

# Applications of Mathematics

---

Bedřich Sousedík

On adaptive BDDC for the flow in heterogeneous porous media

*Applications of Mathematics*, Vol. 64 (2019), No. 3, 309–334

Persistent URL: <http://dml.cz/dmlcz/147720>

## Terms of use:

© Institute of Mathematics AS CR, 2019

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

ON ADAPTIVE BDDC FOR THE FLOW  
IN HETEROGENEOUS POROUS MEDIA

BEDŘICH SOUSEDÍK, Baltimore

Received August 15, 2018. Published online April 23, 2019.

*Abstract.* We study a method based on Balancing Domain Decomposition by Constraints (BDDC) for numerical solution of a single-phase flow in heterogeneous porous media. The method solves for both flux and pressure variables. The fluxes are resolved in three steps: the coarse solve is followed by subdomain solves and last we look for a divergence-free flux correction and pressures using conjugate gradients with the BDDC preconditioner. Our main contribution is an application of the adaptive algorithm for selection of flux constraints. Performance of the method is illustrated on the benchmark problem from the 10th SPE Comparative Solution Project (SPE 10). Numerical experiments in both 2D and 3D demonstrate that the first two steps of the method exhibit some numerical upscaling properties, and the adaptive preconditioner in the last step allows a significant decrease in the number of iterations of conjugate gradients at a small additional cost.

*Keywords:* iterative substructuring; balancing domain decomposition; BDDC; multiscale methods; adaptive methods, flow in porous media; reservoir simulation; SPE 10 benchmark

*MSC 2010:* 65F08, 65F10, 65M55, 65N55

## 1. INTRODUCTION

The Balancing Domain Decomposition by Constraints (BDDC), proposed independently by Cros [7], Dohrmann [9], and Fragakis and Papadarakakis [16], is one of the most popular methods of iterative substructuring. The method was developed as a preconditioner for the solution of systems of linear equations obtained by finite element discretizations of elliptic problems, and it was originally derived as a primal counterpart of the Finite Element Tearing and Interconnecting—Dual, Primal (FETI-DP) method by Farhat et al. [14], [15]. Over the years the BDDC

---

Supported by the U.S. National Science Foundation under grant DMS1521563.

has been extended to other types of problems, for example to the nearly incompressible elasticity by Dohrmann [10], the Stokes problem by Li and Widlund [27], or advection-diffusion problems by Tu and Li [26], [47]. It is also relatively straightforward to extend the BDDC into multiple levels, as noted by Dohrmann [9]. The three-level methods were developed in two and three dimensions by Tu [44], [45], and Mandel et al. [30] extended the algorithm into a multilevel method within a more general multispace BDDC setting. Another class of problems, important in the context of this paper, is the flow in porous media based on *mixed* and *mixed-hybrid* finite element discretizations. Probably the first domain decomposition methods of this class were proposed by Glowinski and Wheeler [17]. Their Method II was preconditioned using BDD by Cowsar et al. [6], using BDDC by Tu [43], and Šístek et al. [37] extended this methodology to the flow in porous media with combined mesh dimensions. This approach is regarded as *hybrid*, because the method iterates on a system of *dual* variables (as Lagrange multipliers) enforcing the continuity of flux variables across the substructure interfaces. An alternative strategy, retaining the original *primal* variables was proposed by Tu [42], [46], who combined the BDDC preconditioner with an earlier algorithmic framework developed by Ewing and Wang [13], cf. also Mathew [32], which allows to solve the saddle-point problem obtained from *mixed* finite element discretization by conjugate gradients. The Nested BDDC by Sousedík [38] provided a multilevel extension by applying the framework from [42] recursively. Most recently, Zampini and Tu [50] presented another approach to multilevel BDDC including an adaptive coarse space construction, which relies on a special, so-called, deluxe scaling.

There are two main ingredients of a BDDC method: a coarse space, which is defined by *constraints* on the values of degrees of freedom, and a scaling (averaging) operator, which provides a mapping between the solution space and the space in which the solves in the preconditioner are performed. The algorithm for adaptive selection of constraints for both methods, the BDDC and FETI-DP, was originally proposed by Mandel and Sousedík [29]. The algorithm was later generalized in a joint work with Šístek [31] into three spatial dimensions and implemented for the BDDC using an approach inspired by a partial subassembly and a change of variables by Li and Widlund [28]. Finally, we also reformulated the algorithm to treat the coarse space explicitly [39]. We note that there are many other approaches to the adaptive construction of the coarse spaces in BDDC, see [35] and the references therein, as well as for BDD, see e.g. [40] and FETI-DP, see e.g. [21]. There have been several scalings studied in the literature. In the *multiplicity* scaling, the weights are chosen proportionally to the number of subdomains sharing a given degree of freedom, and it is regarded as not robust for coefficient jumps. The  $\varrho$ -*scaling* leads to robustness, but it relies on the knowledge of the problem coefficients [23]. The *stiffness* scaling is

based on the diagonal of the stiffness matrix, but in some cases with irregular meshes it may lead to high condition numbers [34], [36]. All these scalings involve diagonal matrices. Finally, the *deluxe* scaling introduced in [11] uses dense matrices, which are computed from inverses of localized Schur complements. It has been observed to be quite robust [33], [50] but also computationally intensive.

In this paper, we build on the primal strategy. The starting point is the two-level algorithm from [38], which we combine with adaptive selection of constraints following [39] and apply it to the flow in heterogeneous porous media. To this end, we use a reservoir from the 10th SPE Comparative Solution Project (SPE 10), cf. e.g. [1], [5] as the benchmark problem. The BDDC method from [38] solves for both the flux and pressure variables. The fluxes are resolved in three steps: the coarse solve is followed by mutually independent subdomain solves, and last we look for a divergence-free flux correction and pressure using conjugate gradients (CG) with the BDDC preconditioner. The coarse solve in the first step is exactly the same as the coarse solve used in the BDDC preconditioner in the step three. It is assumed that the initial constraints preserve the iterates in a *balanced* subspace, in which the preconditioned operator is positive definite. Our goal here is to adapt the method to the flow in realistic reservoirs, characterized by highly heterogeneous permeability coefficients in as simple way as possible. In particular, we translate the ideas used for elliptic problems in [39] to mixed formulations of the flow in porous media discretized by the lowest-order Raviart-Thomas finite elements (RT0). The main component of the extension is the use of additional adaptive flux coarse basis functions. The starting point is the condition number bound formulated as a generalized eigenvalue problem, which is replaced by a number of local eigenvalue problems formulated for pairs of adjacent subdomains, and the eigenvectors, corresponding to the eigenvalues larger than a target condition number are used to construct the additional flux coarse basis functions. We note that from this perspective our method can be viewed as a way of numerical upscaling via the coarse basis functions known from the BDDC. Unlike [50] we do not use a change of basis and partial assembly of operators, and we also illustrate that for this problem the multiplicity scaling in combination with the adaptive algorithm and a simple diagonal rescaling of the pressure block in the setup of the problem is sufficient to construct a robust algorithm. Numerical experiments in both 2D and 3D demonstrate that the first two steps of the method exhibit some numerical upscaling properties, and the convergence rate of conjugate gradients in the last step can be estimated a priori in the setup of the adaptive algorithm.

The paper is organized as follows. In Section 2 we introduce the model problem, in Section 3 we recall the BDDC method and the preconditioner, in Section 4 we formulate the algorithm for adaptive selection of the flux constraints, in Section 5 we

discuss some details of implementation, in Section 6 we present results of numerical experiments, and finally in Section 7 we summarize and conclude our work.

For convenience, we identify finite element functions with the vectors of their coefficients in the corresponding finite element basis. These coefficients are also called *variables* or *degrees of freedom*. At a few places we will also identify linear operators with their matrices, in bases that will be clear from the context. For a symmetric positive definite bilinear form  $a$ , we will denote the energy norm by  $\|u\|_a = \sqrt{a(u, u)}$ .

## 2. MODEL PROBLEM

Let  $\Omega$  be a bounded domain in  $\mathbb{R}^n$ , where  $n = 2$  or  $3$ . We would like to find the solution of the following mixed problem, which combines Darcy's law relating flux  $\mathbf{u}$  and pressure  $p$ , and the equation of continuity,

$$(2.1) \quad k^{-1}\mathbf{u} + \nabla p = 0 \quad \text{in } \Omega,$$

$$(2.2) \quad \nabla \cdot \mathbf{u} = f_\Omega \quad \text{in } \Omega,$$

$$(2.3) \quad p = p_N \quad \text{on } \partial\Gamma_N,$$

$$(2.4) \quad \mathbf{u} \cdot \mathbf{n} = g_E \quad \text{on } \partial\Gamma_E,$$

where  $\partial\Omega = \bar{\Gamma}_E \cup \bar{\Gamma}_N$ , and  $\mathbf{n}$  denotes the unit outward normal of  $\Omega$ . The coefficient  $k = k_p/\mu$ , where  $k_p$  is the permeability of the porous medium and  $\mu$  is the viscosity of the fluid. For simplicity, we will set  $\mu = 1$  and so  $k = k_p$ . Without loss of generality we will also assume that  $\Gamma_N = \emptyset$ , which requires the compatibility condition

$$(2.5) \quad - \int_{\Omega} f_\Omega \, dx + \int_{\partial\Omega} g_E \, ds = 0,$$

and the pressure  $p$  will be uniquely determined up to an additive constant. We will further assume that  $g_E = 0$ . These assumptions motivate the definition of a space

$$\mathbf{H}_0(\Omega; \text{div}) = \{\mathbf{v} \in L^2(\Omega); \nabla \cdot \mathbf{v} \in L^2(\Omega) \text{ and } \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\},$$

equipped with the norm

$$\|\mathbf{v}\|_{\mathbf{H}_0(\Omega; \text{div})}^2 = \|\mathbf{v}\|_{L^2(\Omega)}^2 + H_\Omega^2 \|\nabla \cdot \mathbf{v}\|_{L^2(\Omega)}^2,$$

where  $H_\Omega$  is the characteristic size of  $\Omega$ , and the definition of a space

$$L_0^2(\Omega) = \left\{ q: q \in L^2(\Omega) \text{ and } \int_{\Omega} q \, dx = 0 \right\}.$$

The weak form of the problem we wish to solve, is

$$(2.6) \quad \int_{\Omega} k^{-1} \mathbf{u} \cdot \mathbf{v} \, dx - \int_{\Omega} p(\nabla \cdot \mathbf{v}) \, dx = 0 \quad \forall \mathbf{v} \in \mathbf{H}_0(\Omega; \text{div}),$$

$$(2.7) \quad - \int_{\Omega} (\nabla \cdot \mathbf{u}) q \, dx = - \int_{\Omega} f_{\Omega} q \, dx \quad \forall q \in L_0^2(\Omega).$$

We refer, e.g., to the monographs [4], [41] for additional details and discussion.

