Face-based Selection of Corners in 3D Substructuring

Jakub Šístek[a,*], Marta Čertíková[b], Pavel Burda[b], Jaroslav Novotný[c]

[a]*Institute of Mathematics, Academy of Sciences of the Czech Republic*
[b]*Department of Mathematics, Faculty of Mechanical Engineering*
Czech Technical University in Prague
[c]*Department of Mathematics, Faculty of Civil Engineering*
Czech Technical University in Prague

Abstract

In most recent substructuring methods, a fundamental role is played by the coarse space. For some of these methods (e.g. BDDC and FETI-DP), its definition relies on a 'minimal' set of coarse nodes (sometimes called corners) which assures invertibility of local subdomain problems and also of the global coarse problem. This basic set is typically enhanced by enforcing continuity of functions at some generalized degrees of freedom, such as average values on edges or faces of subdomains. We revisit existing algorithms for selection of corners. The main contribution of this paper consists of proposing a new heuristic algorithm for this purpose. Considering faces as the basic building blocks of the interface, inherent parallelism, and better robustness with respect to disconnected subdomains are among features of the new technique. The advantages of the presented algorithm in comparison to some earlier approaches are demonstrated on three engineering problems of structural analysis solved by the BDDC method.

Keywords: domain decomposition, iterative substructuring, finite elements, linear elasticity, parallel algorithms, corner selection

1. Introduction

The Balancing Domain Decomposition based on Constraints (BDDC) is a numerically scalable, nonoverlapping (substructuring), primary domain decomposition method introduced in 2003 by Dohrmann [4]. Its algebraic theory developed by Mandel, Dohrmann and Tezaur in [14] demonstrates close relation to FETI-DP introduced by Farhat, Lesoinne, and Pierson [5]: the eigenvalues

---

*Corresponding author at: Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, CZ–115 67 Praha 1, Czech Republic. Tel.: +420 222 090 710; fax: +420 222 211 638

Email addresses: sistek@math.cas.cz (Jakub Šístek), marta.certikova@fs.cvut.cz (Marta Čertíková), pavel.burda@fs.cvut.cz (Pavel Burda), novotny@mat.fsv.cvut.cz (Jaroslav Novotný)

of the preconditioned problem in BDDC and FETI-DP are the same except possibly those equal to 0 and 1 (see also [2], [13], and [16] for simplified proofs). These results not only provide the theoretical reasoning for nearly identical performance of BDDC and FETI-DP observed earlier, but also imply, that many theoretical results obtained for one method apply readily to the other.

The *coarse space*, defined by constraints on continuity of functions across the interface at *coarse degrees of freedom*, is essential for the performance of both methods. A historical overview of an evolution of the concept of the coarse space is presented, e.g., by Widlund in [23] and by Mandel and Sousedík in [17]. The usual basic choice of coarse degrees of freedom is presented by selecting a set of *coarse nodes* (also called *corners*). This set is usually selected to be 'minimal' in the sense that it is as small as possible while assuring invertibility of local subdomain problems and of the global coarse problem. For 2D problems this choice ensures good convergence properties. However, both methods require additional constraints on some generalized degrees of freedom such as average values on edges or faces of subdomains to achieve good efficiency for 3D problems. This fact was first discovered for FETI-DP: experimentally observed in Farhat, Lesoinne, and Pierson [5] and theoretically supported by Klawonn, Widlund and Dryja in [11]. These observations apply to BDDC through the above-mentioned correspondence between both methods.

A sufficiently robust definition of the coarse space in BDDC and FETI-DP is still not available, especially for complex 3D geometries, and existing methods tend to fail for such problems. Related work on choice of the coarse degrees of freedom has focused on selecting a small and effective coarse space. An algorithm for selecting the smallest set of coarse nodes to avoid coarse mechanism is described by Lesoinne in [12]. Another algorithm, which is already based on pairs of subdomains, was given by Dohrmann in [4]. This task has been recently further discussed by Brož and Kruis in [3] for 2D case. Klawonn and Widlund in [9] and [10] minimize a set of more general coarse degrees of freedom (like weighted averages over edges and faces) to achieve optimal convergence estimates, introducing the concept of an acceptable path. Adaptive selection of coarse degrees of freedom based on local estimates using eigenvectors associated with faces is described by Mandel and Sousedík in [15], and by Mandel, Sousedík, and Šístek in [19]. In this adaptive approach, which provides additional averages on faces leading to optimal decrease of the expected condition number, a sufficient number of initial constraints is required between each pair of subdomains as an input. This assumption is in good agreement with the output of the algorithm proposed in this paper.

While proposing the new algorithm for selection of the basic set of corners is the main contribution of the manuscript, we further explore the potential of adding more coarse nodes into the coarse problem. This approach is technically simple and allows flexible setting of desired approximation. It is observed, that by loosening the requirement of 'minimal' selection and identifying more interface nodes as corners, the performance of the BDDC preconditioner may be cheaply but considerably improved. Numerical experiments on industrial 3D elasticity problems demonstrate the advantages of the new corner selecting

algorithm in comparison to several earlier approaches. They also show the fact, that by enhancing the basic set of constraints by additional coarse nodes, the computational times might be further reduced.

2. BDDC method

In this paper, we study the selection of the initial set of constraints in the context of the BDDC method [4], which is briefly recalled in this section. However, the main ideas of the paper apply to FETI-DP method as well.

