ADAPTIVE – MULTILEVEL BDDC

by

Bedřich Sousedík

M. Eng., Czech Technical University in Prague, 2001

M.S., University of Colorado Denver, 2008

Ph.D., Czech Technical University in Prague, 2008

A thesis submitted to the

University of Colorado Denver

in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Applied Mathematics

2010

This thesis for the Doctor of Philosophy

degree by

Bedřich Sousedík

has been approved

by

Jan Mandel

Lynn S. Bennethum

Pavel Burda

Julien Langou

Samuel W. J. Welch

Date

Sousedík, Bedřich (Ph.D., Applied Mathematics)

Adaptive – Multilevel BDDC

Thesis directed by Professor Jan Mandel

ABSTRACT

The Balancing Domain Decomposition by Constraints (BDDC) method by Dohrmann (2003) is the most advanced method from the Balancing family of iterative substructuring methods by Mandel (1993) for the solution of large systems of linear algebraic equations arising from discretizations of elliptic boundary value problems. The method is closely related to FETI-DP by Farhat et. al. (2001), and it is the same as other two methods proposed independently by Fragakis and Papadrakakis (2003) and by Cros (2003).

Because these are two-level methods, solving the coarse problem exactly becomes a bottleneck when the number of substructures becomes large. The coarse problem in BDDC has the same structure as the original problem, so it is straightforward to apply the BDDC method recursively to solve the coarse problem only approximately. In the first part we formulate a new family of abstract Multispace BDDC methods and give a condition number bound from the abstract additive Schwarz preconditioning theory. The Multilevel BDDC is then treated as a special case of the Multispace BDDC, and it is also shown that the original, two-level, BDDC can be written as a multispace method.

In the second part we propose a method for adaptive selection of the coarse space for the original two-level BDDC method. The method works by adding coarse degrees of freedom constructed from eigenvectors associated with intersections of selected pairs of adjacent substructures. It is assumed that the starting coarse degrees of freedom are already sufficient to prevent relative rigid body motions in any selected pair of adjacent substructures. A heuristic indicator of the condition number is developed and a minimal number of coarse degrees of freedom is added to decrease the indicator under a given threshold.

In the third part we combine the advantages of both approaches to propose a new method called *Adaptive – Multilevel BDDC* that preserves both parallel scalability with increasing number of subdomains and very good convergence properties. Performance of the method is illustrated by several numerical examples in two and three spatial dimensions.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed _____
Jan Mandel

DEDICATION

Dedicated to Alyssa Heberton-Morimoto.

ACKNOWLEDGMENT

I would like to give my deepest thank to my advisor Professor Jan Mandel. He was always a good source of interesting new problems, and let me work on topics that I liked. Without his guidance and mentorship this thesis would have been impossible. A large debt of gratitude goes to the author of the BDDC method Dr. Clark R. Dohrmann for bringing our attention to its multilevel extension, and to Professors Marian Brezina, Jaroslav Kruis, Jaroslav Novotný and Jakub Šístek who provided the data for numerical experiments and helped me with visualization of finite element meshes. In particular, I have been truly impressed by Marian's own visualization package *MyVis*. I am grateful to Prof. Pavel Burda and Dr. Chris Harder for proof-reading of a draft of this thesis.

I would also like to thank the faculty and my fellow students at the University of Colorado Denver for creating excellent working conditions. Last but not least I thank my parents for bringing me up the way they did and for their continuing support. Finally, I thank my family and friends for support and distraction from science.

The doctoral dissertation has been kindly supported by the National Science Foundation under grants CNS-0325314, CNS-0719641, and DMS-0713876. The research completed during my visits at the Institute of Thermomechanics, Academy of Sciences of the Czech Republic has been supported by the Grant Agency of the Czech Republic GA ČR 106/08/0403.

CONTENTS

FIGURES

Figure

TABLES

1. Introduction

A dramatic increase in availability of multicore, *parallel computers* over the last few decades has motivated a progressive research in many areas, including *domain decomposition* methods. Out of many possible interpretations of the domain decomposition cf., e.g., monographs [74, 77] it will be in our context understood as a separation of a physical domain into *subdomains*, called alternatively (with the same meaning) *substructures*. These methods then aim to allow for an efficient and scalable solution of systems of linear equations arising from discretizations of partial differential equations (PDEs), in particular by finite element methods. In this thesis we will restrict our attention to *elliptic problems* such as Laplace equation or problems of linear elasticity.

The important class of domain decomposition methods that we will focus on is known as *iterative substructuring*. From the point of view of linear algebra, the solution of a large problem is replaced by repeated solutions of a number of independent subproblems, to be solved in parallel. The algorithms are formulated as preconditioned Krylov iterative methods such as conjugate gradients (CG), or GMRES. All these methods access the specific problem through only two subroutines, namely the system matrix-vector multiplication, and preconditioner matrix-vector multiplication. There is often a pre-processing step consisting of a transformation of variables before an iterative method is applied; in that case, subroutines to pre-process the right-hand side and to recover the solution also need to be provided.

The following method components form the system and the preconditioner presented to the iterative method:

- The solution space or, as we will see later, some larger space is split into a number of subspaces, and independent problems to be solved in parallel are set up in each one of them, and solved in each iteration.

- Correspondence between the subspaces and the solution space is established to build a global solution from the local solutions.

- A global coarse problem with one or as few as possible variables per subspace is set up and then solved in each iteration to coordinate the solution between all subspaces at once.

Careful splitting into subspaces and preconditioning in the subspaces are needed for scalability with subdomain size. A coarse problem is necessary for scalability with increasing number of subdomains, and hence, processors. We now briefly review some domain decomposition methods important in the context of the BDDC method – the main topic of this thesis. The comprehensive overview of the field has been the subject of numerous monographs, e.g., [47, 74, 77, 81, 87, 91], cf., also the standard finite element monograph [5]. An important role in the present development is also played by another thesis of the author [79], where some of the most popular methods were formulated and compared under so-called minimalist assumptions. This revealed that some of the methods are closely related, equivalent or even the same. The theory has been used in the final chapter to develop an adaptive method that significantly improves convergence of two most advanced methods, BDDC and FETI-DP.

We note that the basic idea of the adaptive method is the same in the two theses. However, the implementation in [79] uses global spaces, projections, and a change of variables to preserve sparsity of global operators, whereas the formulation presented here builds explicitly all subspaces (including the coarse).

Perhaps the earliest *primal domain decomposition methods* were based on the concept of *substructuring*. The domain is divided into non-overlapping substructures, and there is one subspace formed by the degrees of freedom within a substructure. The subspaces overlap in the degrees of freedom on substructure interfaces. The internal degrees of freedom in each substructure are then eliminated; this alone results in a significant decrease of the condition number, typically from $O(h^{-2})$ to $O(h^{-1})$. So even simple diagonal preconditioners and no coarse problem result in good practical parallel methods [35]. The system matrix-vector multiplication is implemented by solving an independent Dirichlet problem in each substructure, which is the same as multiplication by the Schur complement in each substructure. More sophisticated diagonal preconditioners need to know the diagonal entries of the Schur complements, which are not available, but can be estimated from the matrix-vector multiplication [11]. Asymptotically optimal (for small h) diagonal preconditioning of the problem reduced to substructure interfaces, along with various coarse problems, were introduced in pioneering theoretically oriented papers [2, 3, 19, 89]. The coarse problems are constructed from piecewise linear or piecewise constant functions on substructure scale. These methods have condition numbers that, for geometrically regular problems, provably do not grow faster than $log^2 \frac{H}{h}$, where H is the typical substructure size and h is the mesh step or element size.

Methods preconditioning the system reduced to interfaces by a weighted sum of inverses of the reduced substructure matrices [14, 15, 30] have become known as *Neumann-Neumann* methods [20], because multiplication of a vector by the inverse is the same as numerically solving the differential equations on both sides of a substructure interface with the Neumann boundary conditions. A scalable version of the Neumann-Neumann method requires a suitable coarse problem. The *Balancing Domain Decomposition* (BDD) by Mandel [53] constructs the coarse problem from the natural nullspace of the problem (constants for scalar problems, rigid body modes for elasticity). The BDD has become quite popular, because it can be formulated purely algebraically, requiring only substructure matrix-vector multiplications and the substructure nullspace vectors. For example, in application of BDD to mixed discretization of a flow problem [12], the matrix-vector multiplications are provided by pressure to flux and flux to pressure mappings on substructure interfaces. The BDD method and its theory were further developed for problems with coefficient jumps [55, 57], and also for plates and shells [48, 49].

Currently the most advanced version of the primal substructuring method is the *Balancing Domain Decomposition by Constraints* (BDDC) by Dohrmann [17]. It uses (similarly to [49]) a coarse space defined by energy minimization on each substructure separately, subject to given values of coarse degrees of freedom (such as values at substructure corners or edge or face averages), which act as constraints, but the BDDC method uses a careful additive approach: the substructure spaces and the coarse space form an energy orthogonal decomposition of a space of vectors that are not continuous across substructure interfaces;

the only approximation comes from enforcing the continuity across substructure interfaces by averaging. This results in lower condition numbers and faster convergence than other methods. The first proofs that BDDC is asymptotically optimal (the condition number does not grow faster than $log^2 \frac{H}{h}$) were given by Mandel, Dohrmann and Tezaur in [58, 59]. Unlike other primal substructuring methods, BDDC requires only the substructure matrices.

The *dual substructuring methods* enforce the continuity of the solution on substructure interfaces by Lagrange multipliers; for the case of 2 substructures, see [30]. In general, some substructure matrices will have a nontrivial nullspace, and guaranteeing that all local problems have right-hand sides orthogonal to the nullspace gives the original *Finite Element Tearing and Interconnecting* (FETI) method by Farhat and Roux [26], later called FETI-1. Farhat, Mandel and Roux have shown that resolving the nullspace acts as a natural coarse problem [25], but the original FETI method still had only a diagonal preconditioner, with resulting lack of scalability. An asymptotically optimal preconditioner involving a Dirichlet problem in each substructure and the first proof of the polylogarithmic growth of the condition number were given in [66]. Further developments of the FETI method include more complicated coarse spaces for plates and shells [24, 22], which also proved to have polylogarithmic bound on the condition number growth [68]. Currently the most advanced method is *Finite Element Tearing and Interconnecting - Dual, Primal* (FETI-DP) by Farhat et al. [23, 21]. The method enforces continuity of the values of the solution at substructure corners and, in 3D, of the values of face averages, cf. [23, 44], by choosing those values as (primal) coarse degrees of freedom, and enforcing

the equality of the rest of the degrees of freedom on substructure interfaces by Lagrange multipliers. All original and coarse degrees of freedom are eliminated (this involves setting up a coarse problem and solving it in every iteration) and the remaining dual system in terms of the Lagrange multipliers is solved iteratively. Polylogarithmic condition number bounds for FETI-DP were first proved by Mandel and Tezaur [67] and generalized to the case of coefficient jumps between substructures by Klawonn, Widlund and Dryja [43].

The FETI method is conceptually dual to the BDD. In the FETI method, evaluation of the system matrix-vector multiplication involves solving a Neumann problem on each substructure, and the preconditioner involves solving a Dirichlet problem on each substructure. In the BDD, the role of the Dirichlet and the Neumann problems is reversed. The rest of the algebra is somewhat different so the base methods are not dual in any exact sense. However, algebraic relations between the mesh transfer operators in primal and dual methods were proved by Rixen at al. [75] and Klawonn and Widlund [41] have reduced the asymptotic analysis of BDD and FETI to essentially a single inequality. Fragakis and Papadrakakis [29] then classified common components of BDD and FETI type methods and observed numerically that the preconditioned operator in one version [1] of the FETI method, with certain improvements to make it more robust, has the same eigenvalues as the BDD method, and proposed several primal methods derived from FETI class methods. Proof that the eigenvalues of BDDC and FETI-DP are identical except for eigenvalue equal to one was given by Mandel, Dohrmann and Tezaur [59]. These proofs were followed by a surge of important simplified proofs, alternative frameworks, and extensions

to various problems by the best analysts in the substructuring field and their students and coworkers, e.g., [4, 6, 7] and [28, 40, 43, 52, 51, 69]. The most popular methods were also analyzed by Mandel and Sousedík under so-called *minimalist assumptions* [62, 79, 80]. This setting allowed to show that in the case of corner constraints only, methods identical to BDDC were independently derived as primal versions of FETI-DP by Cros [13] and by Fragakis and Papadrakakis [29]. The BDDC and, equivalently, FETI-DP are quite robust. It can be proved that the condition number remains bounded even for large classes of subdomains with rough interfaces in 2D [39, 90] as well as in many cases of strong discontinuities of coefficients, including some configurations when the discontinuities cross substructure boundaries [70, 71].

Solving the coarse problem exactly in the original BDDC method becomes a bottleneck as the number of unknowns and, in particular, the number of substructures gets too large. Since the coarse problem in BDDC, unlike in the FETI-DP, has the same structure as the original problem, it is straightforward to apply the method recursively to solve the coarse problem only approximately [17]. The original, two-level, BDDC has been extended into three-levels by Tu [83, 82, 85, 84] and into a general multilevel method by Mandel, Sousedík and Dohrmann [63, 64]. However, the abstract condition number bounds reveal deteriorating convergence with increasing number of levels. We also note that recently Tu extended BDDC into three-level methods for saddle point problems [86], and with Kim for mortar discretizations [36]. Another research direction includes applications of inexact substructure solvers in the BDDC methods [18, 52], and inexact coarse problem solvers in the FETI methods [33, 37, 38].

Moreover, despite their robustness, the condition number of the BDDC and, equivalently, FETI-DP does deteriorate in many situations of practical importance and an adaptive method is warranted. Enriching the coarse space so that the iterations run in a subspace devoid of "difficult" modes has been a long-standing trick in iterative substructuring methods and used, e.g., in the development of BDD and FETI for plates from the base BDD and FETI methods [24, 48, 49, 68]. Methods that build a coarse space adaptively from local eigenvalue calculations were also devised in a number of other contexts [8, 27, 54, 56, 73]. Adaptive enrichment for BDDC and FETI-DP was proposed by Mandel and Sousedík in [60, 61], with the added coarse functions built from eigenproblems based on adjacent pairs of substructures in 2D. However, the adaptive method was formulated in terms of FETI-DP operators and it was quite complicated. Later, the algorithm has been developed directly in terms of BDDC operators and extended to 3D by Mandel, Sousedík and Šístek [65, 79], resulting in a much simplified formulation and implementation. This implementation framework operates on global matrices, builds no explicit coarse problem, and gets much of its parallelism through the direct solver used to solve an auxiliary decoupled system. To preserve sparsity, the authors used a variant of the change of variables from [51], extended to an arbitrary number of constraints. It has been shown on numerical examples that the heuristic eigenvalue-based estimates work reasonably well and that the adaptive approach can result in the concentration of computational work in a small troublesome part of the problem, which leads to a good convergence behavior at a small added cost. The only requirement for the adaptive algorithm is that there is a sufficient num-

ber of corner constraints to prevent rigid body motions between any pair of adjacent substructures. This requirement has been recognized recently in other contexts [9, 50], and in the context of BDDC by Burda et al. [10].

The main goal of this thesis is to combine the adaptive and multilevel approaches to the BDDC method in order to develop its variant that would preserve parallel scalability with an increasing number of subdomains and also show excellent convergence properties. Some of the material has already been published. This thesis is organized as follows. In Chapter 2 we establish the notation and introduce problem settings and preliminaries. Following [63, 64], in Chapter 3 we formulate the Abstract Multispace BDDC. In Chapter 4 we then present the Multilevel BDDC as a particular instance of the Multispace BDDC. We also derive an abstract condition number bound. Chapter 5 is based on a series of papers by Mandel, Sousedík and Šístek [61, 65, 79, 78], except for Section 5.1 which is new. We describe the adaptive selection of constraints in terms of BDDC operators only ([65, 79]). However, the presented formulation includes an explicit coarse space correction, outlined for 2D in [78], which allows for a multilevel extension of the adaptive algorithm. Chapter 6 is new and contains the main result of this thesis. We combine the adaptive and multilevel approaches and formulate the new method called *Adaptive – Multilevel BDDC* which allows for the adaptive selection of constraints on each decomposition level. Numerical experiments are presented in Chapter 7. It appears that the adaptive algorithm is able to effectively detect troublesome parts on each decomposition level and decrease the number of iterations at a small added cost. Finally, Chapter 8 contains the summary and concluding remarks.

The Multispace and Multilevel BDDC (Chapters 3–4) resulted from a joint research with Jan Mandel and Clark R. Dohrmann. The first idea of the adaptive algorithm (Chapter 5) can be traced back to a joint paper with Jan Mandel [61]. Its extension into 3D was originally published in the author's first thesis [79]. However, this work contains the first publication of this formulation with an explicit coarse space, along with a preconditioner for LOBPCG (Section 5.1). The main result of the thesis, the Adaptive–Multilevel BDDC algorithm is also an original contribution of the author; and only a 2D outline of the algorithm has been recently submitted in the conference proceedings [78]. The 3D version of the algorithm appears here for the first time.

2. Problem Setting and Preliminaries

We wish to solve an abstract linear problem

$$u \in X : a(u, v) = \langle f_X, v \rangle , \quad \forall v \in X, \tag{2.1}$$

where X is a finite dimensional linear space, $a(\cdot, \cdot)$ is a symmetric positive definite bilinear form defined on X, $f_X \in X'$ is the right-hand side with X' denoting the dual space of X, and $\langle \cdot, \cdot \rangle$ is the duality pairing. The form $a(\cdot, \cdot)$ is also called the energy inner product, the value of the quadratic form $a(u, u)$ is called the energy of u, and the norm $\|u\|_a = [a(u, u)]^{1/2}$ is called the energy norm. The operator $A_X : X \mapsto X'$ associated with a is defined by

$$a(u, v) = \langle A_X u, v \rangle , \quad \forall u, v \in X,$$

and (2.1) can be equivalently written as a system of linear algebraic equations

$$A_X u = f_X,$$

which we would like to solve by a preconditioned conjugate gradient method.

Here, a preconditioner is a mapping $B : X' \to X$ and we will look for preconditioners such that $\langle r, Br \rangle$ is also symmetric and positive definite on X'. In iteration k the method computes the residual

$$r^{(k)} = A_X u^{(k)} - f_X \in X',$$

and the preconditioner computes the increment to the approximate solution $u^{(k)}$ as a linear combination of the preconditioned residual $Br^{(k)} \in X$ with preconditioned residuals in earlier iterations.

It is well known that $BA_X : X \to X$ has only real positive eigenvalues, and convergence of the preconditioned conjugate gradients method can be established from the eigenvalues λ of the *preconditioned operator* BA_X; the condition number

$$\kappa = \frac{\lambda_{\max}(BA_X)}{\lambda_{\min}(BA_X)},$$

gives a well-known bound on the error reduction, cf. e.g. [31],

$$\left\|e^{(k)}\right\|_K \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \left\|e^{(0)}\right\|_K,$$

where $e^{(k)} = u^{(k)} - u$ is the error of the solution in iteration k.

2.1 Additive Schwarz preconditioners

Because of their importance in the forthcoming development, we briefly review the theory of abstract additive Schwarz preconditioners. Such a preconditioner is specified by a decomposition of the solution space X into subspaces,

$$X = X_1 + ... + X_M, \tag{2.2}$$

and by symmetric positive definite bilinear forms b_i on X_i. The preconditioner is a linear operator

$$B : X' \to X, \qquad B : r \mapsto u,$$

defined by solving the following variational problems on the subspaces and adding the results:

$$B : r \mapsto u = \sum_{k=1}^{M} u_k, \quad u_k \in X_k : \quad b_k(u_k, y_k) = \langle r, y_k \rangle, \quad \forall y_k \in X_k. \tag{2.3}$$

Theorem 2.1 (Dryja and Widlund [20]) *If there exist constants C_0, ω, and a symmetric matrix $E = (e_{ij})_{i,j=1}^{M}$ such that*

$$\forall u \in X \ \exists u_k \in X_k, k = 1, \ldots, M : u = \sum_{k=1}^{M} u_k, \ \ \sum_{k=1}^{M} \|u_k\|_{b_k}^2 \leq C_0 \|u\|_a^2 \tag{2.4}$$

$$\forall k = 1, \ldots, M \ \forall u_k \in X_k : \|u_k\|_a^2 \leq \omega \|u_k\|_{b_k}^2 \tag{2.5}$$

$$\forall u_k \in X_k, k = 1, \ldots, M : a(u_i, u_j) \leq e_{ij} \|u_i\|_a \|u_j\|_a \tag{2.6}$$

then

$$\kappa = \frac{\lambda_{\max}(BA_X)}{\lambda_{\min}(BA_X)} \leq C_0 \omega \rho(E).$$

This theorem with proof can be found in Dryja and Widlund [20, Theorem 1], or in the monograph by Toselli and Widlund [81, Theorem 2.7].

Remark 2.2 *Because $\rho(E) \leq \|E\|_\infty$ and we can choose $e_{ij} = 1$ if $X_i \cap X_j \neq \{0\}$, $e_{ij} = 0$ otherwise, we have the easy estimate*

$$\rho(E) \leq \max_{i=1,\ldots,M} |\{j \in \{1, \ldots, M\} : X_i \cap X_j \neq \{0\}\}| \leq M. \tag{2.7}$$

In the case when $M = 1$, the preconditioner simplifies to

$$r \longmapsto u \in X : \quad b(u, v) = \langle r, v \rangle, \quad \forall v \in X,$$

and Theorem 2.1 reduces to the statement that if

$$\frac{1}{\omega} \|u\|_a^2 \leq \|u\|_b^2 \leq C_0 \|u\|_a^2 \quad \forall u \in X, \tag{2.8}$$

then $\kappa \leq \omega C_0$.

2.2 Abstract variational BDDC preconditioner

Next, we present the original BDDC preconditioner in a particularly simple form originally formulated in [61] which has been inspired by a view of the Neumann-Neumann methods, going back once more to Dryja and Widlund [20].

Suppose that the bilinear form a is defined and symmetric positive semidefinite on a larger space $W \supset X$. The abstract version of the original BDDC preconditioner [17, 58] is characterized by a selection of an intermediate space V,

$$X \subset V \subset W. \tag{2.9}$$

Assumption 2.3 *The form $a(\cdot,\cdot)$ is positive definite on V.*

Algorithm 2.4 (Abstract BDDC) *Given the space V and a projection Q, such that $a(\cdot,\cdot)$ is positive definite on V, and*

$$X \subset V, \quad Q : V \to X,$$

define the preconditioner $B : r \in X' \longmapsto u \in X$ by

$$B : r \longmapsto u = Qv, \quad v \in V : \quad a(v,z) = \langle r, Qz \rangle, \quad \forall z \in V. \tag{2.10}$$

This formulation is remarkably simple, involving only the bilinear form a, the space V, and the projection Q. We note that in typical substructuring applications the spaces W and X are given, and the space V and the projection Q are to be chosen.

At this point we would like to premise that the main goal of this dissertation is to address several different ways of selecting V and Q. But before proceeding any further, let us recall the abstract condition number bound of the original two-level BDDC method, cf. also [61, Theorem 2].

Theorem 2.5 *Let Q be a projection onto X, let us define a space $V^{\mathcal{M}}$ by*

$$V^{\mathcal{M}} = \left\{ v \in V : \forall z \in V : Qv = Qz \Longrightarrow \|v\|_a^2 \leq \|z\|_a^2 \right\},$$

and let ω be a minimal constant such that

$$\forall v \in V^{\mathcal{M}} : \|Qv\|_a^2 \leq \omega \|v\|_a^2 .$$

Then the preconditioner from Algorithm 2.4 satisfies

$$\kappa \leq \omega = \sup_{v \in V} \frac{\|Qv\|_a^2}{\|v\|_a^2}. \tag{2.11}$$

Proof. Define an operator $G : X \to V^{\mathcal{M}}$ by

$$G : u \mapsto v, \quad \frac{1}{2} a(v, v) \to \min, \text{ s.t. } v \in V^{\mathcal{M}}, \ u = Qv. \tag{2.12}$$

Since a is positive definite on $V^{\mathcal{M}}$, the operator G is well defined. Define the bilinear form b on X by $b(u, v) = a(Gu, Gv)$. Now let u and v be as in (2.10). Since v is the solution of

$$\frac{1}{2} a(v, v) - \langle r, Qv \rangle \to \min, \text{ s.t. } v \in V^{\mathcal{M}},$$

it follows that u is the solution of

$$u \in X : \quad b(u, x) = \langle r, x \rangle , \quad \forall x \in X. \tag{2.13}$$

It remains to compare $\|u\|_a^2$ and $\|u\|_b^2$. Let $u \in X$ and define $v = Gu$. Then, from the minimization property (2.12) and the fact that $Qu = u \in X \subset V^{\mathcal{M}}$, it follows that

$$\|u\|_b^2 = \|v\|_a^2 \leq \|u\|_a^2 .$$

On the other hand,

$$\|u\|_a^2 = \|Qv\|_a^2 \leq \omega \|v\|_a^2 = \omega_0 \|u\|_b^2 ,$$

which concludes the proof. ■

3. Abstract Multispace BDDC

We will derive the abstract Multispace BDDC preconditioner from the abstract additive Schwarz theory. However, unlike in this theory, we will decompose some space between X and W rather than X. Suppose again that the bilinear form a is defined and symmetric positive semidefinite on a larger space $W \supset X$. In the design of the multispace preconditioner, we choose spaces V_k, $k = 1, \ldots, M$, such that

$$X \subset \sum_{k=1}^{M} V_k \subset W, \tag{3.1}$$

so this can also be viewed as replacing the space V in (2.9) by a sum $\sum_{k=1}^{M} V_k$. We do not wish to relate the two choices of the intermediate space at this point.