Next, let  $U$  be the lowest-order Raviart-Thomas (RT0) finite element space with a zero normal component on  $\partial\Omega$  and let  $Q$  be a space of piecewise constant finite element basis functions with zero mean on  $\Omega$ . These two spaces, defined on the triangulation  $\mathcal{T}_h$  of  $\Omega$ , where  $h$  denotes the mesh size, are finite dimensional subspaces of  $\mathbf{H}_0(\Omega; \text{div})$  and  $L_0^2(\Omega)$ , respectively, and they satisfy a uniform inf-sup condition, see [4]. Let us define the bilinear forms and the right-hand side by

$$(2.8) \quad a(u, v) = \int_{\Omega} k^{-1} \mathbf{u} \cdot \mathbf{v} \, dx,$$

$$(2.9) \quad b(u, q) = - \int_{\Omega} (\nabla \cdot \mathbf{u}) q \, dx,$$

$$(2.10) \quad \langle f, q \rangle = - \int_{\Omega} f_{\Omega} q \, dx.$$

In the mixed finite element approximation of problem (2.6)–(2.7), we would like to find a pair of fluxes and pressures  $(u, p) \in (U, Q)$  such that

$$(2.11) \quad a(u, v) + b(v, p) = 0 \quad \forall v \in U,$$

$$(2.12) \quad b(u, q) = \langle f, q \rangle \quad \forall q \in Q.$$

We note that  $Q$  is a finite-dimensional subspace of  $L_0^2(\Omega)$  and therefore the unique solvability of the mixed problem (2.11)–(2.12) is guaranteed.

In the next section, we will describe the components of the two-level Nested BDDC, which allows an efficient iterative solution of problem (2.11)–(2.12).

### 3. THE BDDC METHOD

Let us consider a decomposition of  $\Omega$  into a set of nonoverlapping subdomains  $\Omega^i$ ,  $i = 1, \dots, N$ , also called substructures, forming a quasi-uniform triangulation of  $\Omega$  and denote the characteristic subdomain size by  $H$ . Each substructure is a union of finite elements with a matching discretization across the substructure interfaces. Let  $\Gamma^i = \partial\Omega^i \setminus \partial\Omega$  be the set of boundary degrees of freedom of a substructure  $\Omega^i$  shared

with another substructure  $\Omega^j$ ,  $j \neq i$ , and define the interface by  $\Gamma = \bigcup_{i=1}^N \Gamma^i$ . Let us define a *face* as an intersection  $\Gamma^{ij} = \Gamma^i \cap \Gamma^j$ ,  $i \neq j$  and let us denote by  $\mathcal{F}$  the set of all faces between substructures. Note that with respect to the RT0 discretization we define only *faces*, but no *corners* (nor *edges* in 3D) known from other types of substructuring.

We will solve problems similar to (2.11)–(2.12) on each substructure. As we have noted, such problems determine the pressure uniquely up to a constant, so we consider the decomposition of the pressure space

$$(3.1) \quad Q = Q_0 \oplus Q_I, \quad Q_I = Q^1 \times \dots \times Q^N,$$

where  $Q_0$  consists of functions that are constant in each subdomain and have zero average over the whole domain  $\Omega$ , and the product space  $Q_I$  consists of functions that have zero weighted average over one subdomain at a time. That is,

$$(3.2) \quad \int_{\Omega} q_0 \, dx = 0 \quad \forall q_0 \in Q_0 \quad \text{and} \quad \int_{\Omega^i} q^i \, dx = 0 \quad \forall q^i \in Q^i, \quad i = 1, \dots, N.$$

Next, let  $W^i$  be the space of flux finite element functions on a substructure  $\Omega^i$  such that all of their degrees of freedom on  $\partial\Omega^i \cap \partial\Omega$  are zero, and let

$$W = W^1 \times \dots \times W^N.$$

Hence  $U \subset W$  can be viewed as the subspace of flux functions from  $W$  such that  $u \cdot \mathbf{n}$  is continuous across substructure interfaces. Define  $U_I \subset U$  as the subspace of flux functions such that  $u \cdot \mathbf{n}$  is zero on the interface  $\Gamma$ , i.e., the space of “interior” flux functions, and let us also define a mapping  $P: w \in W \mapsto u_I \in U_I$  such that

$$\begin{aligned} a(u_I, v_I) + b(v_I, p_I) &= a(w, v_I) \quad \forall v_I \in U_I, \\ b(u_I, q_I) &= b(w, q_I) \quad \forall q_I \in Q_I. \end{aligned}$$

Functions from  $(I - P)W$  will be called Stokes harmonic, cf. [41], Section 9.4.2.

Let  $\widehat{W}$  be the space of Stokes harmonic functions that are continuous across substructure interfaces, and such that

$$(3.3) \quad U = \widehat{W} \oplus U_I, \quad \widehat{W} \perp_a U_I.$$

We note that from the divergence theorem, for all  $u_I \in U_I$  and  $q_0 \in Q_0$ , we obtain

$$b(u_I, q_0) = - \int_{\Omega} (\nabla \cdot u_I) q_0 \, dx = 0.$$

The BDDC is a two-level method characterized by a selection of certain *coarse degrees of freedom*. In the present setting these will be flux averages over faces shared by a pair of substructures at a time and pressure averages over each substructure. Let us denote by  $\widetilde{W} \subset (I - P)W$  the subspace of Stokes harmonic functions such that their flux coarse degrees of freedom on adjacent substructures coincide; for this reason we will use the terms coarse degrees freedom and *constraints* interchangeably. Specifically, we define a zero-net flux constraint for a face  $\Gamma^{ij}$  as

$$(3.4) \quad \int_{\Gamma^{ij}} (w^i - w^j) \cdot \mathbf{n}^i \, ds = 0, \quad w^i \in W^i, \quad w^j \in W^j,$$

where  $\mathbf{n}^i$  denotes the unit outward normal of  $\Omega^i$ .

**Assumption 3.1.** Initial flux constraints (3.4) are prescribed over all faces.

This set of initial constraints will be enriched by the adaptive method described in Section 4. Now, let us define  $\widetilde{W}_\Pi \subset \widetilde{W}$  as the subspace of functions with values given by the flux coarse degrees of freedom between adjacent substructures, and such that they are Stokes harmonic, and let us also define  $\widetilde{W}_\Delta \subset \widetilde{W}$  as the subspace of all functions such that their flux coarse degrees of freedom vanish. The functions in  $\widetilde{W}_\Pi$  are uniquely determined by the values of their coarse degrees of freedom, and

$$(3.5) \quad \widetilde{W} = \widetilde{W}_\Delta \oplus \widetilde{W}_\Pi.$$

The next ingredient is the projection  $E: \widetilde{W} \rightarrow \widehat{W}$  defined by taking a weighted average of the corresponding degrees of freedom on substructure interfaces, cf. Remark 3.3.

In implementation, we define  $\widetilde{W}$  using a matrix  $C_U$ , which is a block diagonal with blocks  $C_U^i$ ,  $i = 1, \dots, N$ , and is constructed exactly as matrix  $C$  in [29], Section 2.3,

$$(3.6) \quad \widetilde{W} = \{w \in (I - P)W : C_U(I - E)w = 0\}.$$

The values  $C_U v$  will be called local flux coarse degrees of freedom, and the space  $\widetilde{W}$  consists of all functions such that their flux coarse degrees of freedom on adjacent substructures have zero jumps. The decomposition of the space  $Q_I$  given by (3.1) can be also managed by constraints. We remark that this is somewhat non-standard practice in substructuring, because the constraints are commonly related only to the degrees of freedom at the interfaces. So, we define a space  $Q^i$ , for  $i = 1, \dots, N$ , as

$$(3.7) \quad Q^i = \{(q \in Q) |_{\Omega^i} : C_Q^i q = 0\},$$

where the matrices  $C_Q^i$  are selected so that (3.2) is satisfied. In implementation,  $C_Q^i$  is a row vector with entries given by volumes of finite elements in subdomain  $i$ . Now we have all ingredients to recall the two-level BDDC method [38], Algorithm 2.



