \theta_{\hat{D}}^{\hat{D}}(w) \rangle
\leq C \max \{1, \text{TOL} \} \left( 1 + \log \left( \frac{H(\hat{D})}{h(\hat{D})} \right) \right)^2 \sum_{\hat{D} \in \text{neigh}_{\theta \theta}(\lambda)} \langle A_{\hat{D}}^{\Theta \theta} v, v \rangle
$$

for any $\hat{D} \in \Theta_{\theta \theta}$ and $\lambda \in \Lambda(\Theta_{\theta \theta})$. Here the constant $C$ is proportional to $C_{\delta\delta}$ in Lemma 9, but is otherwise independent of $H(\hat{D})$, $h(\hat{D})$ and the physical coefficient $\alpha$.

Adding up the estimate for all subdomain $\hat{D} \in \Theta_{\theta \theta}$, we find that

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

This finishes the proof.

Combining results in Lemma 12 and Lemma 13, we have the final bound for the PB-BDDC preconditioner, which is both weakly scalable and independent of the coefficient $\alpha$.

**Theorem 14.** The condition number of the PB-BDDC preconditioned operator $\kappa(BA)$ is bounded by

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

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

**Remark 15.** As seen in the Lemma 13 and Theorem 14, 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 the PB-BDDC is independent of variations of the coefficient. The main target of this work is achieved.

### 3.5. Relaxed physics-based BDDC

The definition of the coarse objects for the PB-BDDC preconditioner, based on the requirement that the coefficient has to be constant in each PB subdomain, can result in a large coarse space. That is the case for heterogeneous problems where the physical coefficient varies across a wide spectrum of values in a small spatial scale.

In order to deal with a more general class of problems, we propose the relaxed PB-BDDC preconditioner ($r$PB-BDDC) where we only require that the maximal contrast in each PB subdomain is less than some predefined tolerance $r$. We consider a relaxed PB partition $\Theta_{\theta \theta}$ such that

$$
\max_{\tau, \tau' \subset \hat{D}} \alpha_{\tau, \tau'} \leq r, \quad \text{for any } \hat{D} \in \Theta_{\theta \theta}.
$$

Here the threshold $r$ is equal or greater than 1. This way, we can control the size of the coarse problem and the condition number bounds with the choice of $r$. 

This manuscript is for review purposes only.
As the coefficient is no longer constant in each PB subdomain, we need to use a weighted-constraint in the definition of coarse DOFs. More specifically, instead of using (3), we use

\[ c^P_\lambda (u) = \int_{\lambda} \tilde{\alpha} u \frac{d\lambda}{\int_{\lambda} \tilde{\alpha} 1 d\lambda}, \]

where \( \tilde{\alpha}(\xi) = \max_{\tau \in \mathcal{F}, \tau \ni \xi} \alpha_\tau. \)

**Remark** 16. The larger \( r \) becomes the smaller the size of the coarse problem of the rPB-BDDC preconditioner is and the larger its condition number grows to. When \( r = 1 \) the rPB-BDDC preconditioner becomes the PB-BDDC preconditioner. By tuning the threshold \( r \), one can obtain a right balance between the time spent on setting up the preconditioner (especially in forming the coarse space) and the time spent on applying the preconditioner in a Krylov solver. The optimal threshold is of course problem dependent. However, finding a good threshold is not tricky. This is illustrated in section 4.

**Remark** 17. The rPB-BDDC preconditioner makes use of a threshold. This is similar to the adaptive coarse space approach where only eigenfunctions associated with eigenvalues below a predefined threshold are included in the coarse space. However, the rPB-BDDC preconditioner does not involve any eigenvalue or auxiliary problems and is far simpler and cheaper.

**4. Numerical experiments.** In this section, we test the robustness and efficiency of the PB-BDDC and rPB-BDDC preconditioners for the system matrix associated with (2) for different types of variation in the coefficient \( \alpha \), which are similar but generally harder than the ones in [42, 28, 36].