Assumption 3.1 *The form $a(\cdot,\cdot)$ is positive definite on each V_k separately.*

Algorithm 3.2 (Abstract Multispace BDDC) *Given spaces V_k and linear operators Q_k, $k = 1, \ldots, M$ such that $a(\cdot,\cdot)$ is positive definite on each space V_k, and*

$$X \subset \sum_{k=1}^{M} V_k, \quad Q_k : V_k \to X,$$

define the preconditioner $B : r \in X' \longmapsto u \in X$ by

$$B : r \mapsto u = \sum_{k=1}^{M} Q_k v_k, \quad v_k \in V_k : \quad a(v_k, z_k) = \langle r, Q_k z_k \rangle, \quad \forall z_k \in V_k. \tag{3.2}$$

We first formulate the condition number bound in the full strength allowed by the proof. We then find the bound used for the Multilevel BDDC as a corollary.

Theorem 3.3 *Define for all $k = 1, \ldots, M$ the spaces $V_k^{\mathcal{M}}$ by*

$$V_k^{\mathcal{M}} = \left\{ v_k \in V_k : \forall z_k \in V_k : Q_k v_k = Q_k z_k \Longrightarrow \|v_k\|_a^2 \leq \|z_k\|_a^2 \right\}.$$

If there exist constants C_0, ω, and a symmetric matrix $\mathcal{E} = (e_{ij})_{i,j=1}^M$, such that

$$\forall u \in X \quad \exists v_k \in V_k,\ k = 1, \ldots, M : u = \sum_{k=1}^{M} Q_k v_k,\ \ \sum_{k=1}^{M} \|v_k\|_a^2 \leq C_0 \|u\|_a^2 \tag{3.3}$$

$$\forall k = 1, \ldots, M \quad \forall v_k \in V_k^{\mathcal{M}} : \|Q_k v_k\|_a^2 \leq \omega \|v_k\|_a^2 \tag{3.4}$$

$$\forall z_k \in Q_k V_k,\ k = 1, \ldots, M : a(z_i, z_j) \leq e_{ij} \|z_i\|_a \|z_j\|_a, \tag{3.5}$$

then the preconditioner from Algorithm 3.2 satisfies

$$\kappa = \frac{\lambda_{\max}(BA_X)}{\lambda_{\min}(BA_X)} \leq C_0 \omega \rho(\mathcal{E}).$$

Proof. We interpret the Multispace BDDC preconditioner as an abstract additive Schwarz method. So the essential idea of the proof is to map the assumptions of the abstract additive Schwarz estimate from the decomposition (2.2) of the space X to the decomposition (3.1). Define the spaces

$$X_k = Q_k V_k.$$

We will show that the preconditioner (3.2) satisfies (2.3), where b_k is defined by

$$b_k(u_k, y_k) = a(G_k x, G_k z), \quad x, z \in X, \quad u_k = Q_k G_k x, \quad y_k = Q_k G_k z \tag{3.6}$$

with the operators $G_k : X \to V_k^{\mathcal{M}}$ defined by

$$G_k : u \mapsto v_k, \quad \frac{1}{2} a(v_k, v_k) \to \min, \text{ s.t. } v_k \in V_k^{\mathcal{M}},\ u = \sum_{k=1}^{M} Q_k v_k. \tag{3.7}$$

First, from the definition of operators G_k, spaces X_k, and because a is positive definite on V_k by Assumption (3.1), it follows that $G_k x$ and $G_k z$ in (3.6) exist

and are unique, so b_k is defined correctly. To prove (2.3), let v_k be as in (3.2) and note that v_k is the solution of

$$\frac{1}{2}a\left(v_k, v_k\right) - \langle r, Q_k v_k\rangle \to \min, \quad v_k \in V_k.$$

Consequently, the preconditioner (3.2) is an abstract additive Schwarz method and we only need to verify the inequalities (2.4)–(2.6). To prove (2.4), let $u \in X$. Then, with v_k from the assumption (3.3) and with $u_k = Q_k G_k v_k$ as in (3.6), it follows that

$$u = \sum_{k=1}^{M} u_k, \quad \sum_{k=1}^{M} \|u_k\|_{b_k}^2 = \sum_{k=1}^{M} \|v_k\|_a^2 \le C_0 \|u\|_a^2 .$$

Next, let $u_k \in X_k$. From the definitions of X_k and $V_k^{\mathcal{M}}$, it follows that there exist unique $v_k \in V_k^{\mathcal{M}}$ such that $u_k = Q_k v_k$. Using the assumption (3.4) and the definition of b_k in (3.6), we get

$$\|u_k\|_a^2 = \|Q_k v_k\|_a^2 \le \omega \|v_k\|_a^2 = \omega \|u_k\|_{b_k}^2 ,$$

which gives (2.5). Finally, (3.5) is the same as (2.6). ∎

The next Corollary was given without proof in [63, Lemma 1]. This is the special case of Theorem 3.3 that will actually be used. In the case when $M = 1$, this result was proved in [61].

Corollary 3.4 *Assume that the subspaces V_k are energy orthogonal, the operators Q_k are projections, $a\left(\cdot,\cdot\right)$ is positive definite on each space V_k, and*

$$\forall u \in X : \left[u = \sum_{k=1}^{M} v_k, \ v_k \in V_k \right] \Longrightarrow u = \sum_{k=1}^{M} Q_k v_k. \tag{3.8}$$

Then the abstract Multispace BDDC preconditioner from Algorithm 3.2 satisfies

$$\kappa = \frac{\lambda_{\max}(BA_X)}{\lambda_{\min}(BA_X)} \le \omega = \max_k \sup_{v_k \in V_k} \frac{\|Q_k v_k\|_a^2}{\|v_k\|_a^2} . \tag{3.9}$$

Proof. We only need to verify the assumptions of Theorem 3.3. Let $u \in X$ and choose v_k as the energy orthogonal projections of u on V_k. First, since the spaces V_k are energy orthogonal, $u = \sum v_k$, Q_k are projections, and from (3.8) $u = \sum Q_k v_k$, we get that $\|u\|_a^2 = \sum \|v_k\|_a^2$ which proves (3.3) with $C_0 = 1$. Next, the assumption (3.4) becomes the definition of the optimal ω in (3.9). Finally, (3.5) with $\mathcal{E} = I$ follows from the orthogonality of subspaces V_k. ∎

Remark 3.5 *The assumption (3.8) can be written as*

$$\sum_{k=1}^{M} Q_k P_k \Big|_X = I,$$

where P_k is the a-orthogonal projection from $\bigoplus_{j=1}^{M} V_j$ onto V_k. Hence, the property (3.8) is a type of decomposition of unity. In the case when $M = 1$, (3.8) means that the projection Q_1 is onto X.

3.1 Finite element setting

Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 2$ or 3, decomposed into N nonoverlapping subdomains Ω^s, $s = 1, ..., N$, which form a conforming triangulation of the domain Ω. Subdomains will be also called substructures. Each substructure is a union of Lagrangian $P1$ or $Q1$ finite elements with characteristic mesh size h, and the nodes of the finite elements between substructures coincide. The nodes contained in the intersection of at least two substructures are called boundary nodes. The union of all boundary nodes is called the interface Γ. The interface Γ is a union of three different types of open sets: *faces*, *edges*, and *vertices*. The substructure vertices will also be called corners. For the case of regular substructures, such as cubes or tetrahedrons, we can use the standard geometric definition of faces, edges, and vertices; cf., e.g., [42] for a more general definition.

Here, we find it more convenient to use the notation of abstract linear spaces and linear operators between them instead of the space $\mathbb{R}^n$ and matrices. The results can be easily converted to matrix language by choosing a finite element basis. The space of the finite element functions on Ω will be denoted as U. Let W^s be the space of finite element functions on substructure Ω^s, such that all of their degrees of freedom on $\partial\Omega^s \cap \partial\Omega$ are zero. Let

$$W = W^1 \times \cdots \times W^N,$$

and consider a bilinear form $a(\cdot, \cdot)$ arising from the second-order elliptic problem such as Poisson equation or a problem of linear elasticity.

Now $U \subset W$ is the subspace of all functions from W that are continuous across the substructure interfaces. We are interested in the solution of the problem (2.1) with $X = U$,

$$u \in U : a(u, v) = \langle f, v \rangle, \quad \forall v \in U, \tag{3.10}$$

where the bilinear form a is associated on the space U with the system operator A, defined by

$$A : U \mapsto U', \quad a(u, v) = \langle Au, v \rangle \text{ for all } u, v \in U, \tag{3.11}$$

and $f \in U'$ is the right-hand side. Hence, (3.10) is equivalent to

$$Au = f. \tag{3.12}$$

Define $U_I \subset U$ as the subspace of functions that are zero on the interface Γ, i.e., the "interior" functions. Denote by P the energy orthogonal projection from W onto U_I,

$$P : w \in W \longmapsto v_I \in U_I : a(v_I, z_I) = a(w, z_I), \quad \forall z_I \in U_I.$$

Functions from $(I-P)\,W$, i.e., from the nullspace of P, are called discrete harmonic; these functions are a-orthogonal to U_I and energy minimal with respect to increments in U_I. Next, let $\widehat{W}$ be the space of all discrete harmonic functions that are continuous across substructure boundaries, that is

$$\widehat{W} = (I-P)\,U. \tag{3.13}$$

In particular,

$$U = U_I \oplus \widehat{W}, \quad U_I \perp_a \widehat{W}. \tag{3.14}$$

A common approach in substructuring is to reduce the problem to the interface. Problem (3.10) is equivalent to two independent problems on the energy orthogonal subspaces U_I and $\widehat{W}$, and the solution u satisfies $u = u_I + \widehat{u}$, where

$$u_I \in U_I : a(u_I, v_I) = \langle f, v_I \rangle\,, \quad \forall v_I \in U_I, \tag{3.15}$$

$$\widehat{u} \in \widehat{W} : a(\widehat{u}, \widehat{v}) = \langle f, \widehat{v} \rangle\,, \quad \forall \widehat{v} \in \widehat{W}. \tag{3.16}$$

The solution of the interior problem (3.15) decomposes into independent problems, one for each substructure. The reduced problem (3.16) is then solved by preconditioned conjugate gradients. The reduced problem (3.16) is usually written equivalently as

$$\widehat{u} \in \widehat{W} : s(\widehat{u}, \widehat{v}) = \langle g, \widehat{v} \rangle\,, \quad \forall \widehat{v} \in \widehat{W},$$

where s is the form a restricted on the subspace $\widehat{W}$, and g is the reduced right-hand side, i.e., the functional f restricted to the space $\widehat{W}$. The reduced right-hand side g is usually written as

$$\langle g, \widehat{v} \rangle = \langle f, \widehat{v} \rangle - a(u_I, \widehat{v}), \quad \forall \widehat{v} \in \widehat{W}, \tag{3.17}$$

because $a(u_I, \widehat{v}) = 0$ by (3.14). In the implementation, the process of passing to the reduced problem becomes the elimination of the internal degrees of freedom of the substructures, also known as static condensation. The matrix of the reduced bilinear form s in the basis defined by interface degrees of freedom becomes the Schur complement S, and (3.17) becomes the reduced right-hand side. For details on the matrix formulation, see, e.g., [77, Sec. 4.6] or [81, Sec. 4.3].

The BDDC method is a two-level preconditioner characterized by the selection of certain *coarse degrees of freedom*, such as values at the corners and averages over edges or faces of substructures. Define $\widetilde{W} \subset W$ as the subspace of all functions such that the values of any coarse degrees of freedom have a common value for all relevant substructures and vanish on $\partial\Omega$, and $\widetilde{W}_\Delta \subset \widetilde{W}$ as the subspace of all functions such that their coarse degrees of freedom vanish. Next, define $\widetilde{W}_\Pi$ as the subspace of all functions such that their coarse degrees of freedom between adjacent substructures coincide, and such that their energy is minimal. Clearly, functions in $\widetilde{W}_\Pi$ are uniquely determined by the values of their coarse degrees of freedom, and

$$\widetilde{W}_\Delta \perp_a \widetilde{W}_\Pi, \quad \text{and} \quad \widetilde{W} = \widetilde{W}_\Delta \oplus \widetilde{W}_\Pi. \tag{3.18}$$

We assume that

$$a \text{ is positive definite on } \widetilde{W}. \tag{3.19}$$

This will be the case when a is positive definite on the space U, for which problem (2.1) is posed, and there are sufficiently many coarse degrees of freedom. We further assume that the coarse degrees of freedom are zero on all functions

from U_I, that is,

$$U_I \subset \widetilde{W}_\Delta. \tag{3.20}$$

In other words, the coarse degrees of freedom depend on the values on substructure boundaries only. From (3.18) and (3.20), it follows that the functions in $\widetilde{W}_\Pi$ are discrete harmonic, that is,

$$\widetilde{W}_\Pi = (I - P)\,\widetilde{W}_\Pi. \tag{3.21}$$

Next, let E be a projection from $\widetilde{W}$ onto U, defined by taking some weighted average on substructure interfaces. That is, we assume that

$$E : \widetilde{W} \to U, \quad EU = U, \quad E^2 = E. \tag{3.22}$$

Since a projection is the identity on its range, it follows that E does not change the interior degrees of freedom,

$$EU_I = U_I, \tag{3.23}$$

since $U_I \subset U$. Finally, we show that the operator $(I - P)\,E$ is a projection. From (3.23) it follows that E does not change interior degrees of freedom, so $EP = P$. Then, using the fact that $I - P$ and E are projections, we get

$$\begin{aligned} [(I - P)\,E]^2 &= (I - P)\,E\,(I - P)\,E \\ &= (I - P)\,(E - P)\,E \\ &= (I - P)\,(I - P)\,E = (I - P)\,E. \end{aligned} \tag{3.24}$$

Remark 3.6 *In [59, 61], the whole analysis was done in spaces of discrete harmonic functions after eliminating U_I, and the space $\widehat{W}$ was the solution space.*

In particular, $\widetilde{W}$ consisted of discrete harmonic functions only, while the same space here would be $(I-P)\widetilde{W}$. In our context, the decomposition of this space used in [59, 61] would be written as

$$(I-P)\widetilde{W} = (I-P)\widetilde{W}_\Delta \oplus \widetilde{W}_\Pi, \quad (I-P)\widetilde{W}_\Delta \perp_a \widetilde{W}_\Pi. \tag{3.25}$$

In the next section, the space X will be either U or $\widehat{W}$.

3.2 Two-level BDDC as Multispace BDDC

We show several different ways the original two-level BDDC algorithm can be interpreted as multispace BDDC. We first consider BDDC applied to the reduced problem (3.16), that is, (2.1) with $X = \widehat{W}$. This was the formulation considered in [59]. Define the space of discrete harmonic functions with coarse degrees of freedom coinciding across the interface

$$\widetilde{W}_\Gamma = (I-P)\,\widetilde{W}.$$

Because we work in the space of discrete harmonic functions and the output of the averaging operator E is not discrete harmonic, denote

$$E_\Gamma = (I-P)\,E. \tag{3.26}$$

In an implementation, discrete harmonic functions are represented by the values of their degrees of freedom on substructure interfaces, cf., e.g. [81]; hence, the definition (3.26) serves formal purposes only, so that everything can be written in terms of discrete harmonic functions without passing to the matrix formulation.

Algorithm 3.7 ([61], BDDC on the reduced problem) *Define the preconditioner* $r \in \widehat{W}' \longmapsto u \in \widehat{W}$ *by*

$$u = E_\Gamma w_\Gamma, \quad w_\Gamma \in \widetilde{W}_\Gamma : a\left(w_\Gamma, z_\Gamma\right) = \left\langle r, E_\Gamma z_\Gamma \right\rangle, \quad \forall z_\Gamma \in \widetilde{W}_\Gamma. \tag{3.27}$$

Proposition 3.8 ([61]) *The BDDC preconditioner on the reduced problem in Algorithm 3.7 is the abstract Multispace BDDC from Algorithm 3.2 with* $M = 1$ *and the space and operator given by*

$$X = \widehat{W}, \quad V_1 = \widetilde{W}_\Gamma, \quad Q_1 = E_\Gamma. \tag{3.28}$$

Also, the assumptions of Corollary 3.4 are satisfied.

Proof. We only need to note that the bilinear form $a(\cdot,\cdot)$ is positive definite on $\widetilde{W}_\Gamma \subset \widetilde{W}$ by (3.19), and the operator E_Γ defined by (3.26) is a projection by (3.24). The projection E_Γ is onto $\widehat{W}$ because E is onto U by (3.22), and $I - P$ maps U onto $\widehat{W}$ by the definition of $\widehat{W}$ in (3.13). ∎

Using the decomposition (3.25), we can split the solution in the space $\widetilde{W}_\Gamma$ into the independent solution of two subproblems: mutually independent problems on substructures as the solution in the space $\widetilde{W}_{\Gamma\Delta} = (I - P)\widetilde{W}_\Delta$, and a solution of the global coarse problem in the space $\widetilde{W}_\Pi$. The space $\widetilde{W}_\Gamma$ has the same decomposition as in (3.25), i.e.,

$$\widetilde{W}_\Gamma = \widetilde{W}_{\Gamma\Delta} \oplus \widetilde{W}_\Pi, \quad \text{and} \quad \widetilde{W}_{\Gamma\Delta} \perp_a \widetilde{W}_\Pi, \tag{3.29}$$

and Algorithm 3.7 can be rewritten as follows.

Algorithm 3.9 ([59], BDDC on the reduced problem) *Define the preconditioner* $r \in \widehat{W}' \longmapsto u \in \widehat{W}$ *by* $u = E_\Gamma \left(w_{\Gamma\Delta} + w_\Pi \right)$, *where*

$$w_{\Gamma\Delta} \in \widetilde{W}_{\Gamma\Delta} : a\left(w_{\Gamma\Delta}, z_{\Gamma\Delta} \right) = \left\langle r, E_\Gamma z_{\Gamma\Delta} \right\rangle, \quad \forall z_{\Gamma\Delta} \in \widetilde{W}_{\Gamma\Delta}, \tag{3.30}$$

$$w_\Pi \in \widetilde{W}_\Pi : a\left(w_\Pi, z_{\Gamma\Pi} \right) = \left\langle r, E_\Gamma z_{\Gamma\Pi} \right\rangle, \quad \forall z_{\Gamma\Pi} \in \widetilde{W}_\Pi. \tag{3.31}$$

Proposition 3.10 *The BDDC preconditioner on the reduced problem in Algorithm 3.9 is the abstract Multispace BDDC from Algorithm 3.2 with* $M = 2$ *and the spaces and operators given by*

$$X = \widehat{W}, \quad V_1 = \widetilde{W}_{\Gamma\Delta}, \quad V_2 = \widetilde{W}_\Pi, \quad Q_1 = Q_2 = E_\Gamma. \tag{3.32}$$

Also, the assumptions of Corollary 3.4 are satisfied.

Proof. Let $r \in \widehat{W}'$. Define the vectors v_i, $i = 1, 2$ in Multispace BDDC by (3.2) with V_i and Q_i given by (3.32). Let u, $w_{\Gamma\Delta}$, w_Π be the quantities in Algorithm 3.9, defined by (3.30)-(3.31). Using the decomposition (3.29), any $w_\Gamma \in \widetilde{W}_\Gamma$ can be written uniquely as $w_\Gamma = w_{\Gamma\Delta} + w_\Pi$ for some $w_{\Gamma\Delta}$ and w_Π corresponding to (3.2) as $v_1 = w_{\Gamma\Delta}$ and $v_2 = w_\Pi$, and $u = E_\Gamma \left(w_{\Gamma\Delta} + w_\Pi \right)$.

To verify the assumptions of Corollary 3.4, note that the decomposition (3.29) is $a-$orthogonal, $a(\cdot, \cdot)$ is positive definite on both $\widetilde{W}_{\Gamma\Delta}$ and $\widetilde{W}_\Pi$ as subspaces of $\widetilde{W}_\Gamma$ by (3.19), and E_Γ is a projection by (3.24). ∎

Next, we present a BDDC formulation on the space U with explicit treatment of interior functions in the space U_I as in [17, 58], i.e., in the way the BDDC algorithm was originally formulated.

Algorithm 3.11 ([17, 58], original BDDC) *Define the preconditioner* $r \in U' \longmapsto u \in U$ *as follows. Compute the interior pre-correction*

$$u_I \in U_I : a\left(u_I, z_I\right) = \left\langle r, z_I \right\rangle, \quad \forall z_I \in U_I. \tag{3.33}$$

Set up the updated residual

$$r_B \in U', \quad \left\langle r_B, v \right\rangle = \left\langle r, v \right\rangle - a\left(u_I, v\right), \quad \forall v \in U. \tag{3.34}$$

Compute the substructure correction

$$u_\Delta = E w_\Delta, \quad w_\Delta \in \widetilde{W}_\Delta : a\left(w_\Delta, z_\Delta\right) = \left\langle r_B, E z_\Delta \right\rangle, \quad \forall z_\Delta \in \widetilde{W}_\Delta. \tag{3.35}$$

Compute the coarse correction

$$u_\Pi = E w_\Pi, \quad w_\Pi \in \widetilde{W}_\Pi : a\left(w_\Pi, z_\Pi\right) = \left\langle r_B, E z_\Pi \right\rangle, \quad \forall z_\Pi \in \widetilde{W}_\Pi. \tag{3.36}$$

Add the corrections

$$u_B = u_\Delta + u_\Pi.$$

Compute the interior post-correction

$$v_I \in U_I : a\left(v_I, z_I\right) = a\left(u_B, z_I\right), \quad \forall z_I \in U_I. \tag{3.37}$$

Apply the combined corrections

$$u = u_B - v_I + u_I. \tag{3.38}$$

The interior corrections (3.33) and (3.37) decompose into independent Dirichlet problems, one for each substructure. The substructure correction (3.35) decomposes into independent constrained Neumann problems, one for each substructure. Thus, the evaluation of the preconditioner requires three problems to

be solved in each substructure, plus solution of the coarse problem (3.36). In addition, the substructure corrections can be solved in parallel with the coarse problem.

Remark 3.12 *As it is well known [17], the first interior correction (3.33) can be omitted in the implementation by starting the iterations from an initial solution such that the residual in the interior of the substructures is zero,*

$$a\left(u, z_I\right) - \left\langle f_X, z_I \right\rangle = 0, \quad \forall z_I \in U_I,$$

i.e., such that the error is discrete harmonic. Then the output of the preconditioner is discrete harmonic and thus the errors in all the CG iterations (which are linear combinations of the original error and outputs from the preconditioner) are also discrete harmonic by induction.

The following proposition will be the starting point for the multilevel case.

Proposition 3.13 *The original BDDC preconditioner in Algorithm 3.11 is the abstract Multispace BDDC from Algorithm 3.2 with $M = 3$ and the spaces and operators given by*

$$X = U, \quad V_1 = U_I, \quad V_2 = (I - P)\widetilde{W}_\Delta, \quad V_3 = \widetilde{W}_\Pi, \tag{3.39}$$

$$Q_1 = I, \quad Q_2 = Q_3 = (I - P)\, E, \tag{3.40}$$

and the assumptions of Corollary 3.4 are satisfied.

Proof. Let $r \in U'$. Define the vectors v_i, $i = 1, 2, 3$, in Multispace BDDC by (3.2) with the spaces V_i given by (3.39) and with the operators Q_i given

by (3.40). Let u_I, r_B, w_Δ, w_Π, u_B, v_I, and u be the quantities in Algorithm 3.11, defined by (3.33)-(3.38).

First, with $V_1 = U_I$, the definition of v_1 in (3.2) with $k = 1$ is identical to the definition of u_I in (3.33), so $u_I = v_1$.

Next, consider $w_\Delta \in \widetilde{W}_\Delta$ defined in (3.35). We show that w_Δ satisfies (3.2) with $k = 2$, i.e., $v_2 = w_\Delta$. So, let $z_\Delta \in \widetilde{W}_\Delta$ be arbitrary. From (3.35) and (3.34),

$$a\left(w_\Delta, z_\Delta\right) = \langle r_B, Ez_\Delta \rangle = \langle r, Ez_\Delta \rangle - a\left(u_I, Ez_\Delta\right). \tag{3.41}$$

Now from the definition of u_I by (3.33) and the fact that $PEz_\Delta \in U_I$, we get

$$\langle r, PEz_\Delta \rangle - a\left(u_I, PEz_\Delta\right) = 0, \tag{3.42}$$

and subtracting (3.42) from (3.41) gives

$$\begin{aligned} a\left(w_\Delta, z_\Delta\right) &= \langle r, \left(I - P\right) Ez_\Delta \rangle - a\left(u_I, \left(I - P\right) Ez_\Delta\right) \\ &= \langle r, \left(I - P\right) Ez_\Delta \rangle, \end{aligned}$$

because $a\left(u_I, \left(I - P\right) Ez_\Delta\right) = 0$ by orthogonality. To verify (3.2), it is enough to show that $Pw_\Delta = 0$; then $w_\Delta \in (I - P)\widetilde{W}_\Delta = V_2$. Since P is an a-orthogonal projection, it holds that

$$a\left(Pw_\Delta, Pw_\Delta\right) = a\left(w_\Delta, Pw_\Delta\right) = \langle r_B, EPw_\Delta \rangle = 0, \tag{3.43}$$

where we have used $EU_I \subset U_I$ following the assumption (3.23) and the equality

$$\langle r_B, z_I \rangle = \langle r, z_I \rangle - a\left(u_I, z_I\right) = 0,$$

for any $z_I \in U_I$, which follows from (3.34) and (3.33). Since a is positive definite on $\widetilde{W} \supset U_I$ by assumption (3.19), it follows from (3.43) that $Pw_\Delta = 0$.

In exactly the same way, from (3.36) – (3.38), we get that if $w_\Pi \in \widetilde{W}_\Pi$ is defined by (3.36), then $v_3 = w_\Pi$ satisfies (3.2) with $k = 3$. (The proof that $Pw_\Pi = 0$ can be simplified but there is nothing wrong with proceeding exactly as for w_Δ.)