**Algorithm 3.2** (BDDC method). Find the solution  $(u, p) \in (U, Q)$  of problem (2.11)–(2.12) by computing:

1. the coarse component  $u_0 \in \widehat{W}$ : solving  $(\widetilde{w}_0, p_0) \in (\widetilde{W}_\Pi, Q_0)$  from

$$(3.8) \quad a(\widetilde{w}_0, \widetilde{v}_\Pi) + b(\widetilde{v}_\Pi, p_0) = 0 \quad \forall \widetilde{v}_\Pi \in \widetilde{W}_\Pi,$$

$$(3.9) \quad b(\widetilde{w}_0, q_0) = \langle f, q_0 \rangle \quad \forall q_0 \in Q_0,$$

dropping  $p_0$ , and applying the projection

$$u_0 = E\widetilde{w}_0;$$

2. the substructure components  $(u_I, p_I) \in (U_I, Q_I)$  from

$$a(u_I, v_I) + b(v_I, p_I) = -a(u_0, v_I) \quad \forall v_I \in U_I,$$

$$b(u_I, q_I) = \langle f, q_I \rangle - b(u_0, q_I) \quad \forall q_I \in Q_I,$$

dropping  $p_I$ , and adding the solutions as

$$(3.10) \quad u^* = u_0 + u_I;$$

3. the correction and the pressure  $(u_{\text{corr}}, p) \in (U, Q)$  from

$$(3.11) \quad a(u_{\text{corr}}, v) + b(v, p) = -a(u^*, v) \quad \forall v \in U,$$

$$(3.12) \quad b(u_{\text{corr}}, q) = 0 \quad \forall q \in Q.$$

Specifically, use the CG method with the BDDC preconditioner defined in Algorithm 3.4, using the same setup of the coarse problem as in (3.8)–(3.9).

Finally, the flux variables are obtained as

$$u = u^* + u_{\text{corr}}.$$

**Remark 3.3.** The difference between problems (2.11)–(2.12) and (3.11)–(3.12) is that the latter problem has a vanishing second component, and therefore the correction  $u_{\text{corr}}$  is divergence-free by (3.12). Also, we note that the initial flux constraints constructed according to (3.4) do not allow scaling weights in the scaling operator  $E$  to vary along the interface in order for  $u^*$  to satisfy

$$b(u^*, q) = \langle f, q \rangle \quad \forall q \in Q.$$

Therefore, in our numerical experiments, we use the multiplicity scaling unless the coefficient jumps are aligned with subdomain interfaces, see also [38], Remark 2.

The application of the BDDC preconditioner for the computation of  $u_{corr}$  using two- or three-level method was studied by Tu [42], [46]. In [38], we applied Algorithm 3.2 recursively. Here, we will introduce a specific construction of the space  $\widetilde{W}_\Pi$  but before doing so, let us discuss Step 3 of Algorithm 3.2 in more detail.

The first step in substructuring is typically the reduction of the problem to interfaces. In particular, problem (3.11)–(3.12) is reduced to finding  $(\widehat{w}, p_0) \in (\widehat{W}, Q_0)$  such that

$$(3.13) \quad a(\widehat{w}, \widehat{v}) + b(\widehat{v}, p_0) = \langle f^*, \widehat{v} \rangle \quad \forall \widehat{v} \in \widehat{W},$$

$$(3.14) \quad b(\widehat{u}, q_0) = 0 \quad \forall q_0 \in Q_0,$$

where  $f^* \in \widehat{W}'$  is the reduced right-hand side. In implementation, the interiors are eliminated by the static condensation, problem (3.13)–(3.14) is solved iteratively, and the interiors  $(u_I, p_I) \in (U_I, Q_I)$  are recovered in the post-correction. The key observation is, cf. [41], Section 9.4.2, that if we define a *balanced* subspace

$$\widehat{W}_B = \{\widehat{w} \in \widehat{W} : b(\widehat{w}, q_0) = 0 \quad \forall q_0 \in Q_0\},$$

problem (3.13)–(3.14) becomes equivalent to the positive definite problem

$$\widehat{u} \in \widehat{W}_B : a(\widehat{u}, \widehat{v}) = \langle f^*, \widehat{v} \rangle \quad \forall \widehat{v} \in \widehat{W}_B.$$

This observation justifies the use of the CG method preconditioned by the BDDC provided that the initial guess is balanced, e.g., zero, and the outputs of the preconditioner are also balanced. It also implies that the iterates are effectively performed with the flux unknowns, and the pressure components  $p_0$  are resolved in the coarse correction of the preconditioner. The precise formulation of the two-level BDDC preconditioner for saddle-point problems follows. It is the reduced variant of [38], Algorithm 3.

**Algorithm 3.4** (BDDC preconditioner). Define the preconditioner  $(r_B, 0) \in (\widehat{W}', Q'_0) \mapsto (\widehat{w}, p_0) \in (\widehat{W}, Q_0)$  by computing:

1. the coarse correction  $(w_\Pi, p_0) \in (\widetilde{W}_\Pi, Q_0)$  from

$$(3.15) \quad a(w_\Pi, z_\Pi) + b(z_\Pi, p_0) = \langle r_B, E z_\Pi \rangle \quad \forall z_\Pi \in \widetilde{W}_\Pi,$$

$$(3.16) \quad b(w_\Pi, q_0) = 0 \quad \forall q_0 \in Q_0;$$

2. the substructure correction  $w_\Delta \in \widetilde{W}_\Delta$  from

$$a(w_\Delta, z_\Delta) + b(z_\Delta, p_{I\Delta}) = \langle r_B, E z_\Delta \rangle \quad \forall z_\Delta \in \widetilde{W}_\Delta,$$

$$b(w_\Delta, q_I) = 0 \quad \forall q_I \in Q_I;$$

3. the sum and average of the two corrections

$$(3.17) \quad \widehat{w} = E(w_\Pi + w_\Delta).$$

In order to state the condition number bound, we also need to introduce a larger space of *balanced* functions  $\widetilde{W}_B$  such that  $\widehat{W}_B \subset \widetilde{W}_B$  defined as

$$\widetilde{W}_B = \{w \in \widetilde{W} : b(v, q_0) = 0 \quad \forall q_0 \in Q_0\}.$$

The space  $\widetilde{W}_\Pi$  is also balanced, i.e.,  $\widetilde{W}_\Pi \subset \widetilde{W}_B$  by (3.16). Then also the output of the preconditioner (3.17) satisfies  $\widehat{w} \in \widetilde{W}_B$ , and we refer to [38], Lemma 3 for the proof.

Finally, we formulate the condition number bound. If we note that  $E$  is a projection, it is the same as [38], Theorem 4 or [42], Theorem 6.1, cf. also [29], Theorem 3.

**Theorem 3.5.** *The condition number  $\kappa$  of the BDDC preconditioner from Algorithm 3.4 satisfies*

$$(3.18) \quad \kappa \leq \omega = \max \left\{ \sup_{w \in \widetilde{W}_B} \frac{\|(I - E)w\|_a^2}{\|w\|_a^2}, 1 \right\} \leq C \left( 1 + \log \frac{H}{h} \right)^2.$$

The bound  $\omega$  in (3.18) inspires the adaptive selection of the flux constraints.

#### 4. ADAPTIVE SELECTION OF THE FLUX CONSTRAINTS

The basic idea is the same as in our previous work on adaptive BDDC for elliptic problems [29], [31], [39]. The bound  $\omega$  in (3.18) is equal to the maximal eigenvalue  $\lambda_{\max}$  of the generalized eigenvalue problem

$$(4.1) \quad w \in \widetilde{W}_B : a((I - E)w, (I - E)z) = \lambda a(w, z) \quad \forall z \in \widetilde{W}_B.$$

From the Courant-Fisher-Weyl minimax principle, cf. e.g. [8], Theorem 5.2, the bound  $\omega$  can be decreased by adding constraints in the definition of the space  $\widetilde{W}_B$ :

**Lemma 4.1** ([31], [39]). *The generalized eigenvalue problem (4.1) has eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ . Denote the corresponding eigenvectors by  $w_l$ . Then, for any  $k = 1, \dots, n - 1$ , and any linear functionals  $L_l, l = 1, \dots, k$ ,*

$$\max \left\{ \frac{\|(I - E)w\|_a^2}{\|w\|_a^2} : w \in \widetilde{W}_B, L_l(w) = 0 \quad \forall l = 1, \dots, k \right\} \geq \lambda_{k+1},$$

with equality if

$$L_l(w) = a((I - E)w_l, (I - E)w).$$

Because solving the global eigenvalue problem (4.1) is computationally expensive, we replace it by a collection of much smaller problems defined for all pairs of adjacent substructures, where a pair of substructures is adjacent if they share a face. All quantities associated with a pair of adjacent substructures  $\Omega^i$  and  $\Omega^j$  will be denoted by a double superscript  $ij$ . In particular, we define  $W^{ij} = W^i \times W^j$ , and the local space  $\widetilde{W}_B^{ij}$  of Stokes harmonic functions that satisfy the initial constraints at the face  $\Gamma^{ij}$  by

$$(4.2) \quad \widetilde{W}_B^{ij} = \{w \in (I - P^{ij})W^{ij} : C_U^{ij}(I - E^{ij})w = 0\}.$$

We note that the space  $\widetilde{W}_B^{ij}$  is balanced, which is an implication of Assumption 3.1.

In these settings (4.1) becomes a *local* problem to find  $w \in \widetilde{W}_B^{ij}$  such that

$$(4.3) \quad a^{ij}((I - E^{ij})w, (I - E^{ij})z) = \lambda a^{ij}(w, z) \quad \forall z \in \widetilde{W}_B^{ij}.$$

The bilinear form  $a^{ij}$  is associated on  $\widetilde{W}_B^{ij}$  with the Schur complement  $S^{ij}$  defined with respect to the interfaces  $\Gamma^i, \Gamma^j$ , and is positive-definite, cf. [42], Lemma 3.1.