Due to the difficulty of heterogeneous problems, in PB-BDDC and rPB-BDDC methods, we tend to use a large number of objects. Many of them are not corners. In all tested cases, these objects are enough to make the local Neumann problems and global coarse problem well-posed and we can optionally drop corner objects. No corner detection mechanism (see, e.g., [46]) has been needed in any tested case. Alternatively, one might want to consider the perturbed formulation introduced in [7, 8]. However, this approach has not been extended to heterogeneous problems yet.

In all of the experiments, we consider the physical domain \( \Omega = (0,1)^2 \). Unless stated otherwise, we use the uniform triangular meshes of size \( h = 1/72 \) and the regular \( 3 \times 3 \) subdomain partition. In all cases, we report the dimension of the coarse space, denoted by \( \text{dim} \), and the number of iterations required for the conjugate gradient method to reduce the residual norm by a factor of \( 10^6 \). We also provide the condition number \( \kappa \) in most examples.

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

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 \( \alpha_{\text{max}} \), whereas they remain practically constant for both variants of the PB-BDDC preconditioners (PB-BDDC(ce) and PB-BDDC(e)). 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. Figure 4 shows the coarse objects of PB-BDDC(ce) on the interface of the partition.

This manuscript is for review purposes only.
Table 1
Comparison of the iteration count and condition number in the two channels test case.

<table>
<thead>
<tr>
<th>dim →</th>
<th>BDDC(ce)</th>
<th>PB-BDDC(ce)</th>
<th>PB-BDDC(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>α max</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dim</td>
<td># it. κ</td>
<td># it. κ</td>
</tr>
<tr>
<td>10^2</td>
<td>16</td>
<td>21 2.12e3</td>
<td>10 5.47e0</td>
</tr>
<tr>
<td>10^4</td>
<td>64</td>
<td>28 2.87e5</td>
<td>10 5.31e0</td>
</tr>
<tr>
<td>10^6</td>
<td>36</td>
<td>44 2.89e7</td>
<td>10 5.31e0</td>
</tr>
<tr>
<td>10^8</td>
<td>64</td>
<td>64 3.88e9</td>
<td>10 5.31e0</td>
</tr>
</tbody>
</table>

Fig. 4. Distribution of the coefficient in the two-channels test case when α max = 10^6. The coarse objects of PB-BDDC(ce) are shown on the interface with corners labeled by stars and DOFs in edges labeled by circles.

4.2. 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 · 10^-2 from one of the following three lines:

\[ L_1 : x_1 - x_2 - 0.2 = 0, \]
\[ L_2 : x_1 + x_2 - 0.7 = 0, \]
\[ L_3 : x_1 - 0.7x_2 - 0.7 = 0. \]

The coefficient α 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

\[ \text{mod} \left( \text{floor}(10x_i), 2 \right) = 1, \text{ for } i = 1, 2. \]

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

\[ \alpha_{\tau} = (\alpha_{\text{max}}/10)^{1/5\times\text{floor}(0.5\times\text{floor}(10x(\cdot)_1(c_\tau))+1)}, \]

where \( c_\tau \) is the centroid of \( \tau \).

This manuscript is for review purposes only.
The coefficient in (20) is: a) constant in each inclusion b) increasing from left to right

c) increasing as \( \alpha_{\text{max}} \) increases and d) always belongs to \((1, \alpha_{\text{max}})\). For the rest of the
domain, we set \( \alpha = 1 \). The maximal contrast ratio in this experiment is \(10^8\).

We can see from Table 2 that as \( \alpha_{\text{max}} \) becomes larger the condition number and
the number of iterations associated with the standard BDDC(ce) method increases
significantly. In contrast, both variant of the PB-BDDC methods, 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.