Finally, from (3.37), $v_I = P\,(Ew_\Delta + Ew_\Pi)$, so

$$\begin{aligned} u &= u_I + (u_B - v_I) \\ &= u_I + (I - P)\,Ew_\Delta + (I - P)\,Ew_\Pi \\ &= Q_1 v_1 + Q_2 v_2 + Q_3 v_3. \end{aligned}$$

It remains to verify the assumptions of Corollary 3.4.

First, the spaces $\widetilde{W}_\Pi$ and $\widetilde{W}_\Delta$ are a-orthogonal by (3.18) and, from (3.20),

$$(I - P)\,\widetilde{W}_\Delta \subset \widetilde{W}_\Delta,$$

thus $(I - P)\,\widetilde{W}_\Delta \perp_a \widetilde{W}_\Pi$. Clearly, $(I - P)\,\widetilde{W}_\Delta \perp_a U_I$. Since $\widetilde{W}_\Pi$ consists of discrete harmonic functions from (3.21), so $\widetilde{W}_\Pi \perp_a U_I$, it follows that the spaces V_i, $i = 1, 2, 3$, given by (3.39), are a-orthogonal.

Next, $(I - P)\,E$ is a projection by (3.24), and so are the operators Q_i from (3.40).

It remains to prove the decomposition of unity (3.8). Let

$$u' = u_I + w_\Delta + w_\Pi \in U, \quad u_I \in U_I, \quad w_\Delta \in (I - P)\,\widetilde{W}_\Delta, \quad w_\Pi \in \widetilde{W}_\Pi, \quad (3.44)$$

and let

$$v = u_I + (I - P)\,Ew_\Delta + (I - P)\,Ew_\Pi.$$

From (3.44), $w_\Delta + w_\Pi \in U$ since $u' \in U$ and $u_I \in U_I \subset U$. Then $E\,(w_\Delta + w_\Pi) = w_\Delta + w_\Pi$ by (3.22), so

$$\begin{aligned} v &= u_I + (I - P)\,E\,(w_\Delta + w_\Pi) \\ &= u_I + (I - P)\,(w_\Delta + w_\Pi) \\ &= u_I + w_\Delta + w_\Pi = u', \end{aligned}$$

because both w_Δ and w_Π are discrete harmonic. ■

The next theorem shows an equivalence of the three algorithms introduced above.

Theorem 3.14 *The eigenvalues of the preconditioned operators from Algorithm 3.7, and Algorithm 3.9 are exactly the same. They are also the same as the eigenvalues from Algorithm 3.11, except possibly for multiplicity of the eigenvalue equal to one.*

Proof. From the decomposition (3.29), we can write any $w \in \widetilde{W}_\Gamma$ uniquely as $w = w_\Delta + w_\Pi$ for some $w_\Delta \in \widetilde{W}_{\Gamma\Delta}$ and $w_\Pi \in \widetilde{W}_\Pi$, so the preconditioned operators from Algorithms 3.7 and 3.9 are spectrally equivalent and we need only to show their spectral equivalence to the preconditioned operator from Algorithm 3.11. First, we note that the operator $A : U \mapsto U'$ defined by (3.11), and given in the block form as

$$A = \begin{bmatrix} \mathcal{A}_{II}\ \mathcal{A}_{I\Gamma} \\ \mathcal{A}_{\Gamma I}\ \mathcal{A}_{\Gamma\Gamma} \end{bmatrix},$$

with blocks

$$\mathcal{A}_{II} : U_I \to U_I', \quad \mathcal{A}_{I\Gamma} : U_I \to \widehat{W}',$$
$$\mathcal{A}_{\Gamma I} : \widehat{W} \to U_I', \quad \mathcal{A}_{\Gamma\Gamma} : \widehat{W} \to \widehat{W}',$$

is block diagonal and $\mathcal{A}_{\Gamma I} = \mathcal{A}_{I\Gamma} = 0$ for any $u \in U$, written as $u = u_I + \widehat{w}$, because $U_I \perp_a \widehat{W}$. Next, we note that the block $\mathcal{A}_{\Gamma\Gamma} : \widehat{W}' \to \widehat{W}$ is the Schur complement operator corresponding to the form s. Finally, since the block $\mathcal{A}_{II}$ is used only in the preprocessing step, the preconditioned operator from Algorithms 3.7 and 3.9 is simply $M_{\Gamma\Gamma}\mathcal{A}_{\Gamma\Gamma} : r \in \widehat{W}' \to u \in \widehat{W}$.

Let us now turn to Algorithm 3.11. Let the residual $r \in U$ be written as $r = r_I + r_\Gamma$, where $r_I \in U_I'$ and $r_\Gamma \in \widehat{W}'$. Taking $r_\Gamma = 0$, we get $r = r_I$, and it follows that $r_B = u_B = v_I = 0$, so $u = u_I$. On the other hand, taking $r = r_\Gamma$ gives $u_I = 0$, $r_B = r_\Gamma$, $v_I = Pu_B$ and finally $u = (I - P)\, E(w_\Delta + w_\Pi)$, so $u \in \widehat{W}$. This shows that the off-diagonal blocks of the preconditioner M are zero, and therefore it is block diagonal

$$M = \begin{bmatrix} M_{II} & 0 \\ 0 & M_{\Gamma\Gamma} \end{bmatrix}.$$

Next, let us take $u = u_I$, and consider $r_\Gamma = 0$. The algorithm returns $r_B = u_B = v_I = 0$, and finally $u = u_I$. This means that $M_{II}\mathcal{A}_{II}u_I = u_I$, so $M_{II} = \mathcal{A}_{II}^{-1}$. The operator $A : U \to U'$, and the block preconditioned operator $MA : r \in U' \to u \in U$ from Algorithm 3.11 can be written, respectively, as

$$A = \begin{bmatrix} \mathcal{A}_{II} & 0 \\ 0 & \mathcal{A}_{\Gamma\Gamma} \end{bmatrix}, \quad MA = \begin{bmatrix} I & 0 \\ 0 & M_{\Gamma\Gamma}\mathcal{A}_{\Gamma\Gamma} \end{bmatrix},$$

where the right lower block $M_{\Gamma\Gamma}\mathcal{A}_{\Gamma\Gamma} : r \in \widehat{W}' \to u \in \widehat{W}$ is exactly the same as the preconditioned operator from Algorithms 3.7 and 3.9. ∎

The BDDC condition number estimate is well known from [58]. Following Theorem 3.14 and Corollary 3.4, we only need to estimate $\|(I-P)\,Ew\|_a$ on $\widetilde{W}$.

Theorem 3.15 ([58]) *The condition number of the original BDDC algorithm satisfies $\kappa \leq \omega$, where*

$$\omega = \sup_{w \in \widetilde{W}} \frac{\|(I-P)\,Ew\|_a^2}{\|w\|_a^2}. \tag{3.45}$$

Remark 3.16 *In [58], the theorem was formulated by taking the supremum over the space of discrete harmonic functions $(I-P)\widetilde{W}$. However, the supremum remains the same by taking the larger space $\widetilde{W} \supset (I-P)\widetilde{W}$, since*

$$\frac{\|(I-P)\,Ew\|_a^2}{\|w\|_a^2} \leq \frac{\|(I-P)\,E\,(I-P)\,w\|_a^2}{\|(I-P)\,w\|_a^2}$$

from $E\,(I-P) = E$, which follows from (3.23), and from $\|w\|_a \geq \|(I-P)\,w\|_a$, which follows from the a-orthogonality of the projection P.

Before proceeding into the Multilevel BDDC section, let us write concisely the spaces and operators involved in the two-level preconditioner as

$$U_I \overset{P}{\leftarrow} U \overset{E}{\underset{\subset}{\leftarrow}} \widetilde{W}_\Delta \oplus \widetilde{W}_\Pi = \widetilde{W} \subset W.$$

We are now ready to extend this decomposition into the multilevel case.

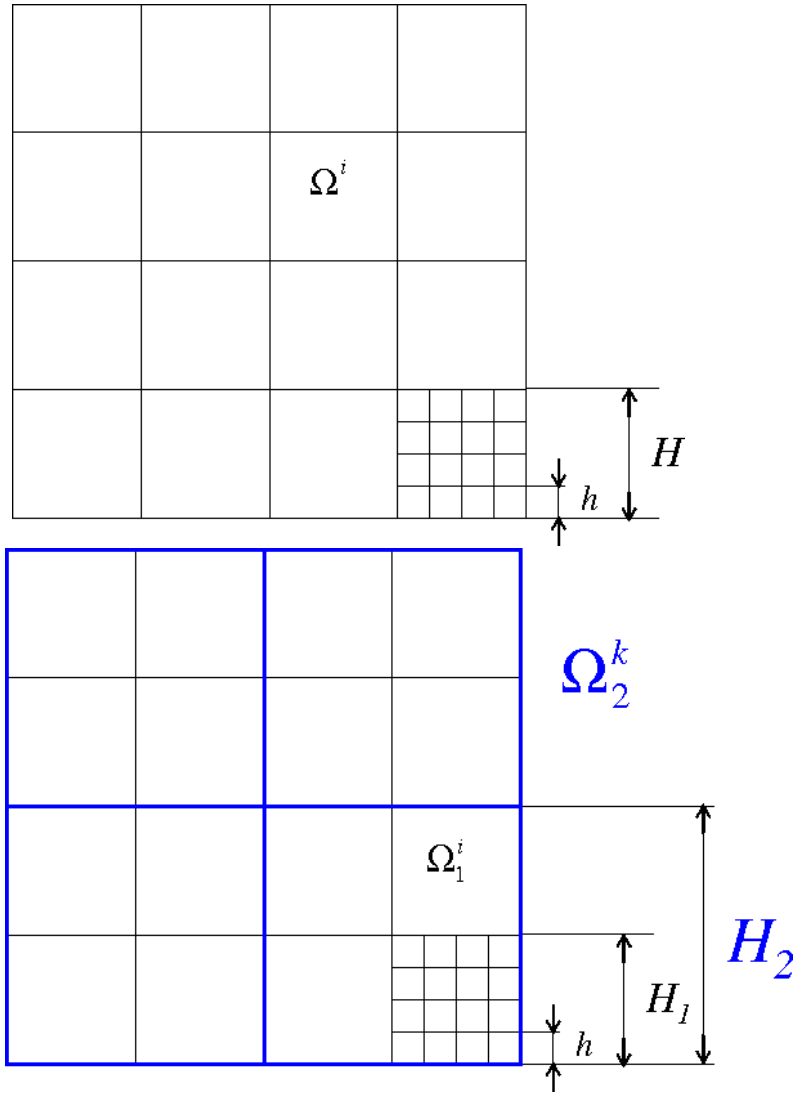


Figure 3.1: An example of a domain decomposition for the two-level (top) and the three-level (bottom) BDDC methods.

4. Multilevel BDDC

In this chapter, we generalize the two-level BDDC preconditioner to multiple levels, using the abstract Multispace BDDC framework from Algorithm 3.2. The substructuring components from Section 3.2 will be denoted by an additional subscript $_1$, as Ω_1^s, $s = 1, \dots N_1$, etc., and called level 1. The level 1 coarse problem (3.36) will be called the level 2 problem. It has the same finite element structure as the original problem (2.1) on level 1, so we put $U_2 = \widetilde{W}_{\Pi 1}$. Level 1 substructures are level 2 elements and level 1 coarse degrees of freedom are level 2 degrees of freedom. Repeating this process recursively, level $i-1$ substructures become level i elements, and the level i substructures are agglomerates of level i elements. Level i substructures are denoted by Ω_i^s, $s = 1, \dots, N_i$, and they are assumed to form a conforming triangulation with a characteristic substructure size H_i. For convenience, we denote by Ω_0^s the original finite elements and put $H_0 = h$. The interface Γ_i on level i is defined as the union of all level i boundary nodes, i.e., nodes shared by at least two level i substructures, and we note that $\Gamma_i \subset \Gamma_{i-1}$. Level $i-1$ coarse degrees of freedom become level i degrees of freedom. The shape functions on level i are determined by minimization of energy with respect to level $i-1$ shape functions, subject to the value of exactly one level i degree of freedom being one and the other level i degrees of freedom being zero. The minimization is done on each level i element (level $i-1$ substructure) separately, so the values of level $i-1$ degrees of freedom are in general discontinuous between level $i-1$ substructures, and only the values of

level i degrees of freedom between neighboring level i elements coincide. For an example of a decomposition for two and a three-level method, see Figure 3.2.

The development of the spaces on level i now parallels the finite element setting in Section 3.1. Denote $U_i = \widetilde{W}_{\Pi i-1}$. Let W_i^s be the space of functions on the substructure Ω_i^s, such that all of their degrees of freedom on $\partial\Omega_i^s \cap \partial\Omega$ are zero, and let

$$W_i = W_i^1 \times \cdots \times W_i^{N_i}.$$

Then $U_i \subset W_i$ is the subspace of all functions from W that are continuous across the interfaces Γ_i. Define $U_{Ii} \subset U_i$ as the subspace of functions that are zero on Γ_i, i.e., the functions "interior" to the level i substructures. Denote by P_i the energy orthogonal projection from W_i onto U_{Ii},

$$P_i : w_i \in W_i \longmapsto v_{Ii} \in U_{Ii} : a\left(v_{Ii}, z_{Ii}\right) = a\left(w_i, z_{Ii}\right), \quad \forall z_{Ii} \in U_{Ii}.$$

Functions from $\left(I - P_i\right) W_i$, i.e., from the nullspace of P_i, are called discrete harmonic on level i; these functions are a-orthogonal to U_{Ii} and energy minimal with respect to increments in U_{Ii}. Denote by $\widehat{W}_i \subset U_i$ the subspace of discrete harmonic functions on level i, that is

$$\widehat{W}_i = \left(I - P_i\right) U_i. \tag{4.1}$$

In particular, $U_{Ii} \perp_a \widehat{W}_i$. Define $\widetilde{W}_i \subset W_i$ as the subspace of all functions such that the values of any coarse degrees of freedom on level i have a common value for all relevant level i substructures and vanish on $\partial\Omega_i^s \cap \partial\Omega$, and $\widetilde{W}_{\Delta i} \subset W_i$ as the subspace of all functions such that their level i coarse degrees of freedom vanish. Define $\widetilde{W}_{\Pi i}$ as the subspace of all functions such that their level i coarse

degrees of freedom between adjacent substructures coincide, and such that their energy is minimal. Clearly, functions in $\widetilde{W}_{\Pi i}$ are uniquely determined by the values of their level i coarse degrees of freedom, and

$$\widetilde{W}_{\Delta i} \perp_a \widetilde{W}_{\Pi i}, \quad \widetilde{W}_i = \widetilde{W}_{\Delta i} \oplus \widetilde{W}_{\Pi i}. \tag{4.2}$$

We assume that the level i coarse degrees of freedom are zero on all functions from U_{Ii}, that is,

$$U_{Ii} \subset \widetilde{W}_{\Delta i}. \tag{4.3}$$

In other words, level i coarse degrees of freedom depend on the values on level i substructure boundaries only. From (4.2) and (4.3), it follows that the functions in $\widetilde{W}_{\Pi i}$ are discrete harmonic on level i, that is

$$\widetilde{W}_{\Pi i} = (I - P_i)\,\widetilde{W}_{\Pi i}. \tag{4.4}$$

Let E be a projection from $\widetilde{W}_i$ onto U_i, defined by taking some weighted average on Γ_i

$$E_i : \widetilde{W}_i \to U_i, \quad E_i U_{Ii} = U_{Ii}, \quad E_i^2 = E_i.$$

Since projection is the identity on its range, E_i does not change the level i interior degrees of freedom, in particular

$$E_i U_{Ii} = U_{Ii}. \tag{4.5}$$

The hierarchy of spaces and operators is shown concisely in Figure 4.1.

The Multilevel BDDC method is defined recursively [17, 63, 64] by solving the coarse problem on level i only approximately, by one application of the preconditioner on level $i+1$. Eventually, at the top level $L-1$, the coarse

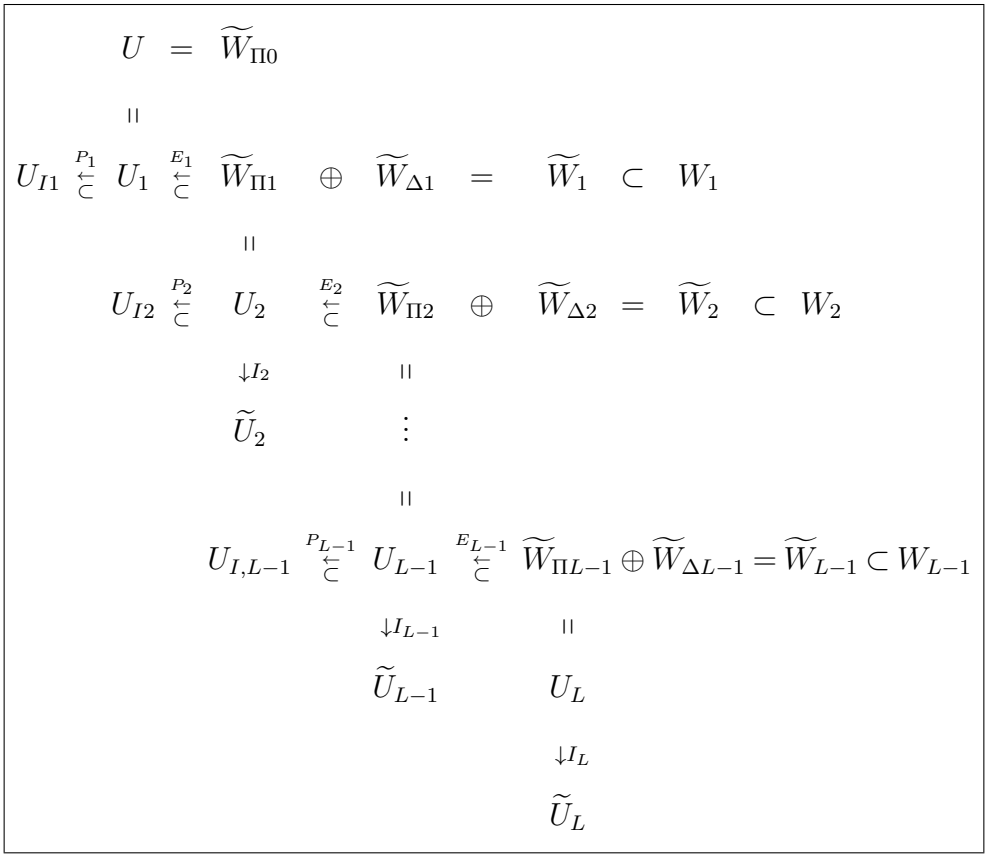


Figure 4.1: Spaces, embeddings and projections in the Multilevel BDDC.

problem, which is the level L problem, is solved exactly. We need a more formal description of the method here, which is provided by the following algorithm.

Algorithm 4.1 (Multilevel BDDC) *Define the preconditioner* $r_1 \in U_1' \longmapsto u_1 \in U_1$ *as follows:*

for $i = 1, \ldots, L-1$**,**

Compute interior pre-correction on level i*,*

$$u_{Ii} \in U_{Ii} : a\left(u_{Ii}, z_{Ii}\right) = \left\langle r_i, z_{Ii} \right\rangle, \quad \forall z_{Ii} \in U_{Ii}. \tag{4.6}$$

Get an updated residual on level i*,*

$$r_{Bi} \in U_i, \quad \left\langle r_{Bi}, v_i \right\rangle = \left\langle r_i, v_i \right\rangle - a\left(u_{Ii}, v_i\right), \quad \forall v_i \in U_i. \tag{4.7}$$

Find the substructure correction on level i*:*

$$w_{\Delta i} \in W_{\Delta i} : a\left(w_{\Delta i}, z_{\Delta i}\right) = \left\langle r_{Bi}, E_i z_{\Delta i} \right\rangle, \quad \forall z_{\Delta i} \in W_{\Delta i}. \tag{4.8}$$

Formulate the coarse problem on level i*,*

$$w_{\Pi i} \in W_{\Pi i} : a\left(w_{\Pi i}, z_{\Pi i}\right) = \left\langle r_{Bi}, E_i z_{\Pi i} \right\rangle, \quad \forall z_{\Pi i} \in W_{\Pi i}, \tag{4.9}$$

If $i = L-1$*, solve the coarse problem directly and set* $u_L = w_{\Pi L-1}$*,*

otherwise set up the right-hand side for level $i+1$*,*

$$r_{i+1} \in \widetilde{W}_{\Pi i}', \quad \left\langle r_{i+1}, z_{i+1} \right\rangle = \left\langle r_{Bi}, E_i z_{i+1} \right\rangle, \quad \forall z_{i+1} \in \widetilde{W}_{\Pi i} = U_{i+1}, \tag{4.10}$$

end.

for $i = L-1, \ldots, 1,$

Average the approximate corrections on substructure interfaces on level i*,*

$$u_{Bi} = E_i \left(w_{\Delta i} + u_{i+1} \right). \tag{4.11}$$

Compute the interior post-correction on level i*,*

$$v_{Ii} \in U_{Ii} : a\left(v_{Ii}, z_{Ii}\right) = a\left(u_{Bi}, z_{Ii}\right), \quad \forall z_{Ii} \in U_{Ii}. \tag{4.12}$$

Apply the combined corrections,

$$u_i = u_{Ii} + u_{Bi} - v_{Ii}. \tag{4.13}$$

end.

We can now show that the Multilevel BDDC can be cast as the Multispace BDDC on energy orthogonal spaces, using the hierarchy of spaces from Figure 4.1.

Lemma 4.2 *The Multilevel BDDC preconditioner in Algorithm 4.1 is the abstract Multispace BDDC preconditioner from Algorithm 3.2 with* $M = 2L-1$*, and the spaces and operators*

$$\begin{aligned}
X = U_1, \quad V_1 = U_{I1}, \quad V_2 = (I - P_1)\widetilde{W}_{\Delta 1}, \quad V_3 = U_{I2}, \\
V_4 = (I - P_2)\widetilde{W}_{\Delta 2}, \quad V_5 = U_{I3}, \quad \ldots \\
V_{2L-4} = (I - P_{L-2})\widetilde{W}_{\Delta L-2}, \quad V_{2L-3} = U_{IL-1}, \\
V_{2L-2} = (I - P_{L-1})\widetilde{W}_{\Delta L-1}, \quad V_{2L-1} = \widetilde{W}_{\Pi L-1},
\end{aligned} \tag{4.14}$$

$$
\begin{aligned}
Q_1 &= I, \quad Q_2 = Q_3 = (I - P_1)\, E_1, \\
Q_4 &= Q_5 = (I - P_1)\, E_1\, (I - P_2)\, E_2, \quad \ldots \\
Q_{2L-4} &= Q_{2L-3} = (I - P_1)\, E_1 \cdots (I - P_{L-2})\, E_{L-2}, \\
Q_{2L-2} &= Q_{2L-1} = (I - P_1)\, E_1 \cdots (I - P_{L-1})\, E_{L-1},
\end{aligned}
\tag{4.15}
$$

and the assumptions of Corollary 3.4 are satisfied.

Proof. Let $r_1 \in U_1'$. Define the vectors v_k, $k = 1, \ldots, 2L-1$ by (3.2) with the spaces and operators given by (4.14)-(4.15), and let u_{Ii}, r_{Bi}, $w_{\Delta i}$, $w_{\Pi i}$, r_{i+1}, u_{Bi}, v_{Ii}, and u_i be the quantities in Algorithm 4.1, defined by (4.6)-(4.13).