Now we can proceed in the same way as in [39]. Let us denote by  $\mathcal{C}$  the matrix corresponding to  $C_U^{ij}(I - E^{ij})$ . The orthogonal projection onto the null  $\mathcal{C}$  is given by

$$\Pi = I - \mathcal{C}^T(\mathcal{C}\mathcal{C}^T)^{-1}\mathcal{C},$$

and we implement the local generalized eigenvalue problems (4.3) as

$$(4.4) \quad \Pi(I - E^{ij})^T S^{ij}(I - E^{ij})\Pi w = \lambda \Pi S^{ij}\Pi w,$$

which can be either solved using a dense eigenvalue solver [29] or eventually, since

$$\text{null}[\Pi S^{ij}\Pi] \subset \text{null}[\Pi(I - E^{ij})^T S^{ij}(I - E^{ij})\Pi],$$

a subspace iterations such as the LOBPCG method [24], which runs effectively in the factorspace, could be also used. From (4.4), we wish the constraints to satisfy

$$L_l(w) = w_l^T \Pi(I - E^{ij})^T S^{ij}(I - E^{ij})\Pi w = 0.$$

That is, we would add into the matrix  $C_U^{ij}$  the rows

$$(4.5) \quad c_l^{ij} = w_l^T \Pi(I - E^{ij})^T S^{ij}(I - E^{ij})\Pi,$$

but because of [39], Proposition 1 each row can be split as  $c_l^{ij} = [c_l^i \quad -c_l^j]$  and either half of  $c_l^{ij}$  is used to augment the matrices  $C_U^i$  and  $C_U^j$ , see (5.3). We note that, due to the discretization using RT0 elements, the added rows are readily available in the form used in substructuring. The adaptive BDDC algorithm follows.

**Algorithm 4.2 (Adaptive BDDC).** Find the smallest  $k$  for every two adjacent substructures to guarantee that  $\lambda_{k+1} \leq \tau$ , where  $\tau$  is a given tolerance threshold (the target condition number), and add the constraints (4.5) to the definition of  $\widetilde{W}$ .

After the adaptive constraints are added, we define the *heuristic condition number indicator* as the largest eigenvalue  $\omega^{ij}$  of all local eigenvalue problems (4.3), that is

$$(4.6) \quad \widetilde{\omega} = \max\{\omega^{ij} : \Omega^i \text{ and } \Omega^j \text{ are adjacent}\}.$$

**Remark 4.3.** It has been shown in [50], Theorem 4.3, see also [35], Theorem 3.10 and [33], Theorem 3.3, that the condition number  $\kappa$  of the adaptive BDDC operator satisfies

$$\kappa \leq \widetilde{\omega} N_F^2,$$

where  $N_F$  is the maximum number of faces of any subdomain. We note that this bound is pessimistic due to the factor  $N_F^2$ , and in fact we observed  $\kappa \approx \widetilde{\omega}$  in all experiments.

## 5. IMPLEMENTATION REMARKS

First, we describe the rescaling used to preserve numerical stability of the method with highly heterogeneous permeability coefficients. The variational problem (2.11)–(2.12) can be written in the matrix form as

$$(5.1) \quad \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}.$$

Assuming that the mesh size  $h \approx 1$ , the entries in  $A$  are  $O(k^{-1})$  and the entries in  $B$  are  $O(1)$ . In particular, in the case of the SPE 10 data set we get  $k^{-1} \approx 10^6 - 10^{12}$ , and we found that some of the subdomain matrices and the matrix of the coarse problem may appear numerically singular. Due to the discontinuous approximation of the pressure,  $B$  is a block-diagonal rectangular matrix. Each block corresponds to a particular subdomain, and it can be rescaled, e.g., by an average of the diagonal entries of  $A$  corresponding to the degrees of the freedom in this subdomain. Collecting this scaling coefficients in a diagonal matrix  $D$ , we replace (5.1) by

$$(5.2) \quad \begin{bmatrix} A & B^T D \\ DB & 0 \end{bmatrix} \begin{bmatrix} u \\ \overline{p} \end{bmatrix} = \begin{bmatrix} 0 \\ Df \end{bmatrix},$$

and the pressure is recovered at the end of computations as  $p = D\overline{p}$ .

**5.1. Coarse degrees of freedom.** The selection of the flux coarse degrees of freedom or, equivalently, flux constraints entails construction of the matrix  $C_U$  in the definition of the space  $\widetilde{W}$  by (3.6). Similarly, the selection of the pressure constraints, which facilitate the decomposition (3.1), entails construction of the matrices  $C_Q^i$ ,  $i = 1, \dots, N$ , in the definition of the spaces  $Q^i$  by (3.7). Following the standard practice in substructuring, in implementation we work with global and local degrees of freedom and the corresponding spaces, and vectors from these spaces are related by a restriction operator (a zero-one matrix). Therefore, the matrix  $C_U$  is constructed as a block-diagonal matrix using blocks  $C_U^i$  that select local flux coarse degrees of freedom from all degrees of freedom of substructure  $i$ , see [29], Section 2.3 for details. In the mixed finite element settings, each local coarse degrees of the freedom selection matrix is constructed simply by augmenting the matrix  $C_U^i$  by a row  $C_Q^i$  as

$$(5.3) \quad \begin{bmatrix} C_U^i \\ C_Q^i \end{bmatrix}, \quad i = 1, \dots, N,$$

and the matrices  $C_U^i$  may be further augmented by the adaptive algorithm, see (4.5).

**5.2. Solution of the local generalized eigenvalue problems.** The choice of an eigensolver for the eigenvalue problems (4.4) is a delicate one. In general, the decision whether to use a dense or a sparse eigensolver depends on the type of the eigenvalue problem, size of the substructures, dimension of the problem, availability of a preconditioner for a sparse solver, and conditioning and numerical sensitivity of the underlying problem. All these factors will clearly affect the overall computational cost and performance of the method. We note that the formulation (4.4) allows to use a matrix-free iterative method such as the LOBPCG [24] in the same way as for elliptic problems, including that it can be further preconditioned by a local version of the BDDC as suggested in [39], Section 5, see also [22]. However, we found that dense eigenvalue solvers are more suitable for the SPE 10 dataset due to their robustness, and we used MATLAB function `eig` in the numerical experiments.

**5.3. Computational cost.** Clearly, the two most computationally expensive parts of the method are the setup of the constraints by solving the set of the local eigenvalue problems, and the factorization of the coarse problem. There are many eigenvalue problems to be solved, but they are small and can be solved in parallel—this feature is similar to the setup of multiscale finite element methods [12]. Assuming that these can be solved efficiently, the bottleneck in computations is the factorization of the coarse problem. Specifically, it is crucial for the application of the method to appropriately balance the effort in the preconditioner and the global linear solver through a judicious choice of  $\tau$ . This could be, for example, achieved as

follows: one can partition the domain into subdomains balancing the sizes of subdomains and assuming certain size of the coarse problem (and ideally also taking into account the coefficient jumps and minimizing the size of interfaces), solve the set of local eigenvalue problems, and based on the eigenvalues determine the number of additional adaptive constraints (and hence the value of  $\tilde{\omega}$ ) which minimize the work needed to factor the coarse problem and the work needed by preconditioned conjugate gradients, including the coarse problem back-substitutions, needed to reduce the error to the desired accuracy based on the well-known error reduction formula of conjugate gradients, see e.g. [18], Theorem 10.2.6.

## 6. NUMERICAL EXPERIMENTS

We implemented the method in MATLAB and studied its convergence for problems with large variations in the permeability coefficients  $k$ . In all experiments we used relative residual tolerance  $10^{-6}$  as the convergence criterion for the conjugate gradients. First, we ran a test with jumps in  $k$  aligned with substructure interfaces, see Figure 1. For this problem we used stiffness scaling, which is in case of the lowest-order Raviart-Thomas (RT0) elements equivalent to the  $\varrho$ -scaling. This also implies that the stiffness scaling works well for irregular meshes (unlike for nodal elements). The conjugate gradients with the BDDC preconditioner converged in 15 steps and the approximate condition number computed from the Lánczos sequence in conjugate gradients was  $\kappa = 4.046$ ; with  $k = 1$  the method converged in 14 steps and  $\kappa = 4.050$ , see the rightmost column in Table 2. In the remaining experiments, we focused on problems with highly heterogeneous coefficients, and we used the multiplicity scaling. Specifically, we simulated the flow in a porous media given by Model 2 of the 10th SPE Comparative Solution Project [5], which is publicly available on the Internet<sup>1</sup> and, in particular, we used a MATLAB dataset described in [1]. The dimensions of the full model are  $1200 \times 2200 \times 170$  (ft), and the distribution of the coefficients  $k$  is given over a regular Cartesian grid with  $60 \times 220 \times 85$  grid-blocks. We used several layers and two 3D cutouts of the model for our numerical experiments. For the experiments in 2D, we used layers 1, 20, 60 and 85 shown in Figures 2–3. In the top layers 1 and 20 the permeability is relatively smooth, whereas the bottom layers 60 and 85 are fluvial and they are characterized by a spaghetti of narrow high-flow channels. In all layers the permeabilities range over at least six orders of magnitude. To drive a flow, we imposed an injection (source) and a production well (sink) in the lower-left and upper-right corners, respectively. The discretization of each layer by the quadrilateral RT0 finite elements yields 39 880 degrees of freedom.