### Table 2

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

<table>
<thead>
<tr>
<th>( \alpha_{\text{max}} )</th>
<th>BDDC(ce)</th>
<th>PB-BDDC(ce)</th>
<th>PB-BDDC(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dim →</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 10^2 )</td>
<td>16</td>
<td>89</td>
<td>39</td>
</tr>
<tr>
<td>( 10^4 )</td>
<td>45</td>
<td>8.33e4</td>
<td>8.93e0</td>
</tr>
<tr>
<td>( 10^6 )</td>
<td>82</td>
<td>6.02e6</td>
<td>8.79e0</td>
</tr>
<tr>
<td>( 10^8 )</td>
<td>97</td>
<td>5.30e8</td>
<td>8.76e0</td>
</tr>
</tbody>
</table>

#### 4.3. Complex channels.

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

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) precon-
ditioner increase significantly as $\alpha_{\text{max}}$ increases. The reason is that there are some
pairs of channels share a corner but not an edge. In PB-BDDC(e), none of these
corners are selected as a coarse objects. Consequently, there is no acceptable path
with TOL independent of the contrast between the associated paired of channels (PB
subdomains) and Assumption 7 does not hold. By including a small number of these
critical corners (represented by stars in Figure 6) in order to satisfy Assumption 7, the
resulting preconditioner, labeled PB-BDDC([c|e]), is perfectly robust w.r.t changes in
the contrast of the coefficient (see Table 3).

**Table 3**

| dim $\alpha_{\text{max}}$ | BDDC(ce) | PB-BDDC(ce) | PB-BDDC(e) | PB-BDDC([c|e]) |
|----------------------------|-----------|-------------|-------------|-----------------|
| $10^2$                     | 22 (2.99e3) | 12 (6.09e0) | 19 (1.25e3) | 13 (1.21e1)    |
| $10^4$                     | 43 (3.82e5) | 12 (5.94e0) | 34 (1.62e5) | 13 (1.27e1)    |
| $10^6$                     | 70 (3.83e7) | 12 (5.94e0) | 51 (1.63e7) | 13 (1.27e1)    |
| $10^8$                     | 95 (6.45e9) | 12 (5.94e0) | 99 (1.63e9) | 13 (1.27e1)    |

**4.4. Sinusoidal variation.** In this experiment, we consider a coefficient that
varies like a sinusoid. We use a finer uniform triangular mesh of size $h = 1/144$. For
an element $\tau \in T$, the coefficient $\alpha_{\tau}$ is defined by

$$\log_{10}(\alpha_{\tau}) = \kappa \sin(w\pi(x_1(c_{\tau}) + x_2(c_{\tau}))) + \alpha_{\text{shift}},$$

where $\kappa = 3$, $w = 14$, and $c_{\tau}$ is the the centroid of $\tau$. We note that when $\kappa$ and/or
$w$ become larger the problem is more difficult. The distribution when $\alpha_{\text{shift}} = 0$ is

---

This manuscript is for review purposes only.
shown in Figure 7. It is as if there are many channels going through subdomain edges at the same time.

In this test case, the coefficient varies very rapidly. We test the standard BDDC method and the rPB-BDDC method introduced in subsection 3.5, by allowing the upper bound \( r \) for the maximal contrast in each PB subdomain to vary among \( \{10^1, 10^2, 10^3\} \). Only iteration counts are reported as the condition number estimation becomes too expensive for the mesh being used.

This is a difficult problem and the standard BDDC(ce) method requires almost a hundred iterations to converge (see Table 4). The relaxed physics-based methods, rPB-BDDC(ce) and rPB-BDDC(e), are able to significantly reduce the number of iterations. That comes with the cost of solving larger coarse problems. However, by using a suitable threshold \( r \), we can obtain a decent preconditioner, e.g., rPB-BDDC(e) with \( r = 10^3 \), which requires only 11 iterations using a reasonably small coarse space of size 64. In addition, the rPB-BDDC method is also perfectly robust with shifting in the value of the coefficient. The iteration count does not change when \( \alpha_{\text{shift}} \) takes values in \( \{0, 6\} \).