First, with $V_1 = U_{I1}$, the definition of v_1 in (3.2) is (4.6) with $i = 1$ and $u_{I1} = v_1$. We show that in general, for level $i = 1, \ldots, L-1$, and space $k = 2i-1$, we get (3.2) with $V_k = U_{Ii}$, so that $v_k = u_{Ii}$ and in particular $v_{2L-3} = u_{IL-1}$. So, let $z_{Ii} \in U_{Ii}$, $i = 2, \ldots, L-1$, be arbitrary. From (4.6) using (4.10) and (4.7),

$$
\begin{aligned}
a(u_{Ii}, z_{Ii}) = \langle r_i, z_{Ii} \rangle = \langle r_{Bi-1}, E_{i-1} z_{Ii} \rangle = \\
= \langle r_{i-1}, E_{i-1} z_{Ii} \rangle - a\left(u_{Ii-1}, E_{i-1} z_{Ii}\right).
\end{aligned}
\tag{4.16}
$$

Since from (4.6), using the fact that $P_{i-1} E_{i-1} z_{Ii} \in U_{Ii-1}$, it follows that

$$
\langle r_{i-1}, P_{i-1} E_{i-1} z_{Ii} \rangle - a\left(u_{Ii-1}, P_{i-1} E_{i-1} z_{Ii}\right) = 0,
$$

we get from (4.16),

$$
a(u_{Ii}, z_{Ii}) = \langle r_{i-1}, (I - P_{i-1})\, E_{i-1} z_{Ii} \rangle - a\left(u_{Ii-1}, (I - P_{i-1}) E_{i-1} z_{Ii}\right),
$$

and because $a\left(u_{Ii-1}, (I - P_{i-1}) E_{i-1} z_{Ii}\right) = 0$ by orthogonality, we obtain

$$
a(u_{Ii}, z_{Ii}) = \langle r_{i-1}, (I - P_{i-1})\, E_{i-1} z_{Ii} \rangle .
$$

Repeating this process recursively using (4.16), we finally get

$$\begin{aligned} a(u_{Ii}, z_{Ii}) = \langle r_{i-1}, (I - P_{i-1})\, E_{i-1} z_{Ii} \rangle = \quad & \ldots \\ = \langle r_1, (I - P_1)\, E_1 \cdots (I - P_{i-1})\, E_{i-1} z_{Ii} \rangle\,. & \end{aligned}$$

Next, consider $w_{\Delta i} \in \widetilde{W}_{\Delta i}$ defined by (4.8). We show that for $i = 1, \ldots, L-1$, and $k = 2i$, we get (3.2) with $V_k = \widetilde{W}_{\Delta i}$, so that $v_k = w_{\Delta i}$, and in particular $v_{2L-2} = w_{\Delta L-1}$. So, let $z_{\Delta i} \in \widetilde{W}_{\Delta i}$ be arbitrary. From (4.8) using (4.7),

$$a\,(w_{\Delta i}, z_{\Delta i}) = \langle r_{Bi}, E_i z_{\Delta i} \rangle = \langle r_i, E_i z_{\Delta i} \rangle - a\,(u_{Ii}, E_i z_{\Delta i})\,. \tag{4.17}$$

From the definition of u_{Ii} by (4.6) and since $P_i E_i z_{\Delta i} \in U_{Ii}$ it follows that

$$\langle r_i, P_i E_i z_{\Delta i} \rangle - a\,(u_{Ii}, P_i E_i z_{\Delta i}) = 0,$$

so (4.17) gives

$$a\,(w_{\Delta i}, z_{\Delta i}) = \langle r_i, (I - P_i)\, E_i z_{\Delta i} \rangle - a\,(u_{Ii}, (I - P_i)\, E_i z_{\Delta i})\,.$$

Next, because $a\,(u_{Ii}, (I - P_i)\, E_i z_{\Delta i}) = 0$ by orthogonality, and using (4.10),

$$a\,(w_{\Delta i}, z_{\Delta i}) = \langle r_i, (I - P_i)\, E_i z_{\Delta i} \rangle = \langle r_{Bi-1}, E_{i-1}\,(I - P_i)\, E_i z_{\Delta i} \rangle\,.$$

Repeating this process recursively, we finally get

$$\begin{aligned} a\,(w_{\Delta i}, z_{\Delta i}) = \langle r_i, (I - P_i)\, E_i z_{\Delta i} \rangle = \quad & \ldots \\ = \langle r_1, (I - P_1)\, E_1 \cdots (I - P_i)\, E_i z_{\Delta i} \rangle\,. & \end{aligned}$$

To verify (3.2), it remains to show that $P_i w_{\Delta i} = 0$; then $w_{\Delta i} \in (I - P_i)\widetilde{W}_{\Delta i} = V_k$. Since P_i is an a-orthogonal projection, it holds that

$$a\,(P_i w_{\Delta i}, P_i w_{\Delta i}) = a\,(w_{\Delta i}, P_i w_{\Delta i}) = \langle r_{Bi}, E_i P_i w_{\Delta i} \rangle = 0,$$

where we have used $E_i U_{Ii} \subset U_{Ii}$ following the assumption (4.5) and the equality

$$\langle r_{Bi}, z_{Ii} \rangle = \langle r_i, z_{Ii} \rangle - a\left(u_{Ii}, z_{Ii}\right) = 0$$

for any $z_{Ii} \in U_{Ii}$, which follows from (4.6) and (4.7).

In exactly the same way, we get that if $w_{\Pi L-1} \in \widetilde{W}_{\Pi L-1}$ is defined by (4.9), then $v_{2L-1} = w_{\Pi L-1}$ satisfies (3.2) with $k = 2L-1$.

Finally, from (4.11)-(4.13) for any $i = L-2, \ldots, 1$, we get

$$\begin{aligned}
u_i &= u_{Ii} + u_{Bi} - v_{Ii} \\
&= u_{Ii} + (I - P_i) E_i \left(w_{\Delta i} + u_{i+1}\right) \\
&= u_{Ii} + (I - P_i) E_i \left[w_{\Delta i} + u_{Ii+1} + (I - P_{i+1}) E_{i+1} \left(w_{\Delta i+1} + u_{i+2}\right)\right] \\
&= u_{Ii} + \\
&+ (I - P_i) E_i \left[w_{\Delta i} + \ldots + (I - P_{L-1}) E_{L-1} \left(w_{\Delta L-1} + u_{\Pi L-1}\right)\right],
\end{aligned}$$

and, in particular for u_1,

$$\begin{aligned}
u_1 &= u_{I1} + \\
&+ (I - P_1) E_1 \left[w_{\Delta 1} + \ldots + (I - P_{L-1}) E_{L-1} \left(w_{\Delta L-1} + u_{\Pi L-1}\right)\right] \\
&= Q_1 v_1 + Q_2 v_2 + \ldots + Q_{2L-2} v_{2L-2} + Q_{2L-1} v_{2L-1}.
\end{aligned}$$

It remains to verify the assumptions of Corollary 3.4.

The spaces $\widetilde{W}_{\Pi i}$ and $\widetilde{W}_{\Delta i}$, for all $i = 1, \ldots, L-1$, are a-orthogonal by (4.2) and from (4.3),

$$(I - P_i) \widetilde{W}_{\Delta i} \subset \widetilde{W}_{\Delta i},$$

thus $(I - P_i) \widetilde{W}_{\Delta i}$ is a-orthogonal to $\widetilde{W}_{\Pi i}$. Since $\widetilde{W}_{\Pi i} = U_{i+1}$ consists of discrete harmonic functions on level i from (4.4), and $U_{Ii+1} \subset U_{i+1}$, it follows by induction that the spaces V_k, given by (4.14), are a-orthogonal.

We now show that the operators Q_k defined by (4.15) are projections. From our definitions, coarse degrees of freedom on substructuring level i (from which we construct the level $i+1$ problem) depend only on the values of degrees of freedom on the interface Γ_i and $\Gamma_j \subset \Gamma_i$ for $j \geq i$. Then,

$$(I - P_j)E_j(I - P_i)E_i(I - P_j)E_j = (I - P_i)E_i(I - P_j)E_j. \tag{4.18}$$

Using (4.18), and since $(I - P_1)\, E_1$ is a projection by (3.24), we get

$$\begin{aligned}\left[(I - P_1)\, E_1 \cdots (I - P_i)\, E_i\right]^2 &= (I - P_1)\, E_1\, (I - P_1)\, E_1 \cdots (I - P_i)\, E_i \\ &= (I - P_1)\, E_1 \cdots (I - P_i)\, E_i,\end{aligned}$$

so the operators Q_k from (4.15) are projections.

It remains to prove the decomposition of unity (3.8). Let $u_i \in U_i$, such that

$$u_i' = u_{Ii} + w_{\Delta i} + u_{i+1}, \tag{4.19}$$

$$u_{Ii} \in U_{Ii}, \quad w_{\Delta i} \in (I - P_i)\, \widetilde{W}_{\Delta i}, \quad u_{i+1} \in U_{i+1} \tag{4.20}$$

and

$$v_i = u_{Ii} + (I - P_i)\, E_i w_{\Delta i} + (I - P_i)\, E_i u_{i+1}. \tag{4.21}$$

From (4.19), $w_{\Delta i} + u_{i+1} \in U_i$ since $u_i \in U_i$ and $u_{Ii} \in U_{Ii} \subset U_i$. Then $E_i\left[w_{\Delta i} + u_{i+1}\right] = w_{\Delta i} + u_{i+1}$ by (4.5), so

$$\begin{aligned}v_i &= u_{Ii} + (I - P_i)E_i\left[w_{\Delta i} + u_{i+1}\right] = u_{Ii} + (I - P_i)\left[w_{\Delta i} + u_{i+1}\right] = \\ &= u_{Ii} + w_{\Delta i} + u_{i+1} = u_{Ii} + w_{\Delta i} + u_{i+1} = u_i',\end{aligned}$$

because $w_{\Delta i}$ and u_{i+1} are discrete harmonic on level i. The fact that u_{i+1} in (4.19) and (4.21) are the same on arbitrary level i can be proved in exactly the

same way using induction and putting u_{i+1} in (4.19) as

$$u_{i+1} = u_{Ii+1} + \ldots + w_{\Delta L-1} + w_{\Pi L-1},$$
$$u_{Ii+1} \in U_{Ii+1}, \quad w_{\Delta L-1} \in (I - P_{L-1})\,\widetilde{W}_{\Delta L-1}, \quad w_{\Pi L-1} \in \widetilde{W}_{\Pi L-1},$$

and in (4.21) as

$$u_{i+1} = u_{Ii+1} + \ldots + (I - P_{i+1})\,E_{i+1} \cdots (I - P_{L-1})\,E_{L-1}\,(w_{\Delta L-1} + w_{\Pi L-1})\,,$$

which concludes the proof. ∎

The following bound follows from writing of the Multilevel BDDC as Multispace BDDC in Lemma 4.2 and the estimate for Multispace BDDC in Corollary 3.4.

Lemma 4.3 *If for some $\omega \geq 1$,*

$$\begin{aligned}
\|(I - P_1)E_1 w_{\Delta 1}\|_a^2 &\leq \omega\,\|w_{\Delta 1}\|_a^2 \quad \forall w_{\Delta 1} \in (I - P_1)\,\widetilde{W}_{\Delta 1},\\
\|(I - P_1)E_1 u_{I2}\|_a^2 &\leq \omega\,\|u_{I2}\|_a^2 \quad \forall u_{I2} \in U_{I2},\\
&\ldots\\
\|(I - P_1)E_1 \cdots (I - P_{L-1})E_{L-1} w_{\Pi L-1}\|_a^2 &\leq \omega\,\|w_{\Pi L-1}\|_a^2 \quad \forall w_{\Pi L-1} \in \widetilde{W}_{\Pi L-1},
\end{aligned} \tag{4.22}$$

then the Multilevel BDDC preconditioner (Algorithm 4.1) satisfies $\kappa \leq \omega$.

Proof. Choose the spaces and operators as in (4.14)-(4.15) so that $u_{I1} = v_1 \in V_1 = U_{I1}$, $w_{\Delta 1} = v_2 \in V_2 = (I - P_1)\,\widetilde{W}_{\Delta 1}$, …, $w_{\Pi L-1} = v_{2L-1} \in V_{2L-1} = \widetilde{W}_{\Pi L-1}$. The bound now follows from Corollary 3.4. ∎

The following lemma is an immediate consequence of Lemma 4.3, and it can be viewed as a multilevel analogy of Theorem 3.15. In fact, in the same way as

Theorem 3.15, formulated as Theorem 5.2, will serve as a starting point for the adaptive selection of constraints for the two-level BDDC method, the following Lemma, formulated as Theorem 6.1, will serve as a starting point for adaptive selection of constraints for the Multilevel BDDC method.

Lemma 4.4 *If for some* $\omega_i \geq 1$,

$$\|(I - P_i)E_i w_i\|_a^2 \leq \omega_i \|w_i\|_a^2, \quad \forall w_i \in \widetilde{W}_i, \quad i = 1, \ldots, L-1, \tag{4.23}$$

then the Multilevel BDDC preconditioner (Algorithm 4.1) satisfies $\kappa \leq \prod_{i=1}^{L-1} \omega_i$.

Proof. Note from Lemma 4.3 that $(I - P_1)\widetilde{W}_{\Delta 1} \subset \widetilde{W}_{\Delta 1} \subset \widetilde{W}_1$, $U_{I2} \subset \widetilde{W}_{\Pi 1} \subset \widetilde{W}_1$, and generally $(I - P_i)\widetilde{W}_{\Delta i} \subset \widetilde{W}_{\Delta i} \subset \widetilde{W}_i$, $U_{Ii+1} \subset \widetilde{W}_{\Pi i} \subset \widetilde{W}_i$. ∎

5. Adaptive Coarse Degrees of Freedom

We formulate an algorithm for adaptive selection of the coarse degrees of freedom for the two-level BDDC method. It was presented in [61] starting from corner constraints only, formulated in terms of FETI-DP, with the result translated to BDDC. Later the method has been extended in [65, 79] to the case of a general space $\widetilde{W}$ and implemented in BDDC directly using a projection on the subspace $\widetilde{W}$. The current formulation allows for an explicit coarse space and hence for a multilevel extension.

The space $\widetilde{W}$ is constructed using coarse degrees of freedom. These can be, e.g., values at corners, and averages over edges or faces. The space $\widetilde{W}$ is then given by the requirement that the coarse degrees of freedom on adjacent substructures coincide; for this reason, the terms coarse degrees of freedom and constraints are used interchangeably. The edge (or face) averages are necessary in 3D problems to obtain scalability with subdomain size. Ideally, one can prove the polylogarithmic condition number bound

$$\kappa \leq const \left(1 + \log \frac{H}{h}\right)^2, \tag{5.1}$$

where H is the subdomain size and h is the finite element size.

Remark 5.1 *The initial selection of constraints in the proposed adaptive approach will be done such that (5.1) is satisfied. See, e.g., [43] for theoretical reasoning for these constraints.*

To choose the space $\widetilde{W}$, cf. [61, Section 2.3], suppose we are given a linear operator $C : W \to X$ and define,

$$\widetilde{W} = \{w \in W : C\,(I - E)\,w = 0\}\,. \tag{5.2}$$

The values Cw will be called local coarse degrees of freedom, and the space $\widetilde{W}$ consists of all functions w whose local coarse degrees of freedom on adjacent substructures have zero jumps. To represent their common values, i.e., the global coarse degrees of freedom of vectors $u \in \widetilde{W}$, suppose there is a space U_c and linear operators $Q_P^{\mathrm{T}} : U \to U_c, \quad R_c : U_c \to X$ such that R_c is one-to-one, and injection $R : U \to W$ such that

$$CR = R_c Q_P^{\mathrm{T}}. \tag{5.3}$$

The space $\widetilde{W}$ then satisfies

$$\widetilde{W} = \{w \in W : \exists u_c \in U_c : Cw = R_c u_c\}\,,$$

and from (5.3), for $w \in \widetilde{W}$, the unique u_c that satisfies $Cw = R_c u_c$ is given by

$$u_c = Q_P^{\mathrm{T}} v, \quad w = Rv.$$

In order to formulate the adaptive algorithm, we first restate the condition number bound from (2.11) in a way suitable for our purposes, cf. Theorem 3.15.

Theorem 5.2 ([61, Theorem 3]) *The condition number bound (3.45) of the two-level BDDC satisfies $\kappa \leq \omega$, where*

$$\omega = \sup_{w \in \widetilde{W}} \frac{\|(I - P)\,Ew\|_a^2}{\|w\|_a^2} = \sup_{w \in \widetilde{W}} \frac{\|(I - (I - P)\,E)\,w\|_a^2}{\|w\|_a^2}. \tag{5.4}$$

With respect to Remark 3.16, we can conveniently look for the condition number bound ω only in the subspace $\widetilde{W}_\Gamma = (I-P)\,\widetilde{W}$. Next, observe that $(I-E)\,Pv = 0$ for all $v \in W$, so we can define the space $\widetilde{W}$ in (5.2) using discrete harmonic functions $w \in (I-P)\,W$, for which

$$(I-(I-P)\,E)\,w = (I-P)\,(I-E)\,w, \tag{5.5}$$

because $Pw = 0$ if $w \in (I-P)\,W$. Clearly, the bound (5.4) can be found as a maximum eigenvalue of an associated eigenvalue problem, using (5.5) written as

$$\langle (I-P)\,(I-E)\,w, (I-P)\,(I-E)\,z\rangle_a = \lambda\,\langle w,z\rangle_a \quad \forall z \in \widetilde{W}_\Gamma. \tag{5.6}$$

Remark 5.3 *The eigenvalue problem (5.6) corresponds to the right-hand side of (5.4) combined with (5.5). This is motivated by the definition (5.2) of the space $\widetilde{W}$: in the adaptive algorithm we will prescribe certain weighted "jumps" of functions to be zero across substructure interfaces.*

The following is a well known result from linear algebra, cf., e.g., [16, Theorem 5.2].

Lemma 5.4 (Courant-Fisher-Weyl minmax principle) *Let $c(\cdot,\cdot)$ be symmetric positive semidefinite bilinear form on vector space V of dimension n and $b(\cdot,\cdot)$ symmetric positive definite bilinear form on V. Then the generalized eigenvalue problem*

$$w \in V : c(w,u) = \lambda b(w,u) \quad \forall u \in V,$$

has n linearly independent eigenvectors w_ℓ and the corresponding eigenvalues are real and nonnegative and the eigenvectors are stationary points of the Rayleigh

quotient $c(w,w)/b(w,w)$*, with the stationary values equal to* λ_i*. Order* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$*. Then, for any subspace* $V_k \subset V$ *of dimension* $n-k$*,*

$$\max_{w \in V_k, w \neq 0} \frac{c(w,w)}{b(w,w)} \geq \lambda_{k+1},$$

with equality if

$$V_k = \{w \in V : c(w_\ell, w) = 0, \quad \forall \ell = 1, \ldots, k\}.$$

Since the bilinear form on the left-hand side of (5.6) is symmetric positive semidefinite and the bilinear form on the right-hand side is symmetric positive definite, Lemma 5.4, using (5.5), implies

Corollary 5.5 ([61]) *The generalized eigenvalue problem (5.6) has eigenvalues* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$*. Denote the corresponding eigenvectors by* w_ℓ*. Then, for any* $k = 1, \ldots, n-1$*, and any linear functionals* L_ℓ *on* W_Γ*,* $\ell = 1, \ldots, k$*,*

$$\max \left\{ \frac{\|(I-P)(I-E)w\|_a^2}{\|w\|_a^2} : w \in \widetilde{W}_\Gamma, \ L_\ell(w) = 0 \ \forall \ell = 1, \ldots, k \right\} \geq \lambda_{k+1},$$

with equality if

$$L_\ell(w) = a((I-P)(I-E)w_\ell, (I-P)(I-E)w). \tag{5.7}$$

Therefore, because $(I-E)$ is a projection, the optimal decrease of the condition number bound (5.4) can be achieved by adding to the constraint matrix C in the definition of $\widetilde{W}$ the rows c_ℓ defined by $c_\ell^{\mathrm{T}} w = L_\ell(w)$. However, solving the global eigenvalue problem (5.6) is expensive, and the vectors c_ℓ are not of the form required for substructuring, i.e., each c_ℓ with nonzero entries corresponding to only one corner, an edge or a face at a time.

For these reasons, we replace (5.6) by a collection of local problems, each defined by considering only two adjacent subdomains Ω^s and Ω^t. All quantities associated with such pairs will be denoted by the superscript st. Here, subdomains are considered adjacent if they share an edge in 2D, or a face in 3D. We note that edges in 2D will be regarded as faces. Using also (5.5), the generalized eigenvalue problem (5.7) becomes a problem to find $w \in \widetilde{W}_\Gamma^{st}$ such that

$$a^{st}\left(\left(I-P^{st}\right)\left(I-E^{st}\right)w, \left(I-P^{st}\right)\left(I-E^{st}\right)z\right) = \lambda\, a^{st}(w,z) \quad \forall z \in \widetilde{W}_\Gamma^{st}. \tag{5.8}$$

Assumption 5.6 *The corner constraints are already sufficient to prevent relative rigid body motions of any pair of adjacent substructures, so*

$$\forall w \in \widetilde{W}^{st} : A^{st}w = 0 \Rightarrow \left(I-E^{st}\right)w = 0,$$

i.e., the corner degrees of freedom are sufficient to constrain the rigid body modes of the two substructures into a single set of rigid body modes, which are continuous across the interface Γ^{st}.

The maximal eigenvalue ω^{st} of (5.8) is finite due to Assumption 5.6, and we define the heuristic *condition number indicator*

$$\widetilde{\omega} = \max\left\{\omega^{st} : \Omega^s \text{ and } \Omega^t \text{ are adjacent}\right\}. \tag{5.9}$$

Considering two adjacent subdomains Ω^s and Ω^t only, we get the added constraints $L_\ell(w) = 0$ from (5.7) as

$$a^{st}\left(\left(I-P^{st}\right)\left(I-E^{st}\right)w_\ell, \left(I-P^{st}\right)\left(I-E^{st}\right)w\right) = 0 \quad \forall \ell = 1,\ldots,k, \tag{5.10}$$

where w_ℓ are the eigenvectors corresponding to the k largest eigenvalues from (5.8).

To formulate a numerical algorithm, we need to write the generalized eigenvalue problem (5.8) and the added constraints (5.10) in terms of matrices and vectors. To avoid complicated notation, we now drop the superscripts st, or, equivalently, let us consider a domain which consists of only two substructures. For convenience, we will also identify finite element functions with vectors formed by their degrees of freedom, and we will also identify linear operators with their matrices, in bases that will be clear from the context.

The vectors of the local substructure degrees of freedom $w^s \in W^s$ and the vector of the global degrees of freedom $u \in U$ are related by $w^s = R^s u$, where R^s is the restriction operator (a zero-one matrix), so that

$$R^s : U \to W^s, \quad R^s R^{s\mathrm{T}} = I, \tag{5.11}$$

and $R : U \to W$. The Schur complement matrices S^s are assumed to be symmetric and positive semidefinite. Let us consider the vectors and matrices to be given in a block form

$$w = \begin{bmatrix} w^s \\ w^t \end{bmatrix}, \quad S = \begin{bmatrix} S^s & \\ & S^t \end{bmatrix}, \quad R = \begin{bmatrix} R^s \\ R^t \end{bmatrix}. \tag{5.12}$$

We will need a more specific construction of the matrix C in the substructuring framework. We build a block diagonal matrix C satisfying (5.3) by

$$C = \begin{bmatrix} C^s & \\ & C^t \end{bmatrix}, \quad C^s = R_c^s Q_P^{\mathrm{T}} R^{s\mathrm{T}}. \tag{5.13}$$

Then (5.3) follows from (5.13) and (5.11).

Here is an interpretation. The matrix C^s maps a vector of local degrees of freedom on substructure i to a vector of local coarse degrees of freedom on the

substructure, and R_c^s restricts a vector of all global coarse degree of freedom to a vector of local coarse degree of freedom on substructure s. A global coarse degree of freedom is given by a row of Q_P. The operator Q_P^{T} acts on vectors of global degrees of freedom in U and it selects global coarse degrees of freedom in U_c as linear combinations of global degrees of freedom. In our problems, there are corner coarse degrees of freedom, which are values at corners, and edge (face) coarse degrees of freedom, which are linear combinations of values on edges (faces).

Consider the space $\widetilde{W}$ given some initial constraint matrix C containing at least corner constraints. Let us denote $D = C\,(I - E)$ and define the orthogonal projection onto $\operatorname{null} D$ by

$$\Pi = I - D^{\mathrm{T}} \left(DD^{\mathrm{T}}\right)^{-1} D.$$

The generalized eigenvalue problem (5.6) now becomes

$$\Pi\,(I - P)^{\mathrm{T}}\,(I - E)^{\mathrm{T}}\,S\,(I - E)\,(I - P)\,\Pi w = \lambda \Pi S \Pi w. \tag{5.14}$$

Since

$$\operatorname{null} \Pi S \Pi \subset \operatorname{null} \Pi\,(I - P)^{\mathrm{T}}\,(I - E)^{\mathrm{T}}\,S\,(I - E)\,(I - P)\,\Pi, \tag{5.15}$$

the eigenvalue problem (5.14) reduces in the factorspace modulo $\operatorname{null} \Pi S \Pi$ to the problem with the operator on the right-hand side positive definite. In some of the computations, we have used the subspace iteration method LOBPCG [45] to find the dominant eigenvalues and their eigenvectors. The LOBPCG iterations then simply run in the factorspace. To use standard eigenvalue solvers, (5.14) is converted to a matrix eigenvalue problem by penalizing the components in

null D and rigid body modes, already described in [61, 79], and recalled here for completeness. We can formulate 5.14 as a generalized eigenvalue problem with the matrix on the right-hand side positive definite using the following procedure.

Theorem 5.7 ([61]) *Let $a > 0$. Then the nonzero eigenvalues and the eigenvectors of (5.14) are same as those of the generalized eigenvalue problem*

$$\Pi \left(I - P\right)^{\mathrm{T}} \left(I - E\right)^{\mathrm{T}} S \left(I - E\right) \left(I - P\right) \Pi w = \lambda \left(\Pi S \Pi + a \left(I - \Pi\right)\right) w, \quad (5.16)$$

The matrix on the left-hand side is symmetric positive semidefinite and if the pair of substructures intersects boundary with Dirichlet boundary conditions, the matrix on the right-hand side is symmetric positive definite.

In practice, we choose a to be roughly the same magnitude as S. Note that if the eigenvalues are computed approximately, the result will in general depend on a. Also, for subspace iteration methods the matrices, in particular the Schur complement S, need not to be formulated explicitly, and only matrix-vector products are evaluated. However, the matrices on both sides may be still singular because of rigid body modes that move substructures s and t as a whole. To reduce (5.16) to an eigenvalue problem with the matrix on the right-hand side positive definite, we use matrix Z that generates a superspace of rigid body modes of the two substructures

$$\operatorname{null} S \subset \operatorname{range} Z.$$

The matrix Z can be available from finite element software or can be easily computed from the geometry of the finite element mesh with rigid body modes as its columns, e.g., [81, Chapter 8]. To avoid using any information other

than the system matrices, we can instead use as Z a block diagonal matrix of the coarse basis functions of the two subtructures because their span contains the rigid body modes. However, the computations in this case will be more expensive because there are typically more coarse basis functions for the two substructures than the number of the rigid body modes.

Remark 5.8 *We have used in the computations the true rigid body modes computed from the mesh geometry for the two-level method, and the coarse basis functions on higher levels of Multilevel BDDC.*

We find a basis of $\operatorname{null}\left(\Pi S\Pi + a\left(I-\Pi\right)\right)$ by computing the nullspace of a much smaller symmetric positive semi-definite matrix,

$$\operatorname{null}\left(Z^{\mathrm{T}}\left(\Pi S\Pi + a\left(I-\Pi\right)\right)Z\right) = \operatorname{range} K,$$

and applying the QR decomposition

$$ZK = QR, \quad Q^{\mathrm{T}}Q = I,$$

which gives

$$\operatorname{range} Q = \operatorname{null}\left(\Pi S\Pi + a\left(I-\Pi\right)\right).$$

Consequently,

$$\overline{\Pi} = I - QQ^{\mathrm{T}},$$

is the orthogonal projection onto $\operatorname{range}\left(\Pi S\Pi + a\left(I-\Pi\right)\right)$.