---

<sup>1</sup> <http://www.sintef.no/Projectweb/GeoScale/Results/MsMFEM/SPE10/>

The layers were partitioned into subdomains in four ways: using two geometrically regular partitionings with the coarsening ratios  $H/h = 30$  and  $H/h = 10$ , and two irregular partitionings. The details of the partitionings are summarized in Table 1

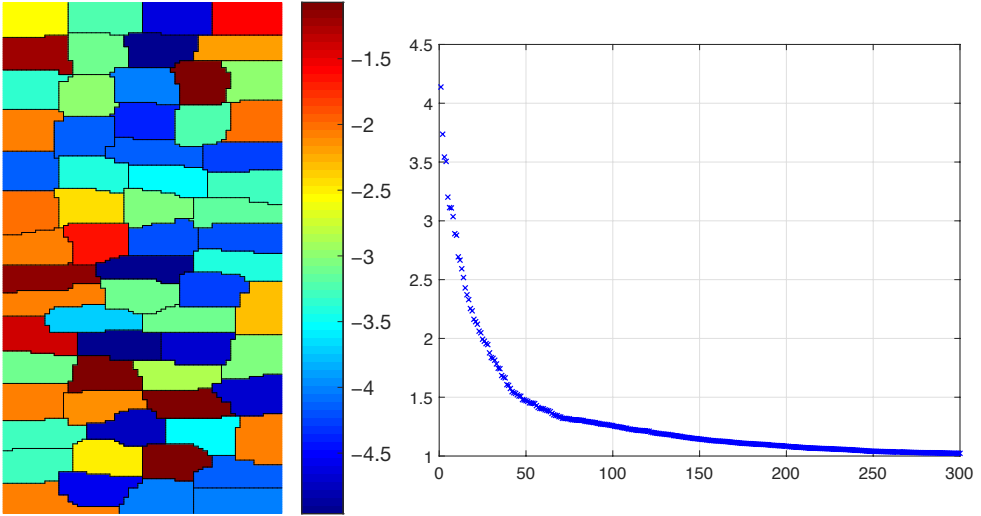


Figure 1. Substructuring and the base 10 logarithm of the permeability  $k$  for the problem with jumps aligned with the substructure interfaces (left panel) and the largest 300 eigenvalues of the BDDC preconditioned operator for this problem (right panel).

type of partitioning	$N$	$n_{\Gamma}$	$n_f$	$n_c$
2D				
regular ( $H/h = 30$ )	14	580	19	33
regular ( $H/h = 10$ )	132	2360	236	368
irregular A	16	756	29	70
irregular B	64	1746	152	315
3D				
regular ( $H/h = 10$ )	27	5400	54	81
irregular	32	7267	129	335

Table 1. Substructuring of the 2D and 3D problems:  $N$  is the number of subdomains,  $n_{\Gamma}$  is the number of (flux) degrees of freedom on interfaces,  $n_f$  is the number of faces, and  $n_c$  is the number of (initial) coarse degrees of freedom.

and illustrated by Figures 2–3. For the experiments in 3D, we used two domains consisting of  $30 \times 30 \times 30$  elements extracted from layers 1–30 and 56–85 of the SPE 10 problem shown in Figure 6. To drive a flow, we imposed an injection (source) and a production well (sink) in two distant corners of the domain. The discretization



by the hexahedral lowest-order Raviart-Thomas (RT0) finite elements yields 110 700 degrees of freedom. The domain was partitioned into subdomains in two ways: using one geometrically regular partitioning with the coarsening ratio  $H/h = 10$ , and an irregular partitioning. The details of the partitionings are summarized in Table 1 and illustrated by Figure 7. All irregular partitionings were obtained using METIS 4.0 [20], and in order to test the adaptive algorithm we did not take into account the permeability coefficients.

It is interesting to note that the adaptive flux coarse basis functions capture to some extent the features of the solution on the finite element mesh, and the quality of this approximation improves as the threshold  $\tau$  in Algorithm 4.2 decreases. We illustrate this fact by relative errors of solutions  $u_0$  and  $u^*$  obtained in Steps 1 and 2 of Algorithm 3.2 with respect to the exact solution  $u_{\text{exact}}$  obtained by a direct solve of the full problem. Specifically, the two relative errors are reported in tables as

$$(6.1) \quad \varepsilon_0 = \frac{\|u_0 - u_{\text{exact}}\|}{\|u_{\text{exact}}\|}, \quad \varepsilon^* = \frac{\|u^* - u_{\text{exact}}\|}{\|u_{\text{exact}}\|}.$$

We also compare the adaptive method with constraints inspired by *Multiscale mixed finite element method (MsMFEM)*, cf. [12], Algorithm 2.5.2 or [2], Section 3.2.1. In particular, instead of the local eigenvalue problems we solved local Darcy's flow problems, that is local counterparts of problem (2.1)–(2.2), with the source term

$$f(x) = \begin{cases} w_i & \text{for } x \in \Omega^i, \\ -w^j & \text{for } x \in \Omega^j, \end{cases}$$

and zero flux boundary condition on  $\partial\Omega^i \cap \partial\Omega^j$ . The source distribution function is set to  $w_i(x) = 1/|\Omega^i|$  in all subdomains except those containing a well, in which

$$w_i(x) = f(x) / \int_{\Omega^i} f(\xi) \, d\xi,$$

to ensure a conservative approximation on the fine grid. In the numerical experiments we then used the set of basic constraints (3.4) enriched by solving the above problem and taking the values of flux degrees of freedom on  $\partial\Omega^i \cap \partial\Omega^j$  as additional constraints. Nevertheless, we note that there are other more advanced solvers based on multiscale strategies available in the literature see, e.g. Yang et al. [49] or la Cour Christensen et al. [25], and a thorough comparison of the methods would be of independent interest.

The results of numerical experiments in 2D are summarized in Tables 2–4. Table 2 shows performance of the nonadaptive method for a homogeneous case with  $k = 1$  and the layers of the SPE 10 problem. It can be seen that for layers 1 and 20 the

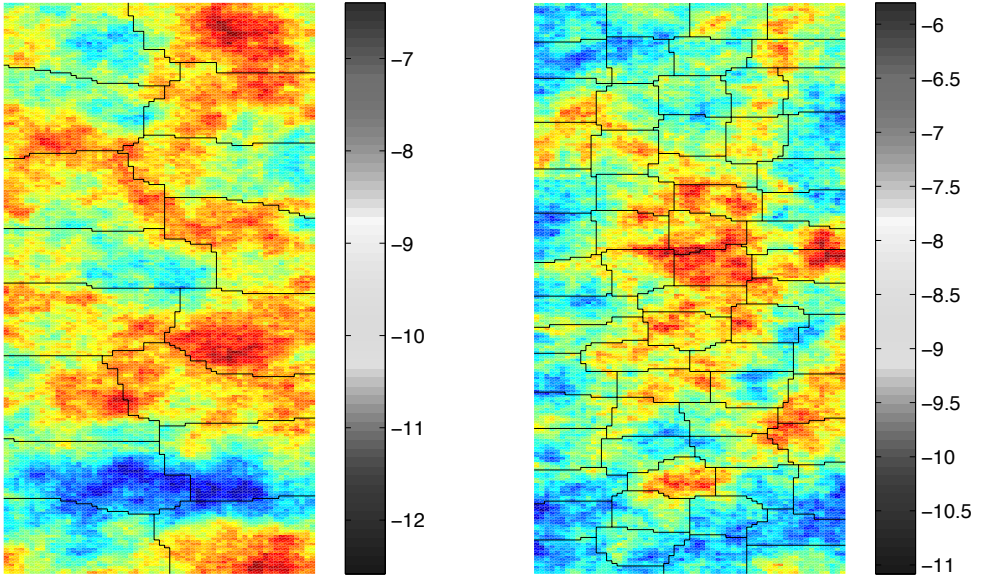


Figure 2. Substructuring and the base 10 logarithm of the permeability  $k$  in layer 1 (left panel) and layer 20 (right panel) of the SPE 10 problem. The left panel also illustrates irregular partitioning A, and the right panel illustrates irregular partitioning B.

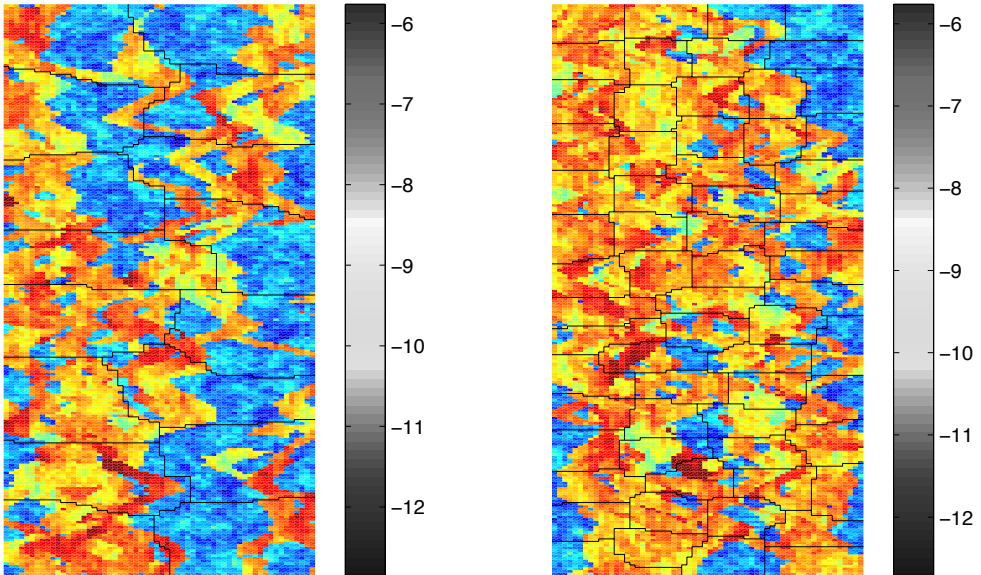


Figure 3. Substructuring and the base 10 logarithm of the permeability  $k$  in layer 60 (left panel) and layer 85 (right panel) of the SPE 10 problem. The left panel illustrates irregular partitioning A, and the right panel illustrates irregular partitioning B.