Table 4

<table>
<thead>
<tr>
<th></th>
<th>BDDC(ce)</th>
<th>rPB-BDDC(ce)</th>
<th>rPB-BDDC(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r )</td>
<td>( \text{dim} )</td>
<td>( 10 )</td>
<td>( 10^2 )</td>
</tr>
<tr>
<td>( \alpha_{\text{shift}} = 0 )</td>
<td># it.</td>
<td>92</td>
<td>7</td>
</tr>
<tr>
<td>( \alpha_{\text{shift}} = 6 )</td>
<td># it.</td>
<td>99</td>
<td>7</td>
</tr>
</tbody>
</table>
4.5. Log-Normal. In this test case, we test the performance of the rPB-BDDC method for a log-normal distribution of the coefficient. This type of distribution is particularly important for geoscience and petroleum engineering applications. We consider \( \alpha_{\text{cont}}(x, w) = 10^Z(x, w) \), where \( Z(x, w) \) is a Gaussian random field with zero mean and Gaussian covariance

\[
C(x, y) = \sigma^2 \exp\left(-\frac{\|x - y\|^2}{\ell^2}\right), \quad \text{with} \quad \sigma = 1.5, \quad \ell^2 = 1e-3.
\]

For this experiment, a uniform triangular mesh of size \( h = 1/128 \) is utilized. Using the spectral decomposition method described in [37], we are able to obtain a realization of \( \alpha_{\text{cont}}(x, w) \) at mesh vertices. The piecewise coefficient \( \alpha_\tau \) on an element \( \tau \) is then defined as the average of \( \alpha_{\text{cont}}(x, w) \) at the three vertices. The distribution of \( \alpha \) with a partition obtained from METIS [26] is shown in Figure 8. The contrast ratio in this test case is nearly \( 10^{10} \). The coarse objects of rPB-BDDC(ce) when \( r = 10^2 \) are also illustrated.

In Table 5, we can see that, compared to the standard BDDC(ce) method, rPB-BDDC(ce) and rPB-BDDC(e) preconditioners require much fewer iterations to converge. They, however, have a larger coarse space. By adjusting the threshold for the maximal contrast in each object, we can reduce the size of the coarse space while maintaining a reasonably fast convergence. This is clearly illustrated in Table 5.

5. Conclusions. In this work, we have proposed a novel type of BDDC preconditioners that are robust for heterogeneous 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 sub-objects. 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.
Table 5
Comparison of the iteration count in the log-normal test case.

<table>
<thead>
<tr>
<th>r</th>
<th>rPB-BDDC(ce)</th>
<th>rPB-BDDC(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim</td>
<td>49</td>
<td>488 284 239</td>
</tr>
<tr>
<td># it.</td>
<td>77</td>
<td>16 25 28</td>
</tr>
</tbody>
</table>

In cases where the physical coefficient is constant in each coarse sub-object, we are able to prove that the associated condition number can be bounded independent of the number of the subdomains and the contrast of the physical coefficient. In other words, the new preconditioner is scalable and robust for heterogeneous problems.

Apart from the new set of coarse objects and a new weighting operator, the (r)PB-BDDC preconditioners are very much the same as the standard BDDC preconditioner. As a result, the implementation of the new preconditioners involve a very simple modification of the standard BDDC implementation. In all of our experiments, the new preconditioners deliver 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 [25, 45, 24, 43, 44, 22, 23, 47, 15, 49, 48, 28, 27, 29, 36, 24], our new methods do not involve any type of eigenvalue or auxiliary problems.

For further work, we want to implement the new preconditioners in the extremely scalable BDDC code in FEMPAR [3, 4, 5, 6]. The multilevel extension and the task-overlapping implementation are particularly interesting in the (r)PB-BDDC case due to generally larger coarse problem. With such extremely scalable implementation, we are interested in applying our new preconditioners to realistic 3D problems, e.g., in geoscience applications.

REFERENCES


This manuscript is for review purposes only.
This manuscript is for review purposes only.