The following theorem follows similarly as Theorem 5.7.

Theorem 5.9 ([61]) *The nonzero eigenvalues and the corresponding eigenvectors of*

$$\Pi \left(I - E\right)^{\mathrm{T}} S \left(I - E\right) \Pi w_k = \lambda_k \left(\Pi S \Pi + a \left(I - \Pi\right)\right) w_k,$$

are the same as nonzero eigenvalues and corresponding eigenvectors of

$$X w_k = \lambda_k \overline{Y} w_k, \tag{5.17}$$

where

$$\begin{aligned} X &= \Pi \left(I - E\right)^{\mathrm{T}} S \left(I - E\right) \Pi, \\ \overline{Y} &= \left(\overline{\Pi} \left(\Pi S \Pi + a \left(I - \Pi\right)\right) \overline{\Pi} + a \left(I - \overline{\Pi}\right)\right). \end{aligned}$$

In addition, $\overline{Y}$ is symmetric positive definite.

Remark 5.10 *The automatic decomposition can result, due to the vagaries of a mesh partitioner, in very irregular substructures including spurious mechanisms, see Figs. 5.2, or 7.8. It such cases the nullspace of the Schur complement is in general unknown. See Section 5.1 for further discussion.*

From the matrix form (5.15) of the eigenvalue problem, the constraints to be added are

$$L_\ell \left(w\right) = w_\ell^{\mathrm{T}} \Pi \left(I - P\right)^{\mathrm{T}} \left(I - E\right)^{\mathrm{T}} S \left(I - E\right) \left(I - P\right) \Pi w = 0.$$

That is, we wish to add to the constraint matrix C the rows

$$c_\ell = w_\ell^{\mathrm{T}} \Pi \left(I - P\right)^{\mathrm{T}} \left(I - E\right)^{\mathrm{T}} S \left(I - E\right) \left(I - P\right) \Pi. \tag{5.18}$$

Proposition 5.11 *The vectors c_ℓ, constructed for a domain consisting of only two substructures Ω^s and Ω^t, have matching entries on the interface between the two substructures, with opposite signs.*

Proof: Consider the vector $w \in W$ that has two entries equal to 1, corresponding to a degree of freedom on the interface, and all other entries equal to 0. Using the definition of c_ℓ and because $(I - E)\, u = 0$ for all $u \in U$, we get $c_\ell w = L_\ell\,(w) = 0$. ∎

It remains to construct the augmentation of the primal constraint matrix Q_P from the augmentation c_ℓ. Due to Proposition 5.11, each row of c_ℓ can be split into two blocks and written as

$$c_\ell = \begin{bmatrix} c_\ell^s & -c_\ell^s \end{bmatrix}.$$

The augmentation of Q_P is then constructed by simply taking c_ℓ^s, and computing

$$q = R^{s\mathrm{T}} c_\ell^{s\mathrm{T}}. \tag{5.19}$$

Because R^s is a $0-1$ matrix, it means that columns q are formed by a scattering of the entries in c_ℓ^{sT}. Each column of q defines a coarse degree of freedom, which is used to augment the matrix Q_P as

$$[\, Q_P \;\; q \,]. \tag{5.20}$$

Unfortunately, the added columns will generally have nonzero entries over all of the interface of Ω^s and Ω^t, including the edges in 3D where Ω^s and Ω^t intersect other substructures. Consequently, the added q are not of the form required for substructuring, i.e., each q with nonzeros in one edge or face only. In the computations reported in Section 7, we drop the adaptively generated edge constraints in 3D. Then it is no longer guaranteed that the condition number indicator $\widetilde{\omega} \leq \tau$. However, the method is still observed to perform well.

Now Corollary 5.5 and the formulation of the constraints (5.18)-(5.20) suggest a way to decrease the indicator $\widetilde{\omega}$ and the proposed adaptive algorithm follows.

Algorithm 5.12 (Adaptive BDDC [61]) *Find the smallest k for every two adjacent substructures to guarantee that $\lambda_{k+1} \leq \tau$, where τ is a given tolerance, and add the constraints (5.10) to the definition of $\widetilde{W}$.*

5.1 Preconditioned LOBPCG

The most important step towards future parallel implementation of the adaptive method seems to be an efficient solution of local generalized eigenvalue problems. An attractive approach is to use an inverse-free method, such as the method by Golub and Ye [32], or LOBPCG by Knyazev [45]. These methods allow the matrices to be in a matrix-free format, i.e., as functions for matrix-vector multiplication, which are readily available in our implementation. In particular, LOBPCG might be more suitable because it allows for the resolution of more eigenpairs at once and it can simply run in the factorspace with the operator on the right-hand side only positive semi-definite. Initial experiments reveal that the non-preconditioned LOBPCG as well as the method of Golub and Ye work well for reasonably hard problems [79]. Unfortunately, it turns out that many iterations are required for problems with extremely irregular structures and/or high jumps in coefficients, and preconditioning of the local eigenproblems seems to be necessary [65].

One desirable property of a preconditioner is that it must be invariant on the null S. Also, unless the component of the solution in the direction of the

nullspace is small, the errors will accumulate, which may eventually result in instability of the code at the Rayleigh-Ritz step, but only after a large number of steps [46]. Because the Schur complement operator S plays a central role in the operators on both sides of the local generalized eigenvalue problems (5.14), (5.16), and (5.17), a straightforward idea is to use a *local* version of the BDDC preconditioner, denoted here as M^{loc}. The only difference is that this preconditioner acts on the larger space W, unlike the preconditioners in Section 3.2, and therefore in place of averaging E we have used an injection R instead. The coarse space correction is obtained using corner (and in 3D edge) constraints, shared by the two substructures. We restrict the action of the preconditioner in the suitable subspaces using the two projections Π resp. $\overline{\Pi}$ introduced previously,

$$\Pi M^{loc}\Pi \qquad \text{resp..} \qquad \overline{\Pi}\Pi M^{loc}\Pi\overline{\Pi}. \tag{5.21}$$

Unfortunately, as we have pointed out already in Remark 5.10, the automatic decomposition of the finite element mesh can result, due to the vagaries of a mesh partitioner (in the experiments we have used METIS 4.0 [34]), in very irregular substructures including spurious mechanisms, see Figs. 5.2, or 7.8. In such cases the nullspace of the Schur complement is in general unknown, and the LOBPCG iterations with the preconditioner (5.21) can fail. To detect the eigenvectors in the nullspace of the operator on the right-hand side in (5.17), resp. (5.16), we have used again LOBPCG with M^{loc} as a preconditioner. However because the Schur complement is singular, so is M^{loc}. To circumvent this, we have applied a shift and the action of the preconditioner was in this case given as $M^{loc} + I$. Once the nullspace is detected, we can enrich the nullspace basis, reconstruct the projection $\overline{\Pi}$ and rerun the preconditioned LOBPCG iterations.

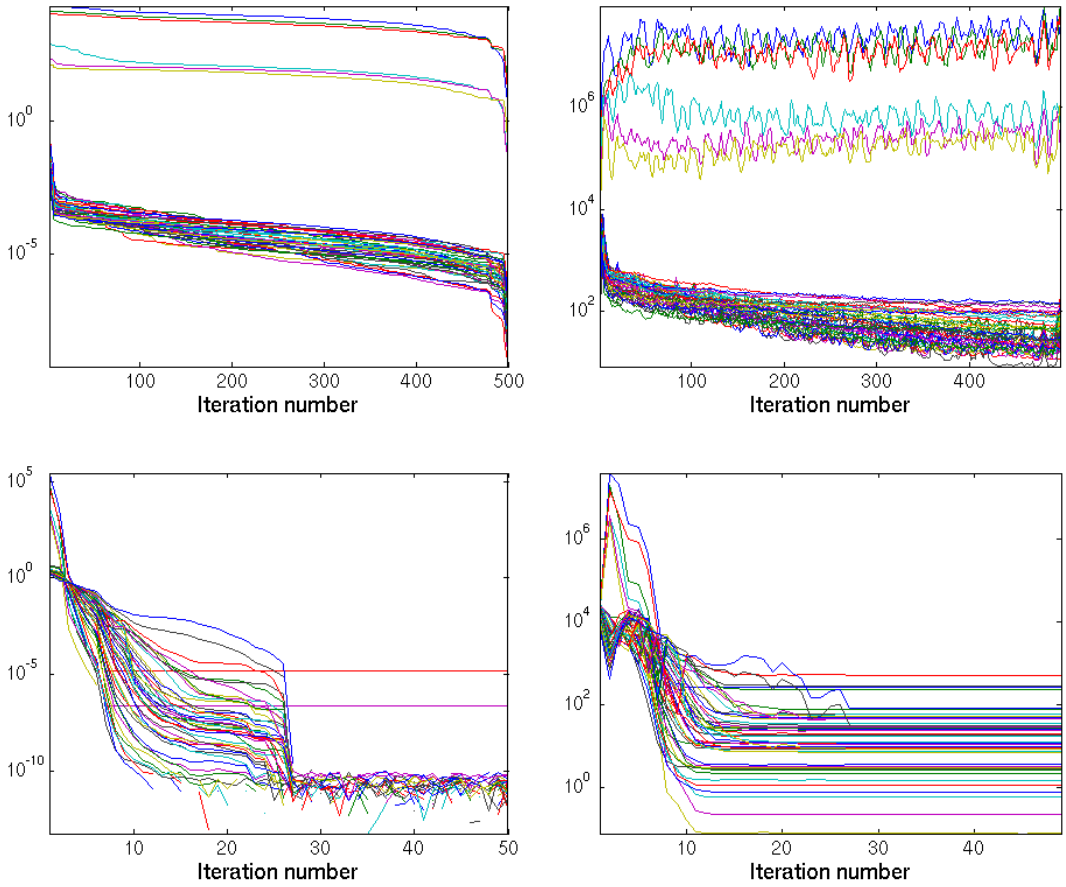


Figure 5.1: Comparison of the non-preconditioned (top) vs. preconditioned LOBPCG (bottom) for one of the faces with large jumps in coefficients of the composite cube problem, cf. Chapter 7 and Fig. 7.2. Estimated eigenvalue errors are in the panels on the left-hand side, and Euclidean norms of residuals for different eigenpairs are in the panels on the right-hand side. We see that LOBPCG without a preconditioner showed essentially no convergence, and with the preconditioner we have reached convergence in less than 30 iterations.

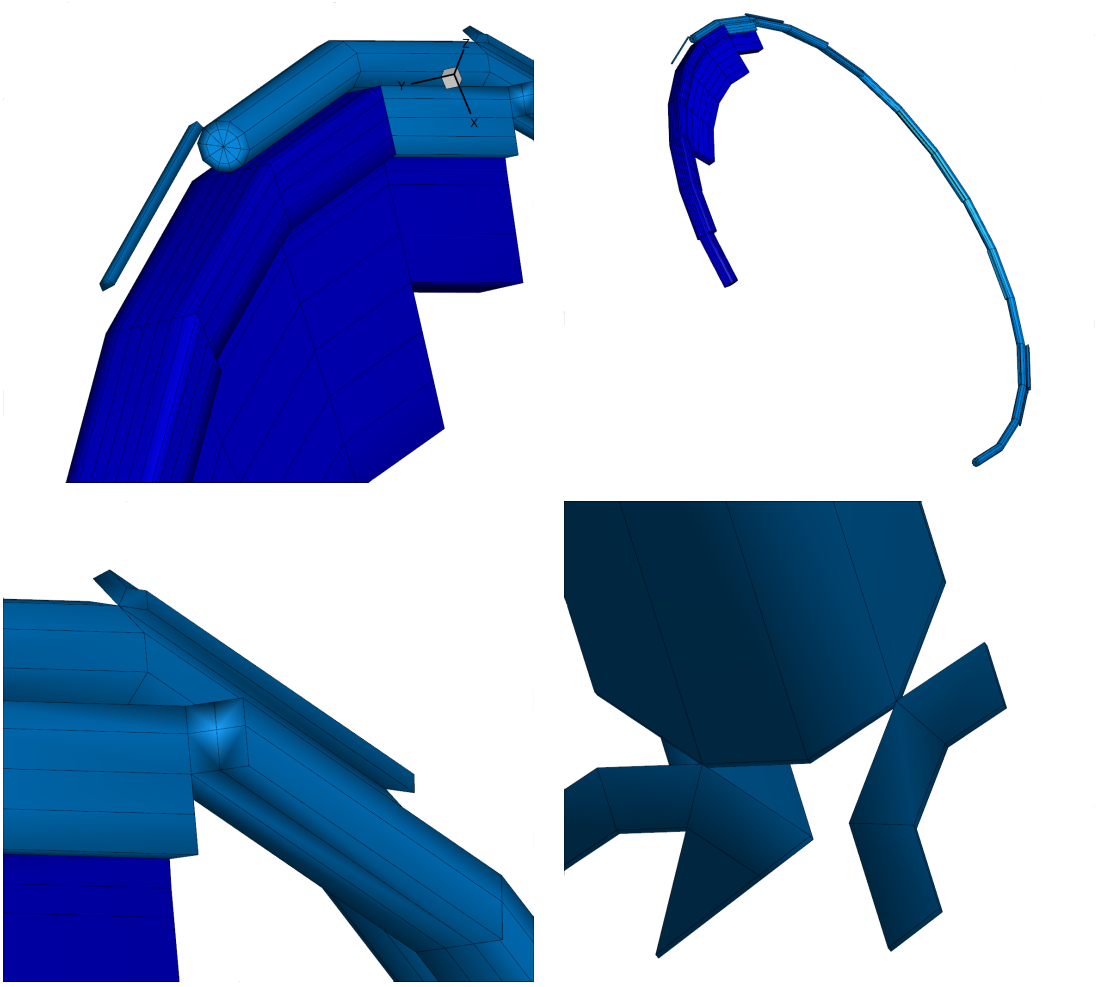


Figure 5.2: A pair of substructures of the mining reel problem from Figure 7.8, obtained from the automatic decomposition by METIS 4.0. We see that one of the substructures has 4 spurious rigid body modes.

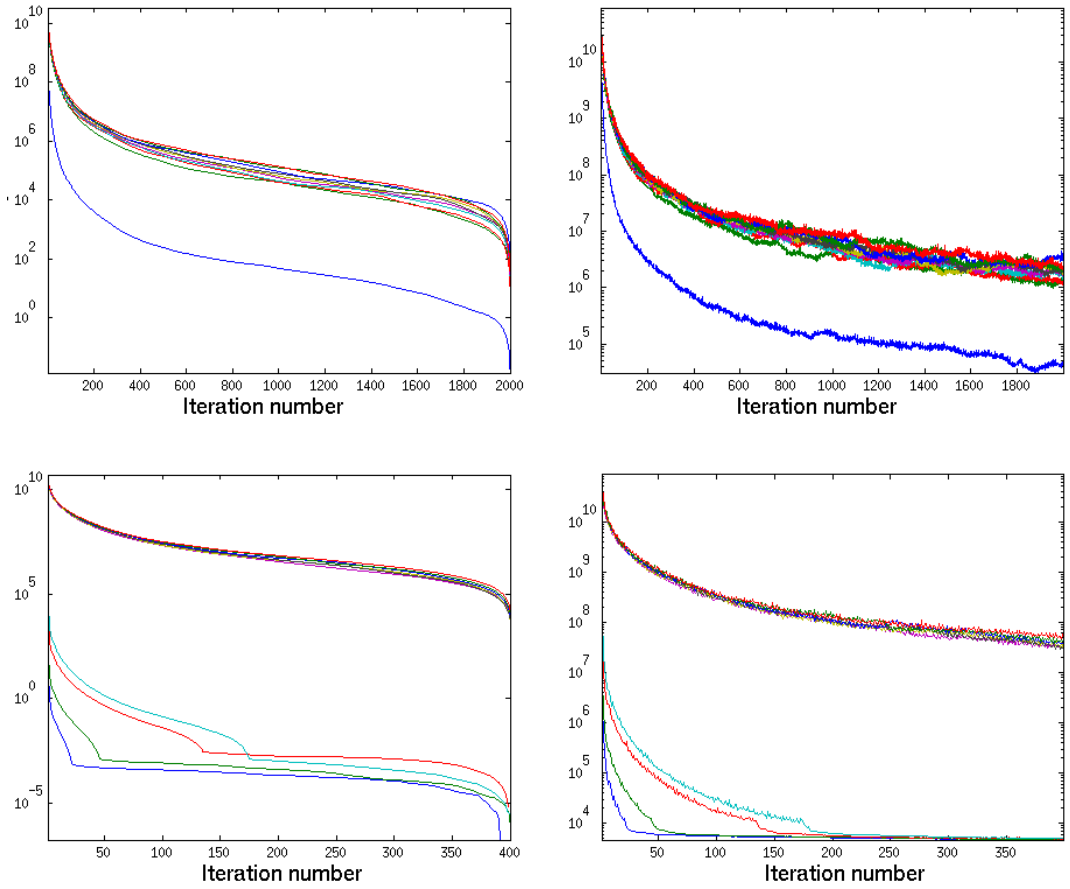
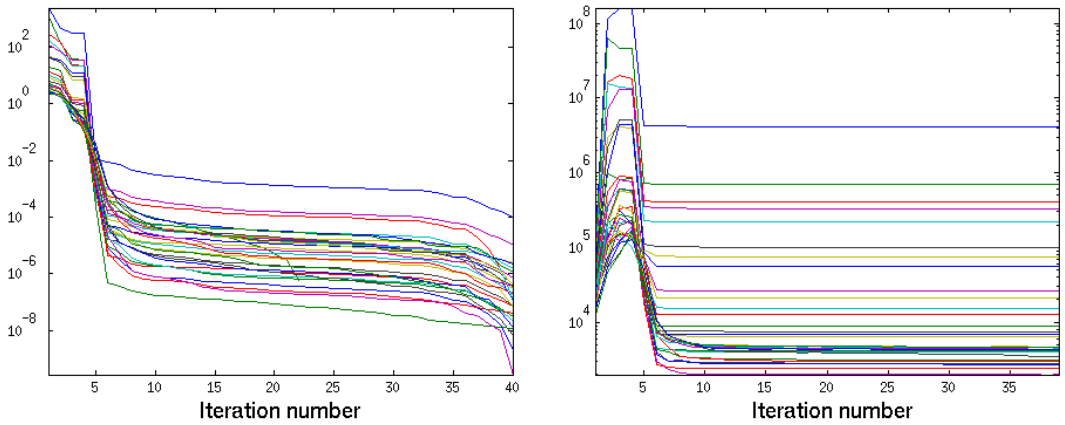


Figure 5.3: Comparison of the non-preconditioned (top) vs. preconditioned LOBPCG (bottom) for the detection of spurious rigid-body modes of the pair of subdomains from Fig. 5.2. Estimated eigenvalue errors are in the panels on the left-hand side, and Euclidean norms of residuals for different eigenpairs are in the panels on the right-hand side. We see that LOBPCG without a preconditioner essentially did not detect the nullspace, and the application of preconditioner led to relatively fast detection of the nullspace.



.

Figure 5.4: Convergence of the preconditioned LOBPCG for the pair of subdomains from Fig. 5.2 with the nullspace detection (Fig. 5.3). Estimated eigenvalue errors are in the panels on the left-hand side, and Euclidean norms of residuals for different eigenpairs are in the panels on the right-hand side.

6. Adaptive – Multilevel BDDC

We build on the previous two chapters to propose a new variant of the Multilevel BDDC with adaptive selection of constraints on each level. The starting point is Lemma 4.4, formulated as a multilevel analogy to Theorem 5.2.

Theorem 6.1 *The condition number bound $\kappa \leq \omega$ of Multilevel BDDC from Algorithm 4.1 satisfies*

$$\omega = \Pi_{i=1}^{L-1} \omega_i \ , \tag{6.1}$$

where

$$\omega_i = \sup_{w \in \widetilde{W}_i} \frac{\|(I - P_i)\, E_i w_i\|_a^2}{\|w_i\|_a^2} = \sup_{w \in \widetilde{W}_i} \frac{\|(I - (I - P_i)\, E_i)\, w_i\|_a^2}{\|w_i\|_a^2}.$$

The development of adaptive selection of constraints in Multilevel BDDC now proceeds similarly as in Chapter 5. We formulate (6.1) as a set of eigenvalue problems for each decomposition level. On each level we solve for every two adjacent substructures a generalized eigenvalue problem and we add the constraints to the definitions of $\widetilde{W}_i$.

The heuristic *condition number indicator* is defined as

$$\widetilde{\omega} = \Pi_{i=1}^{L-1} \max \left\{ \omega_i^{st} : \Omega_i^s \text{ and } \Omega_i^t \text{ are adjacent} \right\}. \tag{6.2}$$

We now describe the *Adaptive – Multilevel BDDC* in more detail. The algorithm consists of two main steps: (i) setup (adaptive selection of constraints), and (ii) loop of the preconditioned conjugate gradients with the Multilevel BDDC from Algorithm 4.1 as a preconditioner. The setup can be summarized as follows (cf. [78, Algorithm 4] for the 2D case):

Algorithm 6.2 (Setup of Adaptive – Multilevel BDDC) *Adding of coarse degrees of freedom to guarantee that the condition number indicator* $\widetilde{\omega} \leq \tau^{L-1}$, *for a given a target value* τ*:*

for levels $i = 1 : L - 1$,

Create substructures with roughly the same numbers of degrees of freedom (one can use a graph partitioner, e.g., METIS 4.0 [34]).

Find a set of initial constraints (in particular sufficient number of corners), and set up the BDDC structures for the adaptive algorithm (the next loop over faces).

for all faces $\mathcal{F}_i$ ***on level*** i,

Compute the largest local eigenvalues and corresponding eigenvectors, until the first m^{st} *is found such that* $\lambda^{st}_{m^{st}} \leq \tau$.

Compute the constraint weights and update the global coarse degrees of freedom selection matrix.

end.

Setup the BDDC structures for level i *and check size of the coarse problem: if small enough, call this level* L *problem, factor it directly, break the loop.*

end.

6.1 Implementation remarks

The matrices of the averaging operator E were constructed with entries proportional to the diagonal entries of the substructure matrices before elimination of interiors, which is also known as a *stiffness scaling* [39].

6.1.1 Initial constraints

Following Remark 5.1, in order to satisfy the polylogarithmic condition number bounds, we have used corners, and in 3D corners with arithmetic averages over edges as initial constraints. It is essential (Assumption 5.6) to generate a sufficient number of corners as initial constraints to prevent rigid body motions between any pair of adjacent substructures. This topic has been addressed in the literature several times cf., e.g., [9, 50] in a different context, or a recent contribution in the context of BDDC by Burda et al. [10]. The selection of corners in our implementation follows the original implementation by Dohrmann [17]. Let $\mathcal{N}^{st}$ denote the set of all nodes shared by substructures Ω^s and Ω^t. The first corner $c_1^{st} \in \mathcal{N}^{st}$ is chosen as a node shared by the largest number of substructures. The second corner $c_2^{st} \in \mathcal{N}^{st}$ is chosen as a node with greatest distance from c_1^{st}. For problems in three dimensions, a third corner $c_3^{st} \in \mathcal{N}^{st}$ is chosen as a node for which the area of the triangle connecting c_1^{st}, c_2^{st}, and c_3^{st} is maximized from vector cross product. However if all nodes in $\mathcal{N}^{st}$ are shared only by the two substructures, the algorithm starts by a random selection of an initial node in $\mathcal{N}^{st}$ and c_1^{st} is identified as a node maximizing the distance from the initial node, as suggested in [10].

6.1.2 Algebraic coarse elements

The substructures in engineering applications were obtained using a graph partitioner METIS 4.0 [34]. The connectivity graph has been weighted in both vertices and edges in order to minimize the number of "cuts". The vertex weights were given by the total number of degrees of freedom in the substructure and the weights of graph edges were determined as the numbers of the degrees of freedom identified on faces by the adaptive algorithm.

The substructures on higher levels were then treated in an algebraic way, unlike the geometric substructures illustrated in Figure 3.2, as (coarse) elements with energy minimal basis functions. The routines for multilevel algorithm must allow for (coarse) elements with variable number of nodes, and they must also allow for variable number of degrees of freedom on each node (corresponding to a face) – the number of nodes and the number of their degrees of freedom is a-priori unknown due to their adaptive selection. For this purpose we transform, renumber and reorder all globs so that the nodes corresponding to corners have the lowest numbers, followed by nodes corresponding to edges, and finally by nodes corresponding to faces identified by the adaptive algorithm.

It is also convenient to use the same assembling routines for higher levels. To this end, the globs that do not consist of single nodes (edges or faces) are replaced by "virtual" nodes with coordinates given by arithmetic averages of coordinates of all nodes belonging to that particular glob. We note that the coordinates of these nodes are in fact not used by the algorithm, so the only purpose is to allow the use of the basic two-level routines without any modifications.

Finally, we remark that instead of interior pre- and post-correction on the lowest decomposition level, cf. equations (4.6)-(4.7) and (4.12)-(4.13), we reduce the problem to interfaces in the pre-processing step, cf. also Remark 3.12.

6.1.3 Adaptive constraints

The adaptive algorithm uses matrices and operators that are readily available in an implementation of the BDDC method with an explicit coarse space, with one exception: in order to satisfy the local partition of unity, cf. [62, eq. (9)],

$$E_i^{st} R_i^{st} = I,$$

we need to generate locally the weight matrices E_i^{st}.

In the computations reported in Chapter 7, we drop the adaptively generated edge constraints in 3D. Then, it is no longer guaranteed that the condition number indicator $\widetilde{\omega} \leq \tau^{L-1}$. However, the method is still observed to perform well. Since the constraint weights are thus supported only on faces, and the entries corresponding to edges are set to be zero, we orthogonalize and normalize the vectors of constraint weights to preserve numerical stability.