convergence does not significantly depend on the partitioning and it is also quite comparable to the homogeneous case with  $k = 1$ . On the other hand, for layers 60 and 85 the variations in coefficients aggravate convergence, which is also quite sensitive to the partitioning. This holds, in particular, for layer 60 which contains both regions that are highly heterogeneous and relatively homogeneous. It can be also seen by comparing the left and right columns in Table 2 that increasing the number of subdomains (that is decreasing the coarsening ratio  $H/h$ ) leads to higher condition numbers and an increase in iteration counts for both the regular and irregular partitionings. This is not the case in the standard theory of domain decomposition methods, but here we suspect it can be attributed to the jumps in coefficients and larger interfaces. The performance of the adaptive algorithm is illustrated by Tables 3–4. Table 3 shows convergence for layer 1 with irregular partitioning A, and Table 4 shows convergence for layer 85 with irregular partitioning B. It can be seen that in all cases lower values of the threshold  $\tau$  lead to fewer iterations, and the value of the condition number indicator  $\tilde{\omega} < \tau$  is in a good agreement with  $\kappa$ , which is the approximate condition number estimate obtained from the Lánczos sequence in conjugate gradients. The adaptive constraints also lead to more significant improvement in convergence than the multiscale constraints. The problem for layer 85 is particularly interesting. From the right panel in Figure 3 we see that the coefficient jumps have very large variations even on the interfaces, which can be seen in the left panel of Figure 4. The right panel displays the eigenvalues of the corresponding eigenproblem:  $\lambda_1 \approx 3769.5$  and all other eigenvalues are less than 20. Figure 5 then displays 300 largest eigenvalues of the (global) BDDC preconditioned

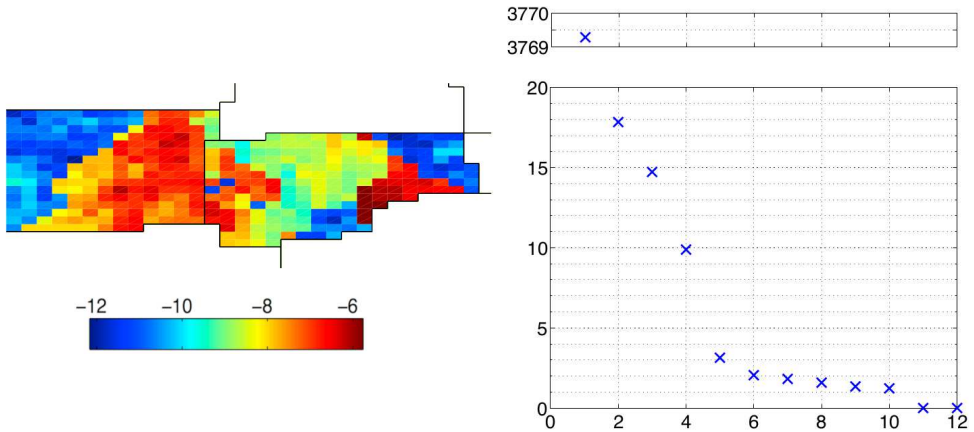


Figure 4. The base 10 logarithm of the permeability  $k$  in the subdomains 1 and 2 from the layer 85 of the SPE 10 problem (Fig. 3, right panel), and the first 12 eigenvalues of the corresponding local generalized eigenvalue problem (4.3). Here  $\lambda_i = 0$  for  $i = 11$  and 12.

layer	$H/h = 30$		$H/h = 10$		irregular A		irregular B	
	$it$	$\kappa$	$it$	$\kappa$	$it$	$\kappa$	$it$	$\kappa$
$(k = 1)$	11	2.790	14	3.980	12	3.151	14	4.050
1	15	8.879	22	9.491	17	6.714	19	11.197
20	14	5.749	19	6.926	15	6.524	18	6.429
60	162	4564.1	513	26359.3	244	11272.6	292	7301.7
85	183	9310.7	446	24492.8	208	7170.4	392	58931.7

Table 2. Convergence of the non-adaptive method for the homogeneous case ( $k = 1$ ) and the six layers of the SPE 10 problem. Here  $\varepsilon_0$  and  $\varepsilon^*$  are the errors (in %) defined by (6.1),  $\tilde{\omega}$  is the condition number indicator from (4.6),  $it$  is the number of iterations for relative residual tolerance  $10^{-6}$ , and  $\kappa$  is the approximate condition number computed from the Lánczos sequence in conjugate gradients.

$\tau$	$\varepsilon_0$ [%]	$\varepsilon^*$ [%]	$\tilde{\omega}$	$n_c$	$it$	$\kappa$
$\infty$	73.21	30.55	13.586	70	17	6.714
(ms)	72.55	27.15	-na-	121	15	5.998
10	71.86	29.11	8.404	73	16	6.231
5	70.77	18.89	4.765	81	13	5.517
3	69.19	11.64	2.997	104	11	2.842
2	69.23	9.42	1.970	153	8	1.915

Table 3. Convergence of the adaptive method for layer 1 of the SPE 10 problem with the irregular partitioning A. Here  $\tau$  is the target condition number from Algorithm 4.2,  $\varepsilon_0$  and  $\varepsilon^*$  are the errors (in %) defined by (6.1),  $\tilde{\omega}$  is the condition number indicator from (4.6),  $it$  is the number of iterations for relative residual tolerance  $10^{-6}$ , and  $\kappa$  is the approximate condition number computed from the Lánczos sequence in conjugate gradients. With  $\tau = \infty$  no adaptive constraints were used, and (ms) indicates use of the multiscale constraints.

$\tau$	$\varepsilon_0$ [%]	$\varepsilon^*$ [%]	$\tilde{\omega}$	$n_c$	$it$	$\kappa$
$\infty$	69.34	42.11	59491.702	315	392	58931.700
(ms)	68.76	39.00	-na-	494	297	8931.930
10000	69.34	42.11	9275.614	316	347	9170.830
1000	68.03	40.29	898.754	360	152	836.227
100	67.21	38.38	98.117	430	54	95.439
10	66.16	35.68	9.885	489	19	9.672
5	63.07	31.85	4.888	536	13	4.836
3	56.19	18.87	2.988	614	10	3.010
2	53.44	14.87	1.997	743	7	1.879

Table 4. Convergence of the adaptive method for layer 85 with the irregular partitioning B.

operator without adaptivity and with adaptive BDDC and target condition number  $\tau = 100$ . We see that without adaptivity there is a single largest eigenvalue: specifically  $\lambda_1 = 59.492$  and  $\lambda_2 = 9.258$ . For the adaptive BDDC with  $\tau = 100$  we get  $\lambda_1 = 96.3$ . Comparing this plot with Table 4 we see that the adaptive BDDC with  $\tau = 100$  introduces 115 adaptive constraints, which corresponds to the number of the largest eigenvalues removed from the spectrum of the BDDC preconditioned operator. We also note that adding a single adaptive constraint reduces the iteration count from 392 to 347, which corresponds to the large gap in the spectrum of the operator without adaptivity. Setting  $\tau$  to a lower value, for example,  $\tau = 3$ , roughly doubles the number of constraints and the number of iterations is reduced to approximately 10. Also, the lower value of  $\tau$  improves the approximation quality of the first two steps of Algorithm 3.2 and, for example, with  $\tau < 3$  we get the error  $\varepsilon^* < 20\%$ .

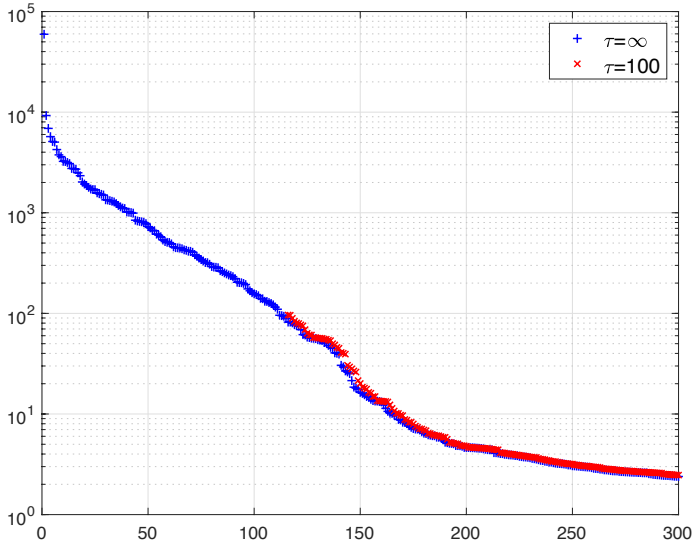


Figure 5. The largest 300 eigenvalues of the BDDC preconditioned operator for the layer 85 of the SPE 10 problem (Fig. 3, right panel) without adaptivity ( $\tau = \infty$ ) and for the adaptive BDDC with the target condition number  $\tau = 100$ .

The results of numerical experiments in 3D are summarized in Tables 5–7. It can be seen from Table 5 that the numbers of iterations are significantly higher than in 2D, and the convergence is slower for the fluvial bottom layers 56–85 comparing with the relatively smooth top layers 1–30. The increase in iterations becomes even more pronounced in the case of the irregular partitioning also due to larger interfaces. The results of experiments with the adaptive algorithm are summarized in Tables 6–7. As in the 2D case, lower values of the threshold  $\tau$  lead in all cases to fewer iterations,

layer	$H/h = 10$		irregular part.	
	$it$	$\kappa$	$it$	$\kappa$
$(k = 1)$	25	17.099	35	22.029
1–30	779	49 075.600	1968	$1.096 \times 10^6$
56–85	3762	$2.576 \times 10^6$	5277	$3.676 \times 10^6$

Table 5. Convergence of the non-adaptive method for the homogeneous case ( $k = 1$ ) and the two 3D cutouts of the SPE 10 problem from Figure 6. The headings are the same as in Table 2.