7. Numerical Examples

The main purpose of this chapter is to compare performance of the standard two-level BDDC with the adaptive and multilevel extensions. For consistency with our previous research [61, 65, 79], we first present results for several academic examples. Results of Multilevel BDDC for scalar problems can be found in [64]. The computations were done in Matlab (version 7.8.0.347 (R2009a)). The generalized eigenproblems on pairs of substructures were solved by setting up the matrices and using standard methods for the symmetric eigenvalue problem in Matlab, and we have also tested LOBPCG by Knyazev [45] with a preconditioner described in Section 5.1. The two-dimensional results are reproduced from [78], the three-dimensional results appear here for the first time.

7.1 Two-dimensional results

The method has been tested on planar elasticity (with $\lambda = 1$, $\mu = 2$) on a square domain discretized by Lagrange bilinear finite elements with 1 182 722 degrees of freedom. The domain was decomposed into $48 \times 48 (= 2\,304)$ subdomains on the second level and into $3 \times 3 (= 9)$ subdomains on the third-level. Such a decomposition leads to the coarsening ratio $H_i/H_{i-1} = 16$, for $i = 1, 2$. In order to test the adaptive selection of constraints, one single edge is jagged on both levels, see Fig. 7.1. Recall that edges in 2D are regarded as faces.

In the first set of experiments, we have compared performance of the non-adaptive BDDC method with 2 and 3 decomposition levels. The results are presented in Tables 7.1 and 7.2. As suggested by Lemma 4.4, the convergence of the algorithm deteriorates when additional levels are introduced.

In the next set of experiments, we have tested the adaptive algorithm for the two-level BDDC. The results are summarized in Table 7.5. The algorithm performs consistently with our previous formulation in [61]. The eigenvalues associated with edges between substructures clearly distinguish between the single problematic edge and the others (Table 7.3). Adding the coarse dofs created from the associated eigenvectors decreases the value of the condition number indicator $\widetilde{\omega}$ and improves convergence at the cost of increasing the number of coarse dofs.

Finally, we have tested the performance of the Adaptive – Multilevel BDDC for the model problem with three-level decomposition (Fig. 7.1). Because the number of coarse degrees of freedom depends on a-priori chosen value of τ and the coarse basis functions on level i become shape basis functions on level $i+1$, the solutions of local eigenvalue problems will depend on τ as well. This fact is illustrated by Table 7.4 for $\tau = 2$, and $\tau = 10$ (the local eigenvalues for $\tau = 3$ were essentially same as for $\tau = 2$). Comparing the values in the two panels of this table, we see that lower values of τ result in worse conditioning of the local eigenvalue problems on higher decomposition level. This immediately raises the conjecture that it might not be desirable to decrease the values of τ arbitrarily low in order to achieve a better convergence of the method. On the other hand, for the model problem, comparing the convergence results for the two-level method (Table 7.5) with the three-level method (Table 7.6), we see that with the adaptive constraints we were able to achieve nearly the same convergence properties of both methods.

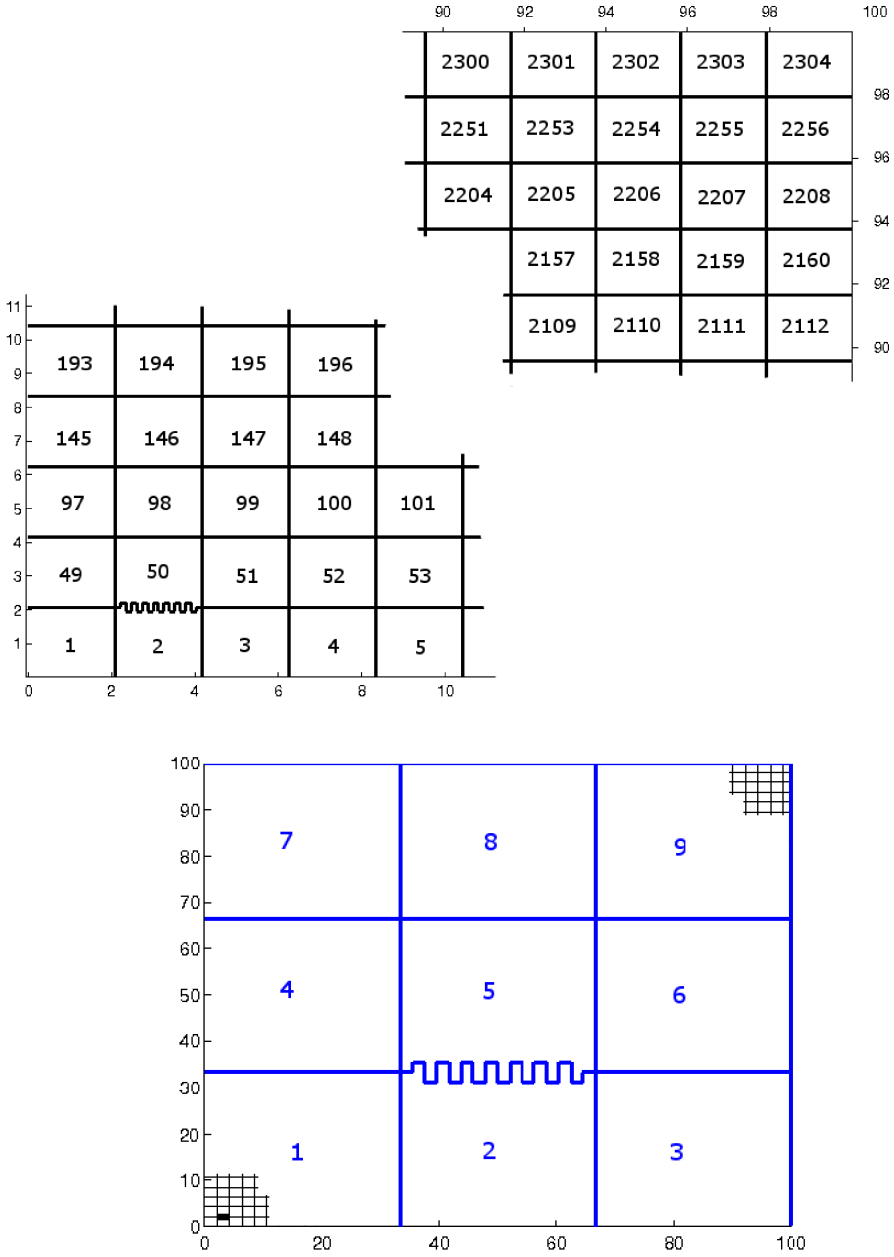


Figure 7.1: Two remote corners of the two-level decomposition into 48×48 ($= 2304$) subdomains (top), and the decomposition into 9 subdomains for the three-level method (bottom). The jagged edge from the lower decomposition level (top) is indicated on the second-level decomposition (bottom) by the thick line.

Table 7.1: Results for the planar elasticity from Fig. 7.1 obtained using non-adaptive 2-level method. Constraints are corners, or corners and arithmetic averages over edges, denoted as c, c+f, resp., and Nc is number of constraints (coarse degrees of freedom), $\mathcal{C}$ is size of the coarse problem related to size of a subdomain problem, κ is the approximate condition number computed from the Lanczos sequence in conjugate gradients, and it is the number of iterations for relative residual tolerance 10^{-8}.

constraint	Nc	$\mathcal{C}$	κ	it
c	4794	9.3	18.41	43
c+f	13818	26.9	18.43	32

Table 7.2: Results for the planar elasticity from Fig. 7.1 obtained using non-adaptive 3-level method. Nc is the number of coarse degrees of freedom on the first (+ the second) decomposition level, $\mathcal{C}$ is the relative coarsening with respect to the size of substructures on the first level (the size of the coarse problem for the three-level method is negligible). Other headings are the same as in Table 7.1.

constraint	Nc	$\mathcal{C}$	κ	it
c	$4794 + 24$	1.0	67.5	74
c+f	$13818 + 48$	3.0	97.7	70

Table 7.3: The largest eigenvalues $\lambda_{st,k}$ of the local eigenvalue problems for several pairs of subdomains s and t of the 2-level elasticity problem from Fig. 7.1 (top), with $(s,t) = (2,50)$ being the jagged edge.

s	t	$\lambda_{st,1}$	$\lambda_{st,2}$	$\lambda_{st,3}$	$\lambda_{st,4}$	$\lambda_{st,5}$	$\lambda_{st,6}$	$\lambda_{st,7}$	$\lambda_{st,8}$
1	2	3.8	2.4	1.4	1.3	1.2	1.1	1.1	1.1
1	49	6.0	3.5	2.7	1.4	1.3	1.1	1.1	1.1
2	3	5.4	2.6	1.6	1.3	1.2	1.1	1.1	1.1
2	50	24.3	18.4	18.3	16.7	16.7	14.7	13.5	13.1
3	4	3.4	2.4	1.4	1.3	1.1	1.1	1.1	1.1
3	51	7.4	4.6	3.7	1.7	1.4	1.3	1.2	1.1
49	50	12.6	5.1	4.3	1.9	1.6	1.3	1.2	1.2
50	51	8.7	4.8	3.9	1.8	1.5	1.3	1.2	1.2
50	98	7.5	4.6	3.7	1.7	1.4	1.3	1.2	1.1

Table 7.4: The largest eigenvalues $\lambda_{st,k}$ of the local problems for several pairs of subdomains s, t on the level $i = 2$, cf. Fig. 7.1 (lower panel), with $\tau = 2$ (top) and with $\tau = 10$ (bottom). The jagged edge is between subdomains 2 and 5.

s	t	$\lambda_{st,1}$	$\lambda_{st,2}$	$\lambda_{st,3}$	$\lambda_{st,4}$	$\lambda_{st,5}$	$\lambda_{st,6}$	$\lambda_{st,7}$	$\lambda_{st,8}$
1	2	16.5	9.0	5.4	2.6	2.1	1.4	1.3	1.3
1	4	6.5	4.7	1.9	1.7	1.3	1.2	1.2	1.1
2	3	23.1	9.4	4.6	3.2	2.1	1.6	1.4	1.3
2	5	84.3	61.4	61.4	55.9	55.8	49.3	48.0	46.9
3	6	13.7	8.8	4.4	2.2	1.9	1.4	1.3	1.2
4	7	6.5	4.7	1.9	1.7	1.3	1.2	1.2	1.1
5	6	18.9	13.1	11.3	3.8	2.6	2.1	1.9	1.5
5	8	17.3	12.9	10.8	3.6	2.3	2.0	1.8	1.4
8	9	13.7	8.8	4.4	2.2	1.9	1.4	1.3	1.2
1	2	7.7	4.5	2.7	1.6	1.4	1.2	1.2	1.1
1	4	3.6	3.0	1.5	1.5	1.2	1.2	1.1	1.1
2	3	10.9	4.8	2.7	1.7	1.5	1.2	1.2	1.1
2	5	23.2	17.2	13.7	13.7	12.7	12.4	11.0	10.9
3	6	6.1	4.2	2.5	1.5	1.3	1.2	1.1	1.1
4	7	3.6	3.0	1.5	1.5	1.2	1.2	1.1	1.1
5	6	9.8	6.2	4.1	2.1	1.6	1.5	1.3	1.2
5	8	8.6	5.9	3.9	2.0	1.5	1.4	1.2	1.2
8	9	6.1	4.2	2.5	1.5	1.3	1.2	1.1	1.1

Table 7.5: Results for the planar elasticity from Fig. 7.1 obtained using the adaptive 2-level method. Here, τ is condition number target, $\widetilde{\omega}$ is condition number indicator, and the other headings are the same as in Table 7.1.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c)	4794	9.3	-	18.41	43
10	4805	9.4	8.67	8.34	34
3	18110	35.3	2.67	2.44	15
2	18305	35.7	1.97	1.97	13

Table 7.6: Results for the planar elasticity from Fig. 7.1 obtained using the adaptive 3-level method. Headings are the same as in Tables 7.2 and 7.5.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c)	$4794 + 24$	1.0	-	67.5	74
10	$4805 + 34$	1.0	$> (9.80)^2$	37.42	60
3	$18110 + 93$	3.9	$> (2.95)^2$	3.11	19
2	$18305 + 117$	4.0	$> (1.97)^2$	2.28	15

7.2 Three-dimensional results

The performance of the Adaptive BDDC in the presence of jumps in material coefficients has been tested on a cube with material parameters $E = 10^6$ Pa and $\nu = 0.45$ (rubber), penetrated by four bars with parameters $E = 2.1{\cdot}10^{11}$ Pa and $\nu = 0.3$ (steel), consisting of 107 811 degrees of freedom, and distributed into 8 substructures with 30 corners, 16 edges and 15 faces, see Fig. 7.2, see also [65]. Comparing the results in Tables 7.8 and 7.9 we see that with $\tau = 10\,000$ only 10 additional averages over faces decrease the number of iterations nearly 3 times, and with $\tau = 2$ the number of iterations decreased more than 13 times compared to the non-adaptive algorithm with arithmetic averages over all globs (c+e+f) whereas the number of constraints increased approximately 2.5 times. Results (and numbers of nonzeros in the action) of the BDDC preconditioner in Table 7.9 can be compared with results obtained by incomplete Cholesky factorization applied to the global matrix in Table 7.7. We see that for a lower number of iterations the fill-in of the Cholesky factor was quite high when compared with the fill-in of the subdomain and the coarse problems in the BDDC method.

The performance of the Adaptive – Multilevel BDDC in the presence of jumps in material coefficients has been tested on a cube designed similarly as the problem above (also with the same material parameters), this time consisting of 823 875 degrees of freedom and distributed into 512 substructures, 721 corners, 1 176 edges and 1 344 faces on the first decomposition level, and 4 substructures, 6 corners, one edge and 4 faces on the second decomposition level, see Fig. 7.3.

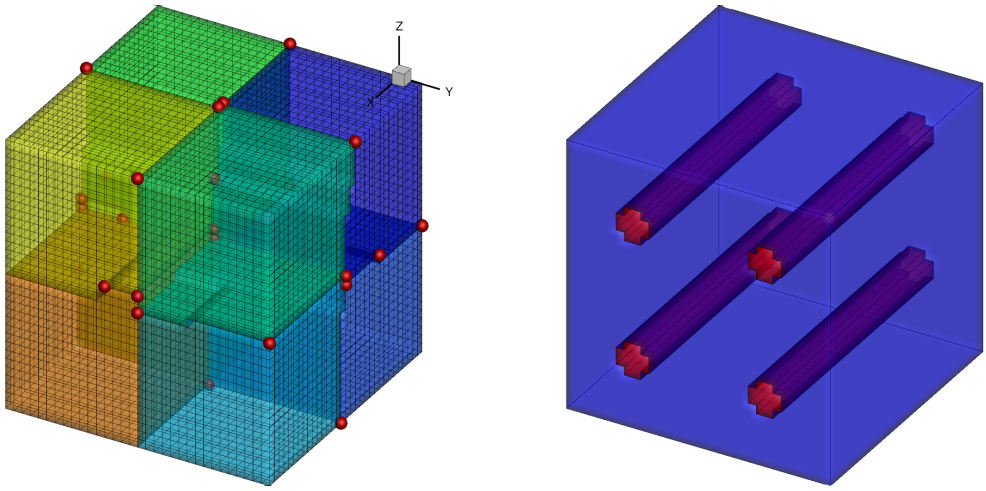


Figure 7.2: Finite element discretization and substructuring of the cube with jumps in coefficients, consisting of 107 811 degrees of freedom, distributed into 8 substructures with 30 corners, 16 edges and 15 faces (the bars cut the substructures only through faces). Notably, similar problems are solved in practice to determine numerically (anisotropic) properties of composite materials [72]. Courtesy of Jakub Šístek.

Table 7.7: Results for the cube from Fig. 7.2 obtained using a preconditioning by incomplete Cholesky factorization. The global stiffness matrix has size 107 811 with 7 737 641 nonzeros. Here, nnz(R) is the number of nonzeros in the upper triangular Cholesky factor R, fill-in is the relative fill-in computed as 2 times nnz(R) divided by the number of nonzeros in the global stiffness matrix, κ is the approximate condition number computed from the Lanczos sequence in conjugate gradients, *it* is number of iterations for relative residual tolerance 10^{-8}. With the zero-level of fill-in the method did not converge.

drop tol.	nnz(R)	fill-in	cond	iter
no fill-in	3 922 726	1.01	-	∞
$1 \cdot 10^{-3}$	9 784 734	2.53	$4.14 \cdot 10^{6}$	331
$1 \cdot 10^{-4}$	30 968 534	8.00	$2.25 \cdot 10^{6}$	170
$1 \cdot 10^{-5}$	88 125 845	22.78	119.12	37
$1 \cdot 10^{-6}$	194 448 707	50.26	3.63	15
$1 \cdot 10^{-7}$	273 649 916	70.73	1.88	10

Table 7.8: Results for the cube from Fig. 7.2 obtained using the non-adaptive 2-level method. Constraints are corners, or corners and arithmetic averages over edges and faces denoted as c, c+e, c+e+f resp., and c+e+f (3eigv), corresponding to corner constraints, arithmetic averages, and 3 weighted averages over each face obtained using the adaptive method. Next, Nc is the number of constraints, $\mathcal{C}$ is the size of the coarse problem related to size of a subdomain problem, κ is the approximate condition number computed from the Lanczos sequence in conjugate gradients, it is number of iterations for relative residual tolerance 10^{-8}.

constraint	Nc	κ	it
c	90	408 114	455
c+e	138	125 378	307
c+e+f	183	18 915.1	211
c+e+f (3eigv)	183	1 267.61	81

Table 7.9: Results for the cube from Fig. 7.2 obtained using the adaptive 2-level method. Here, τ is the condition number target, $\widetilde{\omega}$ is the condition number indicator. An approximate number of nonzeros of the Cholesky factor of a substructure problem is 8 500 000 for all values of τ, and the number of nonzeros in the Cholesky factor of the coarse problem is denoted by nnz(c). The other headings are the same as in Table 7.8.

τ	Nc	nnz(c)	$\widetilde{\omega}$	κ	it
∞(=c+e)	138	6 618	268 390.00	125 378.00	307
10 000	148	7 402	5 096.07	1 843.70	104
1 000	159	8 271	368.78	173.57	38
100	162	8 448	5.94	6.42	24
5	198	13 029	4.99	4.55	21
2	465	87 012	< 2	2.80	16

Comparing the results in Tables 7.10 and 7.11 we see that similar to the previous problem, a relatively small number of (additional) constraints leads to a dramatic decrease in number of iterations in the 2-level method. When the non-adaptive 2-level is replaced by the 3-level method, Tables 7.10 and 7.12, the condition number estimate as well as the number of iterations grows. However, with the adaptive 3-level approach (Table 7.13) we were able to achieve nearly the same convergence properties as in the adaptive 2-level method (Table 7.11).

Table 7.10: Results for the cube from Fig. 7.3 obtained using the non-adaptive 2-level method. The headings are the same as in Table 7.8.

constraint	Nc	κ	it
c	2 163	312 371	$> 3\,000$
c+e	5 691	45 849	1 521
e+e+f	9 723	16 384	916
c+e+f (3eigv)	9 723	3 848	367

Table 7.11: Results for the cube from Fig. 7.3 obtained using the adaptive 2-level method. The headings are the same as in Table 7.9.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c+e)	5 691	3.54	$o(10^4)$	45 848.60	1 521
10 000	5 883	3.66	8 776.50	5 098.60	441
1 000	6 027	3.75	5.33	9.92	32
10	6 149	3.82	6.25	6.66	28
5	9 119	5.67	< 5	4.79	24
2	25 009	15.54	< 2	2.92	18

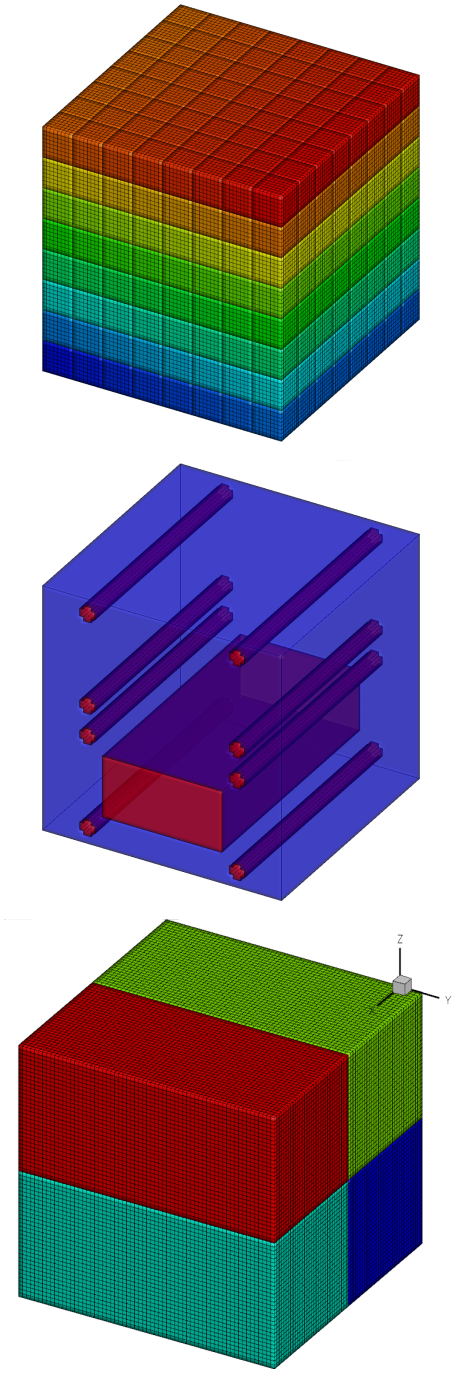


Figure 7.3: Finite element discretization and substructuring of the large cube with jumps in coefficients, consisting of 823 875 degrees of freedom, distributed into 512 substructures with 721 corners, 1 176 edges and 1 344 faces on the first decomposition level (top), and 4 substructures, 6 corners, one edge and 4 faces on the second decomposition level (bottom). Courtesy of Jakub Šístek.

Table 7.12: Results for the cube from Fig. 7.3 obtained using the non-adaptive 3-level method. The headings are the same as in Tables 7.2 and 7.8.

constraint	Nc	$\mathcal{C}$	κ	it
c	$2\,163 + 18$	$0.34 + 0.01$	$o(10^7)$	$>> 3\,000$
c+e	$5\,691 + 21$	$0.88 + 0.01$	$o(10^6)$	$> 3\,000$
c+e+f	$9\,723 + 33$	$1.51 + 0.02$	461 750	1 573
c+e+f (3eigv)	$9\,723 + 33$	$1.51 + 0.02$	125 305	981

Table 7.13: Results for the cube from Fig. 7.3 obtained using the adaptive 3-level method. The headings are the same as in Tables 7.6 and 7.9.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c+e)	$5\,691 + 21$	$0.88 + 0.01$	-	$o(10^6)$	$> 3\,000$
10 000	$5\,883 + 28$	$0.91 + 0.02$	8 776.50	26 874.40	812
1 000	$6\,027 + 34$	$0.94 + 0.02$	766.82	1 449.50	145
100	$6\,027 + 53$	$0.94 + 0.03$	99.05	100.89	59
10	$6\,149 + 65$	$0.96 + 0.04$	7.93	7.91	30
5	$9\,119 + 67$	$1.42 + 0.04$	< 5	6.18	25
2	$25\,009 + 122$	$3.89 + 0.08$	< 2	3.08	18

7.2.1 Applications to engineering problems

7.2.1.1 Application of Multilevel BDDC to a dam

The performance of the Multilevel BDDC has been tested on the realistic engineering problem of a dam discretized using 3 800 080 tetrahedral finite elements with 668 916 nodes and 2 006 748 degrees of freedom, with two variants of substructuring: first decomposed into 400 substructures with 3 990 corners, 3 070 edges and 2 274 faces, see Fig. 7.4, and then decomposed into 1 024 substructures with 10 693 corners, 7 713 edges and 6 182 faces, see Fig. 7.6.

The results with non-adaptive constraints for the decomposition into 400 substructures are summarized in Tables 7.14 and 7.15 for the two- and three-level methods, respectively. Results with non-adaptive constraints for the decomposition into 1 024 substructures are summarized in Tables 7.16 and 7.17 for the two- and three-level methods, respectively. At the first glance, comparing the values in Tables 7.14 and 7.15, it might appear that for the decomposition into 400 substructures, the increase in the number of iterations is not too significant if one uses corners and arithmetic averages over edges (or faces). However, comparing the values in Tables 7.16 and 7.17, it turns out that for the decomposition into 1 024 substructures the number of iterations needed for the 3-level method double, or even triple, compared to the 2-level method.

Nevertheless, we see that for this problem the simple arithmetic averages already work well enough as there are no interfaces that require extra work - the quality of the decomposition is uniform, as seen in Figures 7.4–7.7.

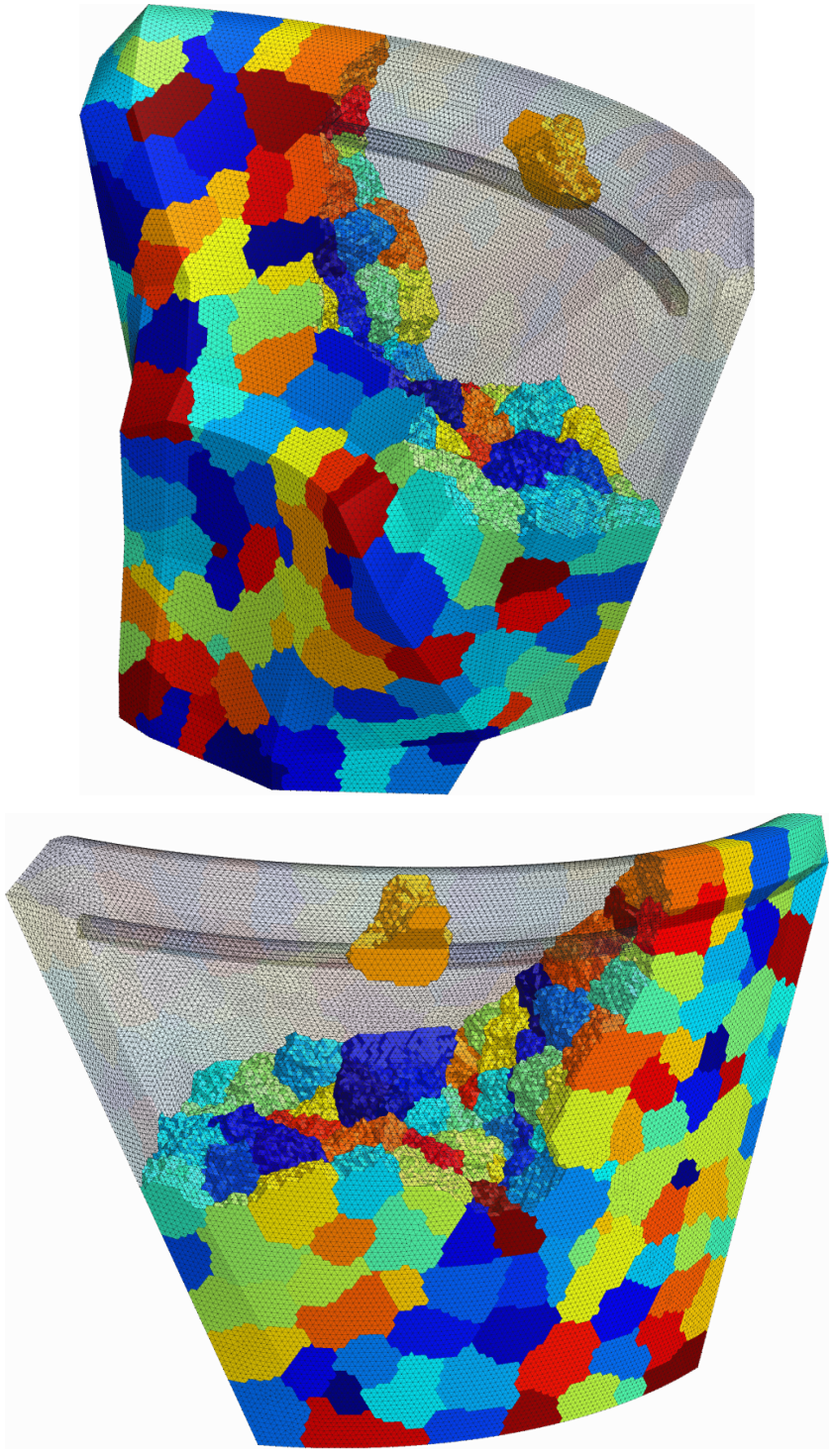


Figure 7.4: Finite element discretization and substructuring of the dam, consisting of 2 006 748 degrees of freedom, distributed into 400 substructures with 3 990 corners, 3 070 edges and 2 274 faces. Courtesy of Jaroslav Kruis.

Table 7.14: Results for the dam (Fig. 7.4, 400 substructures) obtained using the non-adaptive 2-level method. The headings are the same as in Table 7.8.

constraint	Nc	$\mathcal{C}$	κ	it
c	11 970	2.39	34.82	63
c+e	21 180	4.22	18.91	41
c+e+f	28 002	5.58	5.73	24

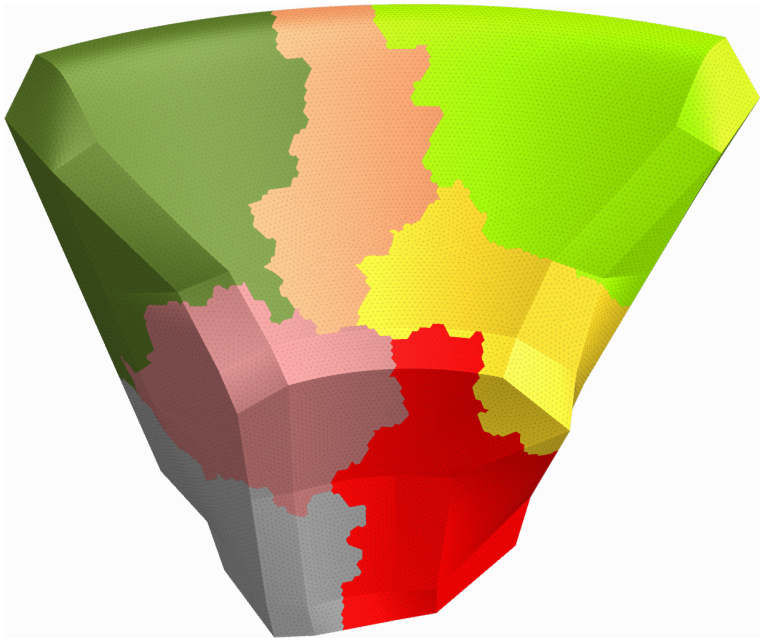


Figure 7.5: Correspondence of finite elements and the subdomains on the second decomposition level. The dam problem with 3 800 080 finite elements, 400 substructures on the first level and 8 substructures on the second level.

Table 7.15: Results for the dam from Figs. 7.4 and 7.5 ($400+8$ substructures) obtained using the non-adaptive 3-level method. The headings are the same as in Tables 7.2 and 7.8.

constraint	Nc	nc/ne/nf	$\mathcal{C}$	κ	it
c	$11\,970+99$	33/15/15	$0.30+0.02$	99.11	87
c+e	$21\,180+144$	33/15/16	$0.53+0.03$	18.88	43
c+e+f	$28\,002+198$	33/15/18	$0.70+0.04$	9.92	32

Table 7.16: Results for the dam (Fig. 7.6, 1 024 substructures) obtained using the non-adaptive 2-level method. The headings are the same as in Table 7.8.

constraint	Nc	$\mathcal{C}$	κ	it
c	32079	16.37	28.53	54
c+e	55218	28.18	13.96	35
c+e+f	73764	37.64	5.00	21

Table 7.17: Results for the dam from Figs. 7.6 and 7.7 ($1\,024+32$ substructures) obtained using the non-adaptive 3-level method. The headings are the same as in Tables 7.2 and 7.8.

constraint	Nc	nc/ne/nf	$\mathcal{C}$	κ	it
c	$32079+768$	256/194/124	$0.51+0.39$	498.82	136
c+e	$55218+1407$	256/213/128	$0.88+0.71$	161.63	71
c+e+f	$73764+1818$	256/213/137	$1.18+0.93$	169.38	77

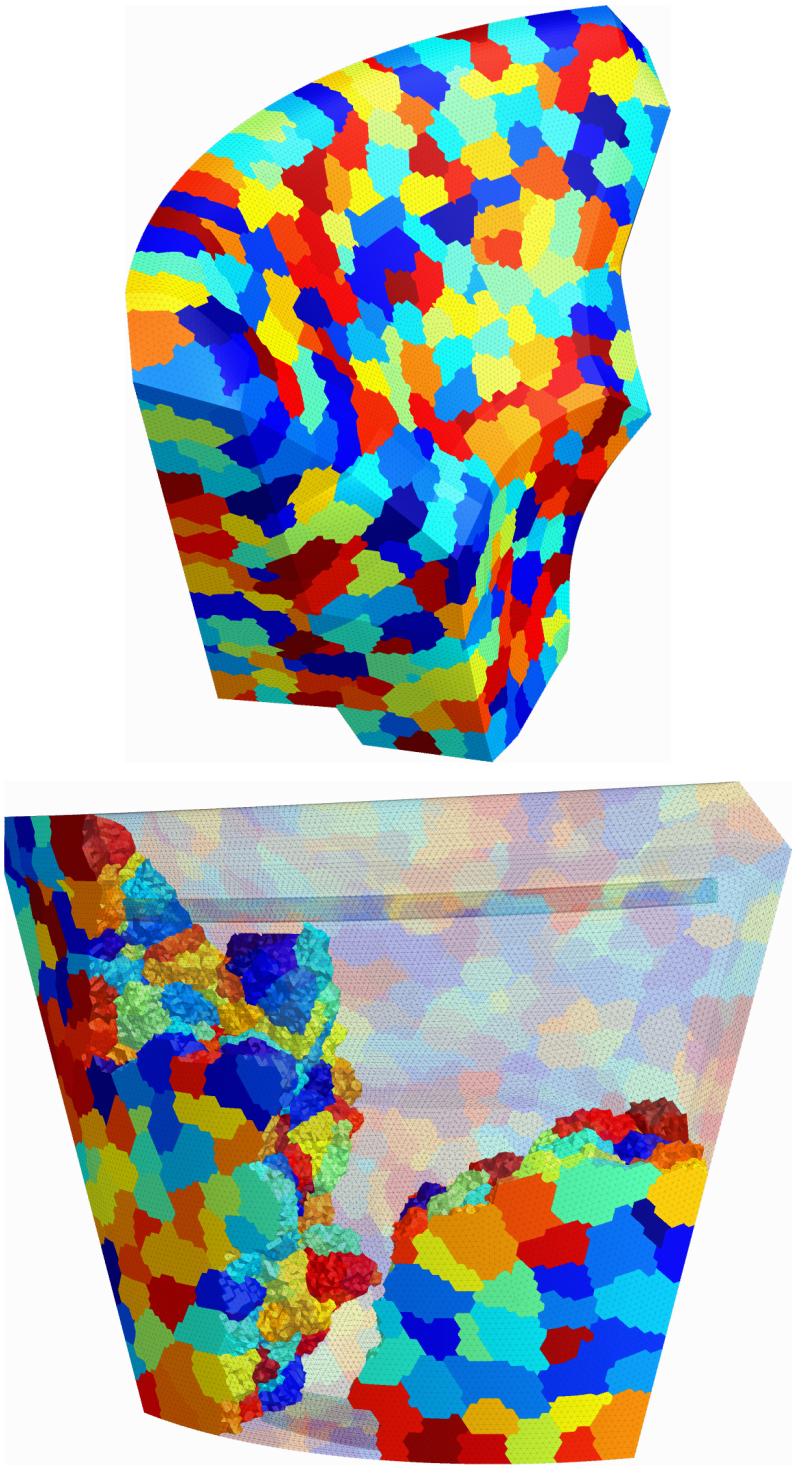


Figure 7.6: Finite element discretization and substructuring of the dam, consisting of 2 006 748 degrees of freedom, distributed into 1 024 substructures with 10 693 corners, 7 713 edges and 6 182 faces. Courtesy of Jaroslav Kruis.

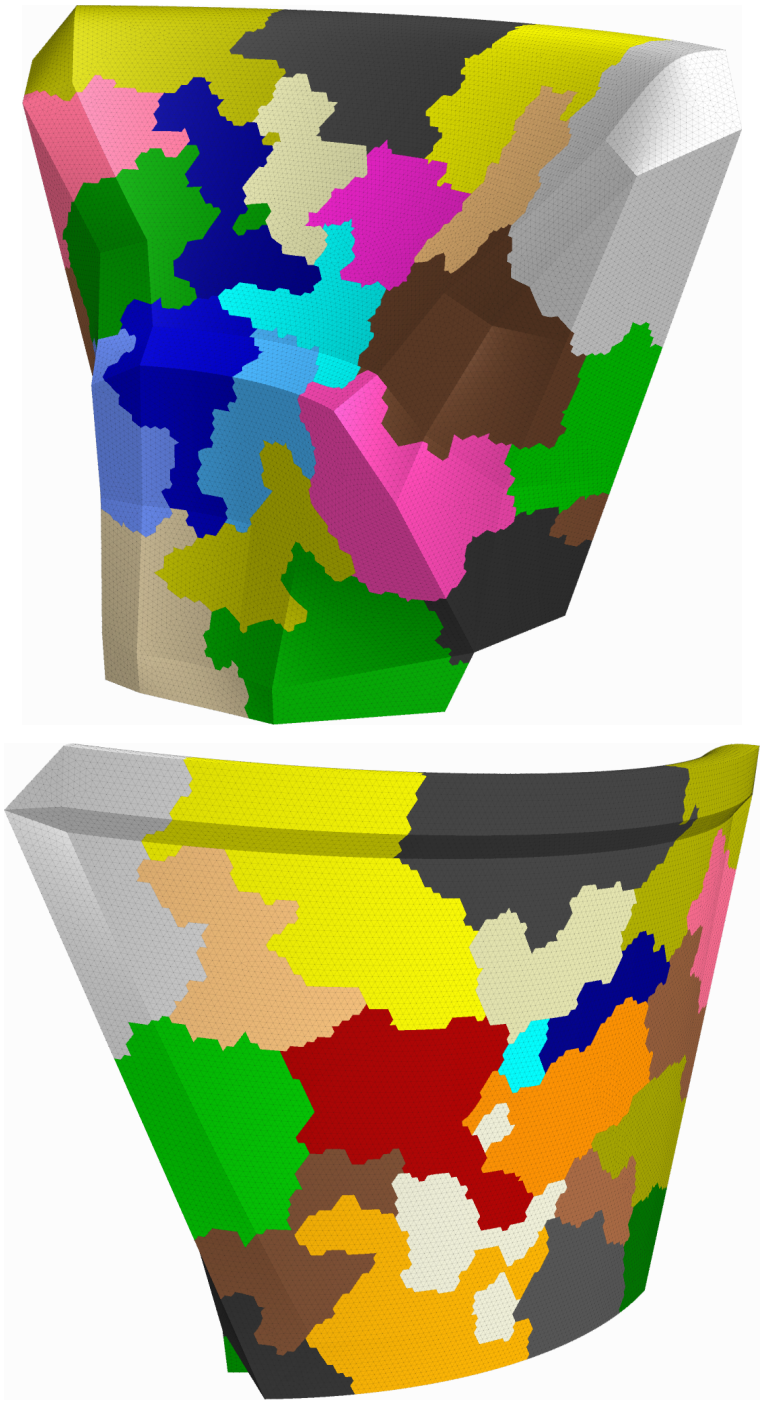


Figure 7.7: Correspondence of finite elements and the subdomains on the second decomposition level. The dam problem with 3 800 080 finite elements, 1 024 substructures on the first level and 32 substructures on the second level.

7.2.1.2 Application of Adaptive – Multilevel BDDC to a mining reel

The performance of Adaptive – Multilevel BDDC has been tested on another realistic engineering problem, modeling a mining reel. The computational mesh consists of 140 816 quadratic finite elements, 579 737 nodes and 1 739 211 degrees of freedom. We have tested two variants of decomposition. In the first variant, the mesh was distributed into 400 substructures with 4 010 corners, 831 edges and 1 906 faces on the first decomposition level (Fig. 7.8), and 8 substructures on the second decomposition level (Fig. 7.9). As may be seen in Figure 7.8, the main problem is substructuring of the steel rope. Subdomains (created by the graph partitioner METIS 4.0 [34]) are thin, long, and often contain spurious mechanisms, cf. also Fig. 5.2. Hence it is not a surprise that a standard, non-adaptive BDDC method with arithmetic averages over edges (and faces) fails. The performance of the adaptive algorithm for the two-level method can be seen in Table 7.18, and for the three-level method in Table 7.19. Comparing the values in these two tables we see that with the reasonably low values of the threshold τ, the convergence is essentially identical.

In the second variant of decomposition, the mesh was distributed into 1 024 substructures with 7 864 corners, 1 197 edges, and 3 895 faces on the first decomposition level (Fig. 7.10), and 32 subdomains on the second decomposition level (Fig 7.11). This decomposition is more adequate for the standard BDDC method (Table 7.20). However, comparing Tables 7.20 and 7.21 we see that the adaptive approach still allows for significant improvement in number of itera-

tions and moreover, convergence of the adaptive two- and three-level method (Tables 7.21 and 7.22) is, similar to the above, essentially identical.

We note that the observed approximate condition number κ computed from the Lanczos sequence in conjugate gradients is no longer close to the target condition number τ. However the algorithm is still observed to perform well.

Table 7.18: Results for the mining reel (Fig. 7.8, 400 substructures) obtained using the adaptive 2-level method. The headings are the same as in Table 7.9.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c+e)	14 523	-	$o(10^7)$	-	-
10 000	16 080	3.70	9 999.85	401 441.00	1 453
1 000	20 331	4.68	999.94	4 205.79	401
500	22 575	5.19	499.93	2 024.16	297
100	29 641	6.82	99.96	1 653.31	173
50	33 049	7.60	49.98	1 647.41	150
10	45 113	10.38	< 10	1 625.31	108
5	54 191	12.46	< 5	1 620.18	93
2	78 475	18.05	< 2	1 608.54	80

Table 7.19: Results for the mining reel from Figs. 7.8 and 7.9 (400 and 8 subdomains) obtained using teh adaptive 3-level method. The headings are the same as in Tables 7.6 and 7.9.

τ	Nc	nc/ne/nf	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
100	29 641 + 170	22/0/8	0.85/0.04	99.94^2	58 828.8	1 129
10	45 113 + 539	22/0/8	1.30/0.12	$< 10^2$	1 623.18	123
2	78 475 + 2 177	22/0/8	2.26/0.50	$< 2^2$	1 607.88	79

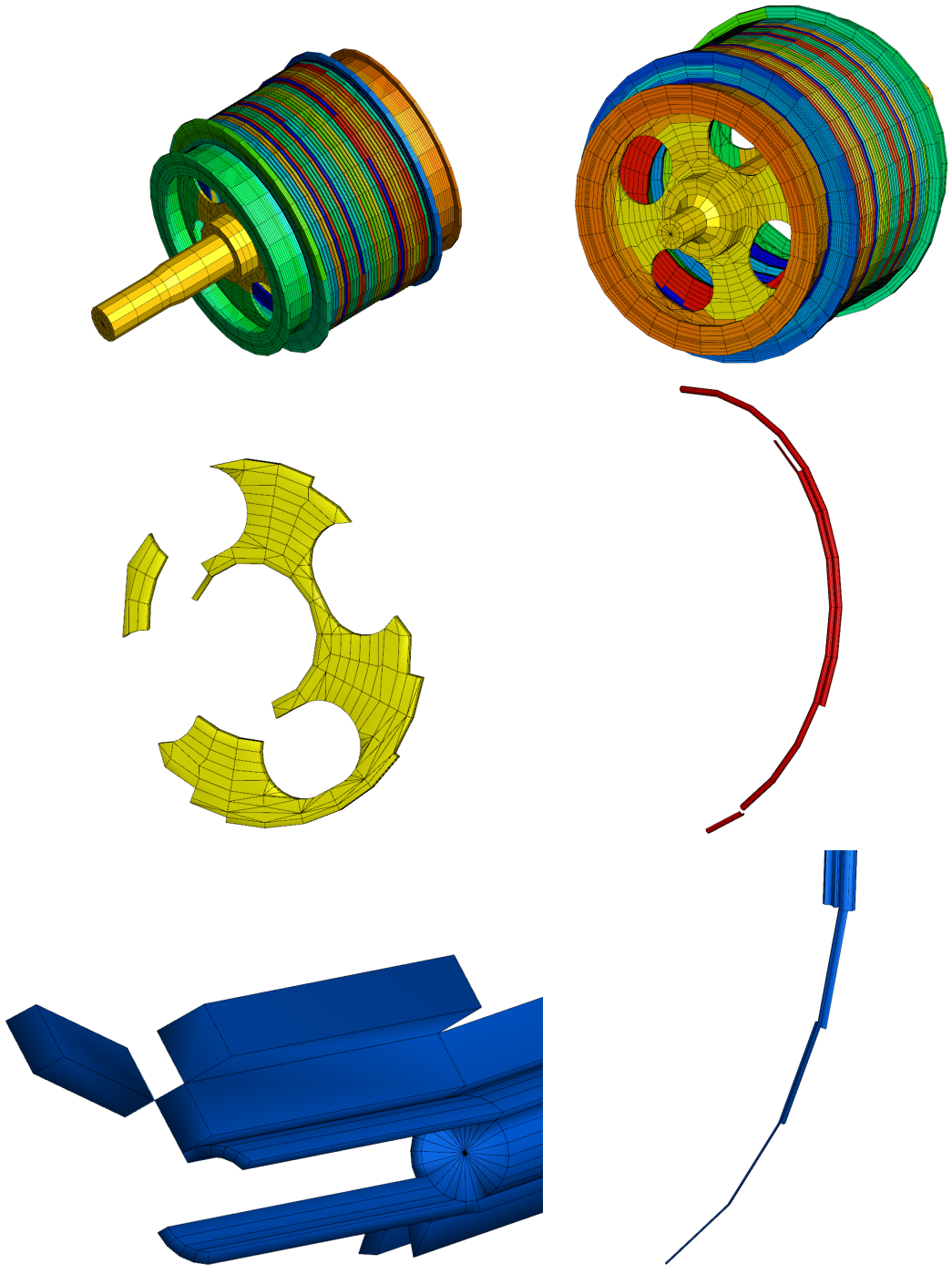


Figure 7.8: Finite element discretization and substructuring of the mining reel problem, consisting of 1 739 211 degrees of freedom, distributed into 400 substructures with 4 010 corners, 831 edges and 1 906 faces. Courtesy of Jan Leština, Jaroslav Novotný and Jakub Šístek.

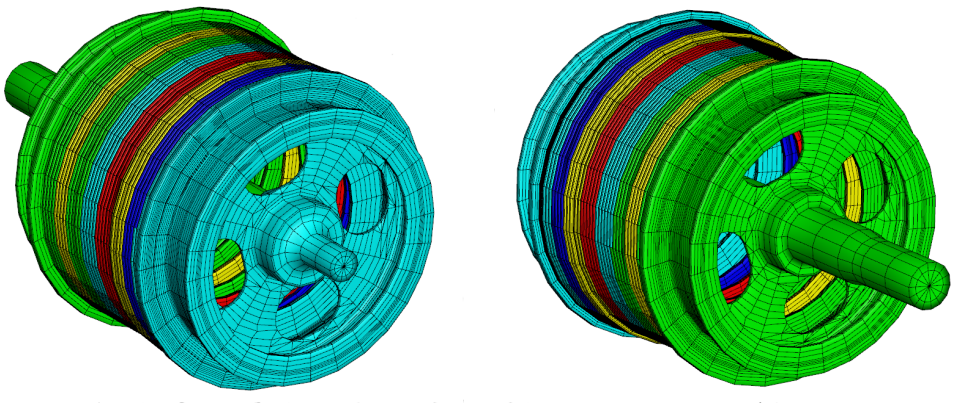


Figure 7.9: Correspondence of finite elements on the zero decomposition level and the subdomains on the second decomposition level. Mining reel with 140 816 finite elements, 400 substructures on the first level and 8 substructures on the second level.

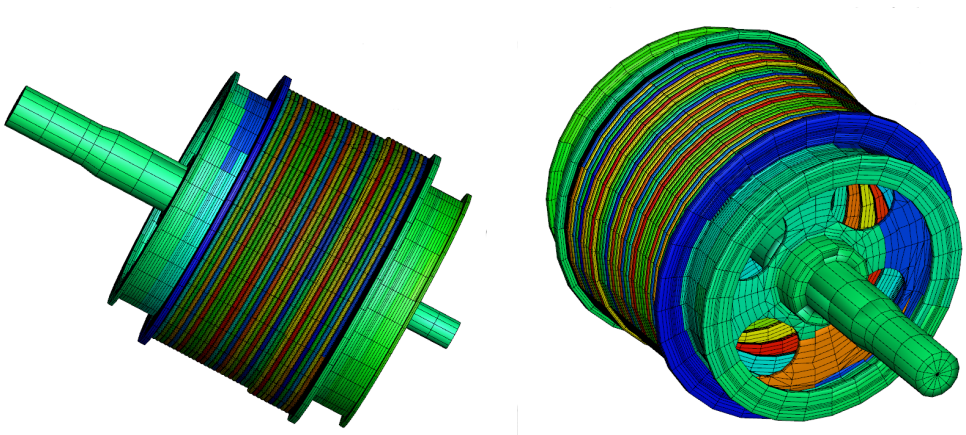


Figure 7.10: Finite element discretization and substructuring of the mining reel problem, consisting of 1 739 211 degrees of freedom, distributed into 1 024 substructures with 7 864 corners, 1 197 edges, and 3 895 faces. Courtesy of Jan Leština, Jaroslav Novotný and Jakub Šístek.

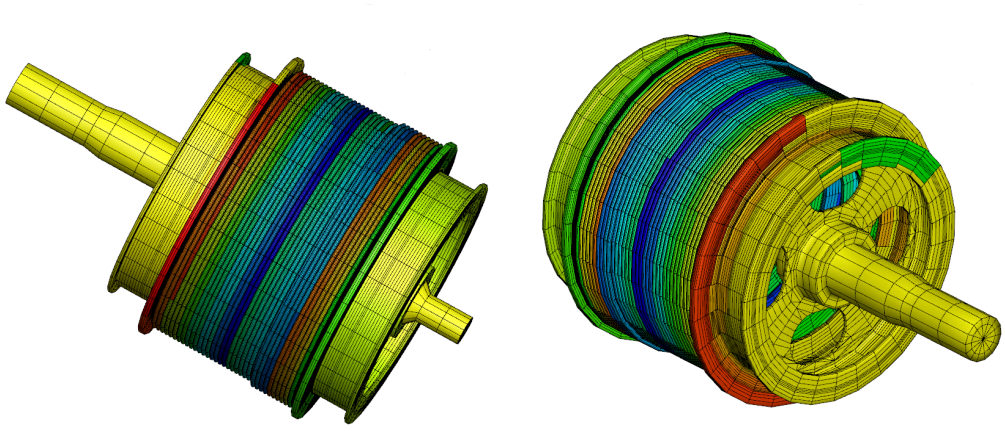


Figure 7.11: Correspondence of finite elements on the zero decomposition level and the subdomains on the second decomposition level. Mining reel with 140 816 finite elements, 1 024 substructures on the first level and 32 substructures on the second level.