$\tau$	$\varepsilon_0$ [%]	$\varepsilon^*$ [%]	$\tilde{\omega}$	$n_c$	$it$	$\kappa$
$\infty$	98.41	86.05	$1.191 \times 10^6$	335	1968	$1.096 \times 10^6$
(ms)	98.29	86.05	-na-	571	1943	$1.079 \times 10^6$
100 000	98.39	85.95	94 328.862	349	1280	92 307.000
10 000	98.14	84.27	9862.559	514	514	10 512.200
1000	97.41	82.73	995.230	989	175	1014.150
100	92.93	72.63	97.989	1331	60	108.673
10	87.47	66.66	9.993	1617	18	11.711
5	85.82	65.46	4.985	1898	13	6.069
3	82.90	63.22	2.997	2331	9	3.007
2	81.62	62.81	2.000	2997	6	1.930

Table 6. Convergence of the adaptive method for layers 1–30 of the SPE 10 with irregular partitioning.

$\tau$	$\varepsilon_0$ [%]	$\varepsilon^*$ [%]	$\tilde{\omega}$	$n_c$	$it$	$\kappa$
$\infty$	99.31	66.24	$3.435 \times 10^6$	81	3762	$2.576 \times 10^6$
(ms)	99.30	66.24	-na-	99	3735	$2.566 \times 10^6$
100 000	99.33	65.78	95 129.959	122	1267	93 040.500
10 000	98.63	62.19	9834.429	188	498	9487.840
1000	98.25	63.81	990.920	373	183	1200.070
100	97.26	64.08	99.793	766	59	124.896
10	91.79	49.39	9.965	1154	17	9.506
5	88.88	43.77	4.991	1342	13	5.545
3	86.52	41.75	2.997	1610	9	3.205
2	85.18	41.05	2.000	1960	6	1.918

Table 7. Convergence of the adaptive method for layers 56–85 of the SPE 10 with the regular partitioning.

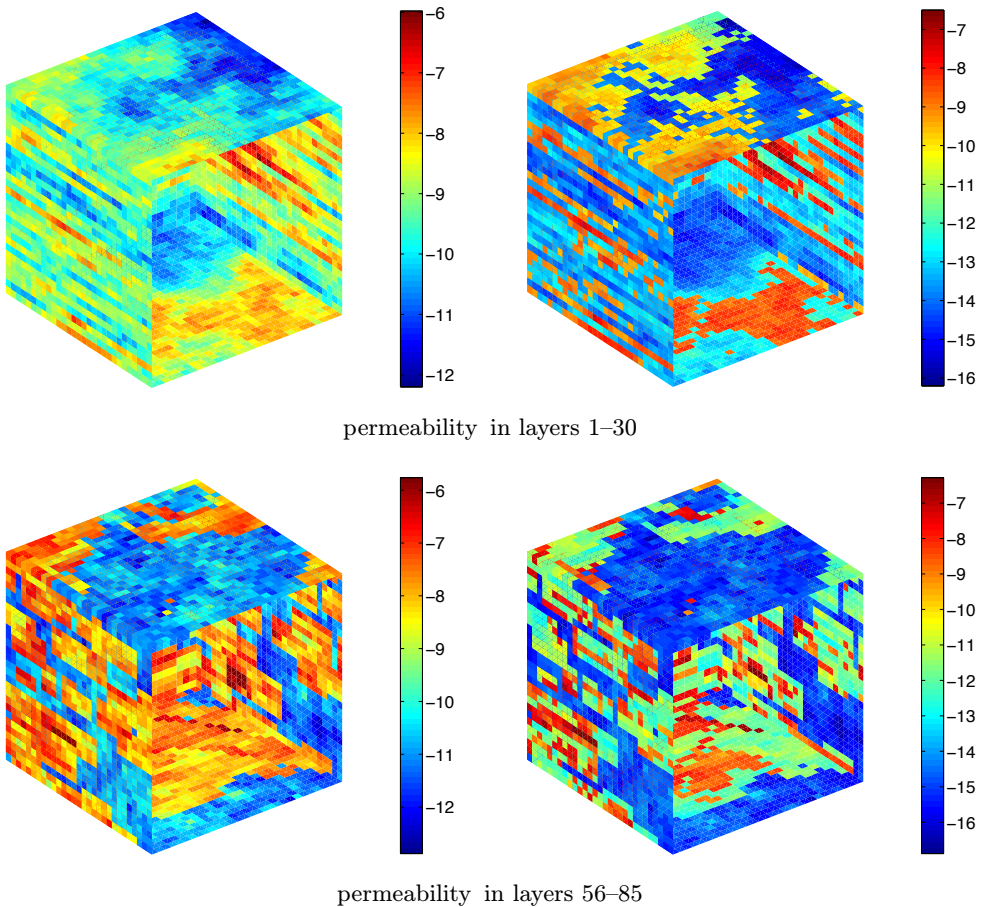


Figure 6. Base 10 logarithm of the permeability  $k$  in  $x$  and  $y$  directions (left), and in  $z$  direction (right) in two cutouts of the SPE 10 problem consisting of  $30 \times 30 \times 30$  elements.

and the values of  $\tau$ ,  $\tilde{\omega}$  and  $\kappa$  are in close agreement. Again, the multiscale constraints provide only a slight improvement of convergence. Table 6 shows convergence for layers 1–30. It can be seen that despite higher condition number of the problem corresponding to the irregular partitioning, the adaptive algorithm allows to decrease the iteration counts for lower values of  $\tau$ . As in 2D, the first few adaptive constraints allow to decrease the number of iterations by a fairly large amount: here adding 14 constraints reduces the number of iterations from the initial value 1968 to 1280. However, for example with  $\tau = 10$  the number of iterations decreases to 18, while the number of constraints grows rather significantly from 335 to 1617. Finally, the values of  $\varepsilon_0$  and  $\varepsilon^*$  are quite larger compared to the 2D experiments. Table 7

shows convergence for layers 56–85 and the regular partitioning  $H/h = 10$ , and the trends are quite similar to the previous case. That is, the adaptive algorithm allows to control the convergence of conjugate gradients, but the number of adaptive constraints is relatively high in particular for lower values of  $\tau$ . These trends are in agreement with the qualitative observations made from Figure 5.

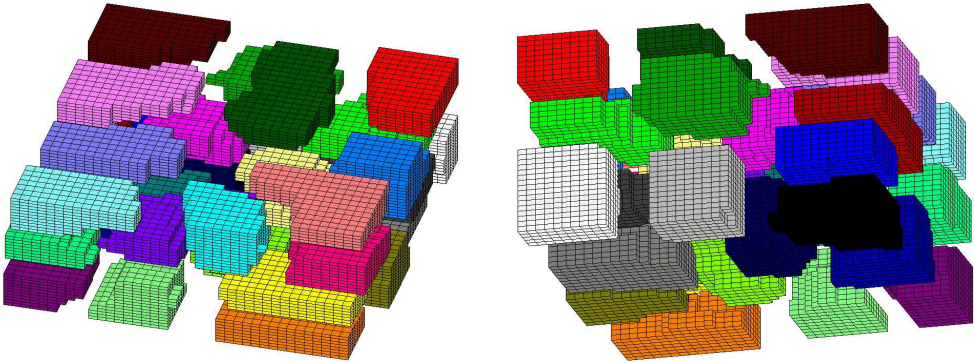


Figure 7. Irregular partitioning of the domain from Figure 6 into 32 subdomains.

## 7. CONCLUSION

We studied a method for solution of the single-phase flow in heterogeneous porous media. We have, in particular, shown that the idea of adaptive BDDC, previously used for elliptic problems, can be also applied in the context of the BDDC method for mixed finite element discretizations using the lowest-order Raviart-Thomas finite elements, and that the adaptive method works well with the usual types of scaling used in substructuring. We illustrated that the resulting algorithm can be successfully applied for adaptive selection of the coarse flux degrees of freedom using several examples corresponding to the SPE 10 benchmark model. The effect of the adaptive construction of the flux coarse basis functions is twofold. First, the first two steps of the BDDC method provide some approximation properties with respect to the exact solution of the full problem, in particular in 2D. Second, the coarse problem provides a better preconditioner for conjugate gradients used in the third step. We also compared the adaptive constraints with constraints inspired by multiscale mixed finite element method, and we found that the adaptive constraints outperform the multiscale constraints.

Next, we experimented with different partitioning of the domains into substructures. While the adaptive method is able to overcome these issues in many cases,



it is evident that a suitable partitioning makes the adaptive method more efficient. We note that the development of optimal partitioning strategies is an open problem, cf. e.g. [3], [48]. However, our experiments indicate that if it is not possible to find a suitable partitioning, the best strategy is to simply minimize the size of interfaces, which may be achieved by a simple geometric partitioning, see also [19].

**Acknowledgement.** I would like to thank the two anonymous referees and the editor for a number of useful comments that helped to improve the quality of the paper.