Table 7.20: Results for the mining reel (Fig. 7.10, 1 024 substructures) obtained by the non-adaptive 2-level method. The headings are the same as in Table 7.8.

constraint	Nc	$\mathcal{C}$	κ	it
c+e	27 183	-	-	>> 2 000
c+e+f	38 868	22.88	$1.18 \cdot 10^6$	1 303
c+e+f (3eigv)	38 868	22.88	72 704.80	674

Table 7.21: Results for the mining reel (Fig. 7.10, 1 024 substructures) obtained using the adaptive 2-level method. The headings are the same as in Table 7.9.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c+e)	27 183	-	$1.76 \cdot 10^6$	-	$>>$ 2 000
10000	28 023	16.50	9 992.61	9 538.18	910
5000	28 727	16.91	4 934.62	4 849.75	673
1000	32 460	19.11	999.90	2 179.79	391
500	35 017	20.62	499.64	1 277.59	318
100	42 849	25.23	99.89	840.74	213
50	46 093	27.14	49.98	784.49	194
10	59 496	35.03	< 10	321.20	129
5	69 249	40.77	< 5	198.68	91
2	92 467	54.44	< 2	91.24	72

Table 7.22: Results for the mining reel from Figs. 7.10 and 7.11 ($1\,024 + 32$ substructures) obtained using the adaptive 3-level method. The headings are the same as in Tables 7.6 and 7.9.

τ	Nc	nc/ne/nf	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
100	$42\,849 + 2\,378$	208/63/94	$0.79 + 1.40$	99.89^2	3 567.02	382
10	$59\,496 + 6\,419$	208/63/95	1.09/3.78	$< 10^2$	320.82	139
5	$69\,249 + 8\,681$	208/63/95	1.27/5.11	$< 5^2$	198.55	98

7.2.1.3 Application of Adaptive – Multilevel BDDC to a bridge

The power of the adaptive algorithm seems to dominate also for finite element discretization with high aspect ratios. An example of such a problem is a bridge construction discretized by 880 000 hexahedral finite elements with 1 057 920 nodes, 3 173 664 dofs, and decomposed into 1 024 substructures with 6 051 corners, 2 099 edges, and 3 034 faces on the first decomposition level (Fig. 7.13), and 8 substructures on the second decomposition level (Fig. 7.12). A smaller variant of the same problem can be found in [65]. The non-adaptive approach failed, and results for the adaptive 2-level method are summarized in Table 7.23. Note that for convergence of this problem, it was necessary to keep the values of τ relatively low when compared to the previous problems, and the values of τ and κ are again quite different. Finally, comparing convergence results for the two- and three-level methods Tables 7.23 and 7.24, we see that the convergence of the two- and three-level method is quite similar.

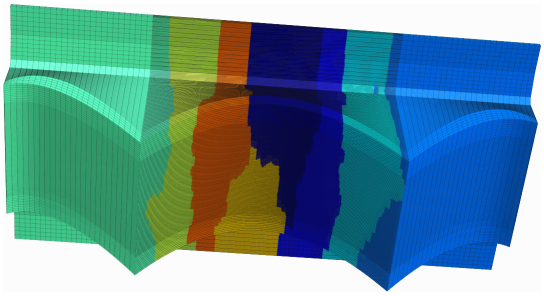


Figure 7.12: Correspondence of finite elements on the zero decomposition level and the subdomains on the second decomposition level. The bridge problem with 880 000 finite elements, 1 024 substructures on the first level and 8 substructures on the second level (a scaled view).

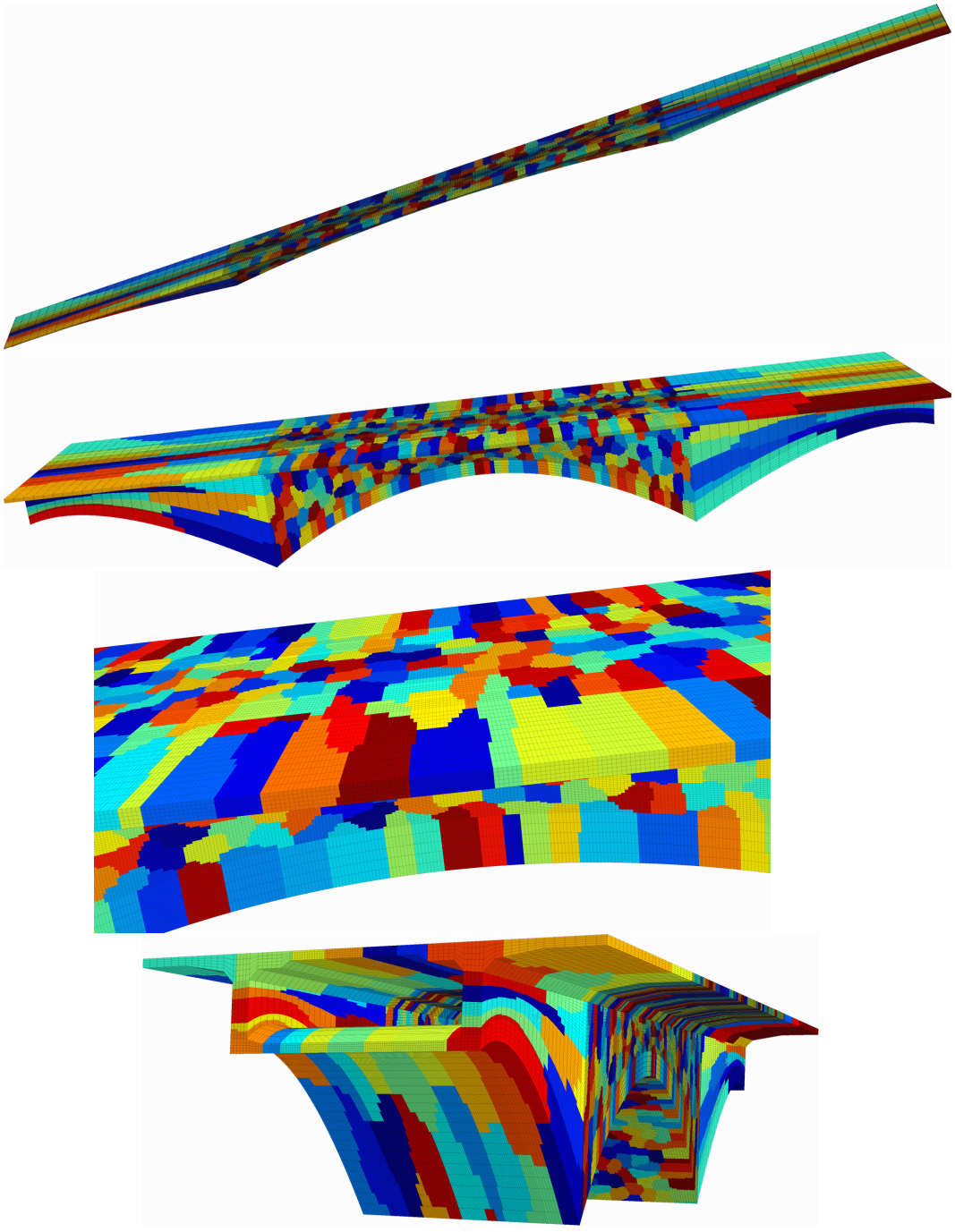


Figure 7.13: Finite element discretization of the bridge construction with 3 173 664 degrees of freedom, distributed into 1 024 subdomains with 6 051 corners, 2 099 edges and 3 034 faces. Courtesy of Jaroslav Kruis.

Table 7.23: Results for the bridge construction from Fig. 7.13 obtained using the adaptive 2-level method. The headings are the same as in Table 7.9.

τ	Nc	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
∞(=c+e)	24 450	7.89	$o(10^7)$	∞	-
100	26 219	8.46	99.95	17 141.40	252
50	27 081	8.74	49.87	11 460.70	191
10	32 219	10.40	9.99	7 014.42	124
5	37 763	12.18	< 5	6 361.90	109
2	61 497	19.84	< 2	5 878.03	90

Table 7.24: Results for the bridge construction from Figs. 7.13 and 7.12 (1 024 + 8 substructures) obtained using the adaptive 3-level method. The headings are the same as in Tables 7.6 and 7.9.

τ	Nc	nc/ne/nf	$\mathcal{C}$	$\widetilde{\omega}$	κ	it
10	32 219 + 197	22/2/9	1.30 + 0.06	$< 10^2$	7 008.82	135
5	37 763 + 309	22/2/9	1.52 + 0.10	$< 5^2$	6 355.71	118
2	61 497 + 1007	22/2/9	2.48 + 0.32	$< 2^2$	5 872.43	94

8. Conclusion

The research presented in this dissertation has been motivated by a persistent need for robust parallel iterative solvers for very large systems of algebraic equations that arise from finite element discretizations of structural mechanics problems. The methods from the iterative substructuring class of domain decomposition seem to be of particular interest, because their philosophy is based naturally on ideas of parallel computing. The methods are typically formulated as preconditioned Krylov subspace methods. We have proposed a new method called *Adaptive – Multilevel BDDC*, combining Multilevel BDDC [63, 64] with adaptive selection of constraints [61, 65, 79]. In the two-level BDDC method, the solution of the coarse problem becomes a bottleneck as the number of subdomains increases. The idea of Multilevel BDDC is to apply recursively the two-level method in order to preserve parallel scalability of the algorithm, although the theory reveals that the condition number grows exponentially with the number of levels. On the other hand the idea of the adaptive method is to locally detect, on each decomposition level, the troublesome parts of a problem and decrease the condition number bound (and the number of iterations) as much as possible with the aid of an a-priori chosen target condition number τ.

The current "prototype" implementation has been programmed and tested in Matlab (version 7.8.0.347 (R2009a)), and run on a 4 Quad Core Opteron 2.0 GHz CPUs computer, with 64 GB RAM and Fedora 10 OS. The code is sequential, and therefore we do not report on CPU times and memory requirements.

The new methods show quite impressive results on various academic and engineering problems, where the standard two-level BDDC with arithmetic averages over edges and faces would perform poorly or even fail. In particular, the Adaptive – Multilevel BDDC outperforms the standard BDDC in the presence of jumps in coefficients not aligned with the substructure boundaries, and on problems with irregular substructures obtained due to the vagaries of a mesh partitioner (both are quite frequent in practical engineering applications). However, for reasonably difficult problems, e.g., the dam problem, it turns out that the Multilevel BDDC performs quite well and there is no need for the adaptive constraints. We have also observed that the increase of the fill-in in the action of the two-level adaptive BDDC is quite small compared to the fill-in of the incomplete Cholesky preconditioner applied to the global stiffness matrix.

We also note that recent, closely related research by Burda, Čertíková, Šístek et al. [10, 76, 88] has focused on a selection of a sufficiently large initial set of corners with the goal of preventing the creation of spurious rigid body mechanisms in the substructuring process. They have also observed numerically that enriching this initially small set of corners with randomly selected interface nodes can quite significantly improve the convergence and computational time.

To allow for a parallel implementation of the Adaptive – Multilevel BDDC, which is our next goal, we have also tested several iterative eigensolvers used in the adaptive algorithm. In particular, we have performed further tests with the LOBPCG [45], for which we have also proposed a local version of the BDDC preconditioner, which is capable of detecting spurious rigid body modes and significantly improving the convergence of the local eigenvalue problems.

REFERENCES

[1] Manoj Bhardwaj, David Day, Charbel Farhat, Michel Lesoinne, Kendall Pierson, and Daniel Rixen. Application of the FETI method to ASCI problems – scalability results on 1000 processors and discussion of highly heterogeneous problems. *Internat. J. Numer. Methods Engrg.*, 47:513–535, 2000.

[2] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring. I. *Math. Comp.*, 47(175):103–134, 1986.

[3] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring. IV. *Math. Comp.*, 53(187):1–24, 1989.

[4] Susanne C. Brenner. Analysis of two-dimensional FETI-DP preconditioners by the standard additive Schwarz framework. *Electron. Trans. Numer. Anal.*, 16:165–185, 2003.

[5] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[6] Susanne C. Brenner and Li-yeng Sung. Balancing domain decomposition for nonconforming plate elements. *Numer. Math.*, 83(1):25–52, 1999.

[7] Susanne C. Brenner and Li-Yeng Sung. BDDC and FETI-DP without matrices or vectors. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1429–1435, 2007.

[8] M. Brezina, C. Heberton, J. Mandel, and P. Vaněk. An iterative method with convergence rate chosen a priori. UCD/CCM Report 140, University of Colorado at Denver, Denver, CO, 1999. `http://ccm.ucdenver.edu/reports/rep140.pdf`.

[9] Jaroslav Brož and Jaroslav Kruis. An algorithm for corner nodes selection in the FETI-DP method. In *Enginnering Mechanics 2009 - CDROM [CD-ROM]*, pages 129–140, Prague, Czech Republic, 2009. Institute of Theoretical and Applied Mechanics AS CR.

[10] P. Burda, M. Čertíková, J. Damašek, A. Novotný, and J. Šístek. Selection of corners for the BDDC method. Submitted to Math. Comput. Simulation, 2009.

[11] Tony F. Chan and Tarek P. Mathew. An application of the probing technique to the vertex space method in domain decomposition. In Roland Glowinski, Yuri A. Kuznetsov, Gérard A. Meurant, Jacques Périaux, and Olof Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 101–111, Philadelphia, PA, 1991. SIAM.

[12] Lawrence C. Cowsar, Jan Mandel, and Mary F. Wheeler. Balancing domain decomposition for mixed finite elements. *Math. Comp.*, 64(211):989–1015, 1995.

[13] Jean-Michel Cros. A preconditioner for the Schur complement domain decomposition method. In Ismael Herrera, David E. Keyes, and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering*, pages 373–380. National Autonomous University of Mexico (UNAM), México, 2003. 14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico, January 6–12, 2002.

[14] Y.-H. De Roeck and P. Le Tallec. Analysis and test of a local domain decomposition preconditioner. In Roland Glowinski, Yuri Kuznetsov, Gérard Meurant, Jacques Périaux, and Olof Widlund, editors, *Proceedings of the Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 112–128. SIAM, 1991.

[15] Y. H. De Roeck, P. Le Tallec, and M. Vidrascu. A domain decomposed solver for nonlinear elasticity. *Comput. Methods Appl. Mech. Engrg.*, 99:187–207, 1992.

[16] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[17] Clark R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[18] Clark R. Dohrmann. An approximate BDDC preconditioner. *Numer. Linear Algebra Appl.*, 14(2):149–168, 2007.

[19] Maksymilian Dryja. A method of domain decomposition for 3-D finite element problems. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 43–61, Philadelphia, PA, 1988. SIAM.

[20] Maksymilian Dryja and Olof B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, 1995.

[21] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and Rixen, D. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[22] Charbel Farhat, Po-Shu Chen, Jan Mandel, and Francois Xavier Roux. The two-level FETI method. II. Extension to shell problems, parallel implementation and performance results. *Comput. Methods Appl. Mech. Engrg.*, 155(1-2):153–179, 1998.

[23] Charbel Farhat, Michael Lesoinne, and Kendall Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7:687–714, 2000.

[24] Charbel Farhat and Jan Mandel. The two-level FETI method for static and dynamic plate problems. I. An optimal iterative solver for biharmonic systems. *Comput. Methods Appl. Mech. Engrg.*, 155(1-2):129–151, 1998.

[25] Charbel Farhat, Jan Mandel, and Fancos-Xavier Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:365–385, 1994.

[26] Charbel Farhat and Francois-Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32:1205–1227, 1991.

[27] J. Fish and V. Belsky. Generalized aggregation multilevel solver. *Internat. J. Numer. Methods Engrg.*, 40(23):4341–4361, 1997.

[28] Yannis Fragakis. Force and displacement duality in domain decomposition methods for solid and structural mechanics. Preprint, 2006. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.75.3691`.

[29] Yannis Fragakis and Manolis 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. Engrg.*, 192:3799–3830, 2003.

[30] Roland Glowinski and Mary F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[31] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Univ. Press, 1989. Second Edition.

[32] Gene H. Golub and Qiang Ye. An inverse free preconditioned krylov subspace method for symmetric generalized eigenvalue problems. *SIAM Journal on Scientific Computing*, 24(1):312–334, 2002.

[33] David Horák. *FETI based domain decomposition methods for variational inequalities*. PhD thesis, Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science, VŠB - Technical University Ostrava, 2007.

[34] George Karypis and Vipin 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.

[35] David E. Keyes and William D. Gropp. A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. *SIAM J. Sci. Statist. Comput.*, 8(2):S166–S202, 1987. Parallel processing for scientific computing (Norfolk, Va., 1985).

[36] Hyea Hyun Kim and Xuemin Tu. A three-level BDDC algorithm for mortar discretizations. *SIAM Journal on Numerical Analysis*, 47(2):1576–1600, 2009.

[37] Axel Klawonn and Oliver Rheinbach. Inexact FETI-DP methods. *International Journal for Numerical Methods in Engineering*, 69(2):284–307, 2007.

[38] Axel Klawonn and Oliver Rheinbach. A hybrid approach to 3-level FETI. *PAMM*, 8(1):10841–10843, 2008. 79th Annual Meeting of the International

Association of Applied Mathematics and Mechanics (GAMM), Bremen 2008.

[39] Axel Klawonn, Oliver Rheinbach, and Olof B. Widlund. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.*, 46(5):2484–2504, 2008.

[40] Axel Klawonn and Olof B. Widlund. A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity. *SIAM J. Sci. Comput.*, 22(4):1199–1219, 2000.

[41] Axel Klawonn and Olof B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Comm. Pure Appl. Math.*, 54(1):57–90, 2001.

[42] Axel Klawonn and Olof B. Widlund. Dual-primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59(11):1523–1572, 2006.

[43] Axel Klawonn, Olof B. Widlund, and Maksymilian Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40(1):159–179, 2002.

[44] Axel Klawonn, Olof B. Widlund, and Maksymilian Dryja. Dual-primal FETI methods with face constraints. In Luca F. Pavarino and Andrea Toselli, editors, *Recent Developments in Domain Decomposition Methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 27–40. Springer-Verlag, 2002.

[45] Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.*, 23(2):517–541, 2001. Copper Mountain Conference (2000).

[46] Andrew V. Knyazev. personal communication, 2008.

[47] Jaroslav Kruis. *Domain decomposition methods for distributed computing.* Saxe-Coburg Publications, Kippen, Stirling, Scotland, 2006.

[48] Patrick Le Tallec, Jan Mandel, and Marina Vidrascu. Balancing domain decomposition for plates. *Contemp. Math.*, 180:515–524, 1994. Proceedings of the 7th International Symposium on Domain Decomposition Methods, Penn State, November 1993.

[49] Patrick Le Tallec, Jan Mandel, and Marina Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Numer. Anal.*, 35:836–867, 1998.

[50] Michael Lesoinne. A FETI-DP corner selection algorithm for three-dimensional problems. In Ismael Herrera, David E. Keyes, and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering*, pages 217–223. National Autonomous University of Mexico (UNAM), México, 2003. 14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico, January 6–12, 2002. http://www.ddm.org.

[51] Jing Li and Olof B. Widlund. FETI-DP, BDDC, and block Cholesky methods. *Internat. J. Numer. Methods Engrg.*, 66(2):250–271, 2006.

[52] Jing Li and Olof B. Widlund. On the use of inexact subdomain solvers for BDDC algorithms. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1415–1428, 2007.

[53] Jan Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[54] Jan Mandel. Intelligent block iterative methods. In J. Robinson, editor, *FEM Today and the Future*, pages 471–477. Robinson and Associates, 1993. Proceedings of the 7th World Congress on Finite Elements, Monte Carlo, November 1993.

[55] Jan Mandel. Hybrid domain decomposition with unstructured subdomains. In *Proceedings of the 6th International Symposium on Domain Decomposition Methods, Como, Italy, 1992*, volume 157 of *Contemp. Math.*, pages 103–112. 1994.

[56] Jan Mandel. An iterative solver for p-version finite elements in three dimensions. *Comput. Methods Appl. Mech. Engrg.*, 116(1-4):175–183, 1994. ICOSAHOM'92 (Montpellier, 1992).

[57] Jan Mandel and Marian Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65(216):1387–1401, 1996.

[58] Jan Mandel and Clark R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[59] Jan Mandel, Clark R. Dohrmann, and Radek Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54(2):167–193, 2005.

[60] Jan Mandel and Bedřich Sousedík. Adaptive coarse space selection in the BDDC and the FETI-DP iterative substructuring methods: Optimal face degrees of freedom. In Olof B. Widlund and David E. Keyes, editors, *Domain Decomposition Methods in Science and Engineering XVI, Lecture Notes in Computational Science and Engineering, vol. 55*, pages 421–428. Springer-Verlag, 2006.

[61] Jan Mandel and Bedřich Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1389–1399, 2007.

[62] Jan Mandel and Bedřich Sousedík. BDDC and FETI-DP under minimalist assumptions. *Computing*, 81:269–280, 2007.

[63] Jan Mandel, Bedřich Sousedík, and Clark R. Dohrmann. On multilevel BDDC. *Lecture Notes in Computational Science and Engineering*, 60:287–294, 2007. Domain Decomposition Methods in Science and Engineering XVII.

[64] Jan Mandel, Bedřich Sousedík, and Clark R. Dohrmann. Multispace and multilevel BDDC. *Computing*, 83(2-3):55–85, 2008.

[65] Jan Mandel, Bedřich Sousedík, and Jakub Šístek. Adaptive BDDC in three dimensions. Submitted to Math. Comput. Simulation, 2009.

[66] Jan Mandel and Radek Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numer. Math.*, 73(4):473–487, 1996.

[67] Jan Mandel and Radek Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[68] Jan Mandel, Radek Tezaur, and Charbel Farhat. A scalable substructuring method by Lagrange multipliers for plate bending problems. *SIAM J. Numer. Anal.*, 36(5):1370–1391, 1999.

[69] Luca F. Pavarino and Olof B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.

[70] Clemens Pechstein and Robert Scheichl. Analysis of FETI methods for multiscale PDEs. *Numer. Math.*, 111(2):293–333, 2008.

[71] Clemens Pechstein and Robert Scheichl. Analysis of FETI methods for multiscale PDEs - Part II: Interface variations. Numerische Mathematik, submitted, 2009.

[72] Vilém Pompe and Matěj Hraška. Measurement of basic properties of an epoxy matrix (in Czech). *Transfer, Electronic magazine VZLÚ, a.s. - Research and Development for Aircraft Industry*, 5:63–72, 2007. `http://www.vzlu.cz/download.php?file=67`.

[73] Gene Poole, Yong-Cheng Liu, and Jan Mandel. Advancing analysis capabilities in ANSYS through solver technology. *Electronic Transactions on Numerical Analysis*, 15:106–121, 2003.

[74] Alfio Quarteroni and Alberto Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.

[75] Daniel J. Rixen, Charbel Farhat, Radek Tezaur, and Jan Mandel. Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver. *Internat. J. Numer. Methods Engrg.*, 46:501–534, 1999.

[76] J. Šístek, J. Novotný, J. Mandel, M. Čertíková, and Burda, P. BDDC by a frontal solver and stress computation in a hip joint replacement. *Math. Comput. Simulation*, 80(6):1310–1323, 2010.

[77] Barry F. Smith, Petter E. Bjørstad, and William D. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, Cambridge, 1996.

[78] Bedřich Sousedík and Jan Mandel. On Adaptive-Multilevel BDDC. In *Nineteenth International Conference on Domain Decomposition*. Springer-Verlag, 2010. Submitted.

[79] Bedřich Sousedík. *Comparison of Some Domain Decomposition Methods*. PhD thesis, Czech Technical University in Prague, Faculty of Civil Engineering, Department of Mathematics, 2008.

[80] Bedřich Sousedík and Jan Mandel. On the equivalence of primal and dual substructuring preconditioners. *Electron. Trans. Numer. Anal.*, 31:384–402, 2008.

[81] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

[82] Xuemin Tu. *BDDC Domain Decomposition Algorithms: Methods with Three Levels and for Flow in Porous Media*. PhD thesis, Department of Mathematics, New York University, 2006.

[83] Xuemin Tu. Three-level BDDC. Domain decomposition methods in science and engineering, Proceedings DD16, January 2005, 2006.

[84] Xuemin Tu. Three-level BDDC in three dimensions. *SIAM J. Sci. Comput.*, 29(4):1759–1780, 2007.

[85] Xuemin Tu. Three-level BDDC in two dimensions. *Internat. J. Numer. Methods Engrg.*, 69(1):33–59, 2007.

[86] Xuemin Tu. A three-level BDDC algorithm for saddle point problems. submitted to Numer. Math., 2008.

[87] Marta Čertíková. Domain decomposition methods. In Okrouhlík, M., editor, *Numerical methods in computational mechanics*, pages 165–203. Institute of Thermomechanics, Prague, 2008. http://www.it.cas.cz/elektronicka-kniha-numerical-methods-computation-mechanics.

[88] Marta Čertíková. *Computational aspects of the FEM for solving boundary value problems*. PhD thesis, Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Mathematics, 2010.

[89] Olof B. Widlund. Iterative substructuring methods: algorithms and theory for elliptic problems in the plane. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 113–128. SIAM, Philadelphia, PA, 1988.

[90] Olof B. Widlund. Accomodating irregular subdomains in domain decomposition theory. In M. Bercovier, M.J. Gander, R. Kornhuber, and O. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XVIII*, volume 70 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2009. Proceedings of 18th International Conference on Domain Decomposition, Jerusalem, Israel, January 2008.

[91] Barbara I. Wohlmuth. *Discretization methods and iterative solvers based on domain decomposition*, volume 17 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001.