### References

- [1] *J. E. Aarnes, T. Gimse, K.-A. Lie*: An introduction to the numerics of flow in porous media using Matlab. Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF. Springer, Berlin, 2007, pp. 265–306. [zbl](#) [MR](#) [doi](#)
- [2] *J. E. Aarnes, S. Krogstad, K.-A. Lie*: A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.* *5* (2006), 337–363. [zbl](#) [MR](#) [doi](#)
- [3] *J. E. Aarnes, S. Krogstad, K.-A. Lie*: Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.* *12* (2008), 297–315. [zbl](#) [MR](#) [doi](#)
- [4] *F. Brezzi, M. Fortin*: Mixed and Hybrid Finite Element Methods. Springer Series in Computational Mathematics 15, Springer, New York, 1991. [zbl](#) [MR](#) [doi](#)
- [5] *M. A. Christie, M. J. Blunt*: Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.* *4* (2001), 308–317. [doi](#)
- [6] *L. C. Cowzar, J. Mandel, M. F. Wheeler*: Balancing domain decomposition for mixed finite elements. *Math. Comput.* *64* (1995), 989–1015. [zbl](#) [MR](#) [doi](#)
- [7] *J.-M. Cros*: A preconditioner for the Schur complement domain decomposition method. 14th Int. Conf. on Domain Decomposition Methods in Science and Engineering (I. Herrera et al., eds.). National Autonomous University of Mexico (UNAM), México, 2003, pp. 373–380. [zbl](#) [MR](#)
- [8] *J. W. Demmel*: Applied Numerical Linear Algebra. Society for Industrial and Applied Mathematics, Philadelphia, 1997. [zbl](#) [MR](#) [doi](#)
- [9] *C. R. Dohrmann*: A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.* *25* (2003), 246–258. [zbl](#) [MR](#) [doi](#)
- [10] *C. R. Dohrmann*: A substructuring preconditioner for nearly incompressible elasticity problems. Technical report SAND 2004–5393, Sandia National Laboratories, 2004.
- [11] *C. R. Dohrmann, O. B. Widlund*: Some recent tools and a BDDC algorithm for 3D problems in  $H(\text{curl})$ . Domain Decomposition Methods in Science and Engineering XX. Lecture Notes Computational Science and Engineering 91, Springer, Heidelberg, 2013, pp. 15–25. [zbl](#) [MR](#) [doi](#)
- [12] *Y. Efendiev, T. Y. Hou*: Multiscale Finite Element Methods: Theory and Applications. Surveys and Tutorials in the Applied Mathematical Sciences 4, Springer, New York, 2009. [zbl](#) [MR](#) [doi](#)
- [13] *R. E. Ewing, J. Wang*: Analysis of the Schwarz algorithm for mixed finite elements methods. *RAIRO, Modélisation Math. Anal. Numér.* *26* (1992), 739–756. [zbl](#) [MR](#) [doi](#)
- [14] *C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, D. Rixen*: FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Int. J. Numer. Methods Eng.* *50* (2001), 1523–1544. [zbl](#) [MR](#) [doi](#)

- [15] *C. Farhat, M. Lesoinne, K. Pierson*: A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.* *7* (2000), 687–714. [zbl](#) [MR](#) [doi](#)
- [16] *Y. Fragakis, M. Papadrakakis*: The mosaic of high performance domain decomposition methods for structural mechanics: formulation, interrelation and numerical efficiency of primal and dual methods. *Comput. Methods Appl. Mech. Eng.* *192* (2003), 3799–3830. [zbl](#) [doi](#)
- [17] *R. Glowinski, M. F. Wheeler*: Domain decomposition and mixed finite element methods for elliptic problems. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*. SIAM, Philadelphia, 1988, pp. 144–172. [zbl](#) [MR](#)
- [18] *G. H. Golub, C. F. Van Loan*: *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, 1996. [zbl](#) [MR](#)
- [19] *M. Hanek, J. Šístek, P. Burda*: The effect of irregular interfaces on the BDDC method for the Navier-Stokes equations. *Proc. Int. Conf. Domain Decomposition Methods in Science and Engineering XXIII. Lecture Notes Computational Science and Engineering* *116*, Springer, Cham, 2017, pp. 171–178. [zbl](#) [MR](#) [doi](#)
- [20] *G. Karypis, V. Kumar*: METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 4.0. Technical report, Department of Computer Science, University of Minnesota, 1998. [zbl](#) [MR](#) [doi](#)
- [21] *A. Klawonn, M. Kühn, O. Rheinbach*: Adaptive coarse spaces for FETI-DP in three dimensions. *SIAM J. Sci. Comput.* *38* (2016), A2880–A2911. [zbl](#) [MR](#) [doi](#)
- [22] *A. Klawonn, M. Kühn, O. Rheinbach*: A closer look at local eigenvalue solvers for adaptive FETI-DP and BDDC. Technical report, Universität zu Köln, 2018. Available at <https://kups.ub.uni-koeln.de/9020/>.
- [23] *A. Klawonn, O. Rheinbach, O. B. Widlund*: An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.* *46* (2008), 2484–2504. [zbl](#) [MR](#) [doi](#)
- [24] *A. V. Knyazev*: Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.* *23* (2001), 517–541. [zbl](#) [MR](#) [doi](#)
- [25] *M. la Cour Christensen, U. Villa, A. P. Engsig-Karup, P. S. Vassilevski*: Numerical multilevel upscaling for incompressible flow in reservoir simulation: an element-based algebraic multigrid (AMGe) approach. *SIAM J. Sci. Comput.* *39* (2017), B102–B137. [zbl](#) [MR](#) [doi](#)
- [26] *J. Li, X. Tu*: Convergence analysis of a balancing domain decomposition method for solving a class of indefinite linear systems. *Numer. Linear Algebra Appl.* *16* (2009), 745–773. [zbl](#) [MR](#) [doi](#)
- [27] *J. Li, O. B. Widlund*: BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.* *44* (2006), 2432–2455. [zbl](#) [MR](#) [doi](#)
- [28] *J. Li, O. B. Widlund*: FETI-DP, BDDC, and block Cholesky methods. *Int. J. Numer. Methods Eng.* *66* (2006), 250–271. [zbl](#) [MR](#) [doi](#)
- [29] *J. Mandel, B. Sousedik*: Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Eng.* *196* (2007), 1389–1399. [zbl](#) [MR](#) [doi](#)
- [30] *J. Mandel, B. Sousedik, C. R. Dohrmann*: Multispace and multilevel BDDC. *Computing* *83* (2008), 55–85. [zbl](#) [MR](#) [doi](#)
- [31] *J. Mandel, B. Sousedik, J. Šístek*: Adaptive BDDC in three dimensions. *Math. Comput. Simul.* *82* (2012), 1812–1831. [zbl](#) [MR](#) [doi](#)
- [32] *T. P. Mathew*: Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems. I. Algorithms and numerical results. *Numer. Math.* *65* (1993), 445–468. [zbl](#) [MR](#) [doi](#)
- [33] *D.-S. Oh, O. B. Widlund, S. Zampini, C. R. Dohrmann*: BDDC algorithms with deluxe scaling and adaptive selection of primal constraints for Raviart-Thomas vector fields. *Math. Comput.* *87* (2018), 659–692. [zbl](#) [MR](#) [doi](#)

- [34] *C. Pechstein*: Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems. Lecture Notes in Computational Science and Engineering 90, Springer, Berlin, 2013. [zbl](#) [MR](#) [doi](#)
- [35] *C. Pechstein, C. R. Dohrmann*: A unified framework for adaptive BDDC. ETNA, Electron. Trans. Numer. Anal. *46* (2017), 273–336. [zbl](#) [MR](#)
- [36] *C. Pechstein, R. Scheichl*: Analysis of FETI methods for multiscale PDEs. Part II: interface variation. Numer. Math. *118* (2011), 485–529. [zbl](#) [MR](#) [doi](#)
- [37] *J. Šístek, J. Březina, B. Sousedík*: BDDC for mixed-hybrid formulation of flow in porous media with combined mesh dimensions. Numer. Linear Algebra Appl. *22* (2015), 903–929. [zbl](#) [MR](#) [doi](#)
- [38] *B. Sousedík*: Nested BDDC for a saddle-point problem. Numer. Math. *125* (2013), 761–783. [zbl](#) [MR](#) [doi](#)
- [39] *B. Sousedík, J. Šístek, J. Mandel*: Adaptive-multilevel BDDC and its parallel implementation. Computing *95* (2013), 1087–1119. [zbl](#) [MR](#) [doi](#)
- [40] *N. Spillane, D. J. Rixen*: Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. Int. J. Numer. Methods Eng. *95* (2013), 953–990. [zbl](#) [MR](#) [doi](#)
- [41] *A. Toselli, O. Widlund*: Domain Decomposition Methods—Algorithms and Theory. Springer Series in Computational Mathematics 34, Springer, Berlin, 2005. [zbl](#) [MR](#) [doi](#)
- [42] *X. Tu*: A BDDC algorithm for a mixed formulation of flow in porous media. ETNA, Electron. Trans. Numer. Anal. *20* (2005), 164–179. [zbl](#) [MR](#)
- [43] *X. Tu*: A BDDC algorithm for flow in porous media with a hybrid finite element discretization. ETNA, Electron. Trans. Numer. Anal. *26* (2007), 146–160. [zbl](#) [MR](#)
- [44] *X. Tu*: Three-level BDDC in three dimensions. SIAM J. Sci. Comput. *29* (2007), 1759–1780. [zbl](#) [MR](#) [doi](#)
- [45] *X. Tu*: Three-level BDDC in two dimensions. Int. J. Numer. Methods Eng. *69* (2007), 33–59. [zbl](#) [MR](#) [doi](#)
- [46] *X. Tu*: A three-level BDDC algorithm for a saddle point problem. Numer. Math. *119* (2011), 189–217. [zbl](#) [MR](#) [doi](#)
- [47] *X. Tu, J. Li*: A balancing domain decomposition method by constraints for advection-diffusion problems. Commun. Appl. Math. Comput. Sci. *3* (2008), 25–60. [zbl](#) [MR](#) [doi](#)
- [48] *E. Večharynski, Y. Saad, M. Sosonkina*: Graph partitioning using matrix values for preconditioning symmetric positive definite systems. SIAM J. Sci. Comput. *36* (2014), A63–A87. [zbl](#) [MR](#) [doi](#)
- [49] *Y. Yang, S. Fu, E. T. Chung*: A two-grid preconditioner with an adaptive coarse space for flow simulations in highly heterogeneous media. Available at <https://arxiv.org/abs/1807.07220> (2018), 17 pages.
- [50] *S. Zampini, X. Tu*: Multilevel balancing domain decomposition by constraints deluxe algorithms with adaptive coarse spaces for flow in porous media. SIAM J. Sci. Comput. *39* (2017), A1389–A1415. [zbl](#) [MR](#) [doi](#)

*Author's address: Bedřich Sousedík, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA, e-mail: sousedik@umbc.edu.*