After a discretization of a linearized partial differential equation of elliptic type in a given domain Ω by means of finite element method (FEM), a system of linear algebraic equations

$$\mathbf{A}\mathbf{x} = \mathbf{f} \tag{1}$$

with a symmetric positive definite matrix $\mathbf{A}$ and a right-hand side $\mathbf{f}$ is solved for the unknown vector $\mathbf{x}$. Components of $\mathbf{x}$ represent function values at mesh nodes and they are often called *degrees of freedom*. In 3D linear elasticity, there are 3 unknown values of displacement (3 degrees of freedom) at every mesh node.

The first step in the BDDC method is the reduction of the problem to the interface. This is quite standard and described in the literature, e.g., Toselli and Widlund [22]: the underlying discretized domain Ω is split into N nonoverlapping subdomains (also called *substructures*) Ω_i, $i = 1, \ldots, N$ with common interface Γ, and problem (1) is reduced to the Schur complement problem with respect to interface

$$\mathbf{S}\mathbf{u} = \mathbf{g} \tag{2}$$

with a symmetric positive definite matrix $\mathbf{S}$. The vector $\mathbf{u}$ now represents the subset of degrees of freedom in $\mathbf{x}$ that correspond to the interface Γ. Solution $\mathbf{u}$ of the problem (2) can be also represented as the minimum of the functional

$$\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{S}\mathbf{u} - \mathbf{u}^{\mathrm{T}}\mathbf{g} \to min\,, \quad \mathbf{u} \in \widehat{W} \tag{3}$$

on the space $\widehat{W}$ of unknowns on the interface Γ. The space $\widehat{W}$ can be identified with the space of *discrete harmonic functions*, that are fully determined by their values of unknowns on the interface Γ and have minimal energy on every subdomain.

The problem (2) is then solved by the preconditioned conjugate gradient (PCG) method, for which BDDC acts as the preconditioner. The main idea of the BDDC method is shortly described bellow. More details, together with connection to FETI-DP, can be found in Mandel, Dohrmann and Tezaur [14] or Mandel and Sousedík [16].

A preconditioner $\mathbf{M}$ for the system (2) should realize an approximation of $\mathbf{S}^{-1}$ such that obtaining a preconditioned residual $\mathbf{p} = \mathbf{M}\mathbf{r}$ can be considerably easier than solving the original problem (2). Construction of the BDDC

preconditioner is based on the idea that instead of minimising (3) on the space $\widehat{W}$, which represents solving the system (2), the minimization is performed on some larger space $\widetilde{W}$ such that $\widehat{W} \subset \widetilde{W}$:

$$\frac{1}{2}\widetilde{\mathbf{u}}^{\mathrm{T}}\widetilde{\mathbf{S}}\widetilde{\mathbf{u}} - \widetilde{\mathbf{u}}^{\mathrm{T}}\widetilde{\mathbf{g}} \to min\,, \quad \widetilde{\mathbf{u}} \in \widetilde{W}, \tag{4}$$

where $\widetilde{\mathbf{S}}$ is a symmetric positive definite extension of $\mathbf{S}$ to $\widetilde{W}$ and $\widetilde{\mathbf{g}}$ is an extension of $\mathbf{g}$. The space $\widetilde{W}$ has to be chosen so that the symmetric positive definite extension $\widetilde{\mathbf{S}}$ on $\widetilde{W}$ exists. At the same time, solving problem (4) should be considerably easier than solving the original problem (3), while providing good approximation of the solution of (3). The BDDC preconditioner is then defined as

$$\mathbf{M} = \mathbf{E}\widetilde{\mathbf{S}}^{-1}\mathbf{E}^{\mathrm{T}}\,, \tag{5}$$

where $\mathbf{E}$ represents a projection from $\widetilde{W}$ onto $\widehat{W}$ realized by a kind of averaging.

3. Coarse degrees of freedom

In BDDC, the space $\widetilde{W}$ is specified by relaxing the requirement of the continuity of discrete harmonic functions across the interface. The functions of $\widetilde{W}$ are required to be continuous only at selected *coarse degrees of freedom*. In this paper, we focus on the simplest choice of coarse degrees of freedom, which is a function value at a selected node on the interface. Such node is then called *coarse node* or *corner*. More general coarse degrees of freedom are commented at the end of this section and are considered in computations.

In terms of mechanics, the transition from $\widehat{W}$ to $\widetilde{W}$ can be interpreted as making cuts into the continuous function along the interface, leaving the function continuous across the interface only at the corners. A schematic illustration of the continuity constraints is depicted in Figure 1: functions from $\widehat{W}$ are continuous across the interface, functions from $\widetilde{W}$ are continuous only at selected coarse nodes.

The space $\widetilde{W}$ can be decomposed as $\widetilde{\mathbf{S}}$-orthogonal direct sum

$$\widetilde{W} = \widetilde{W}_1 \oplus \cdots \oplus \widetilde{W}_N \oplus \widetilde{W}_C\,, \tag{6}$$

where $\widetilde{W}_i$, $i = 1, \ldots, N$, are local, subdomain spaces and $\widetilde{W}_C$ is the global *coarse space*, defined as the $\widetilde{\mathbf{S}}$-orthogonal complement of all spaces $\widetilde{W}_i$, i.e.

$$\mathbf{w}_C \in \widetilde{W}_C \;\Leftrightarrow\; \mathbf{w}_C^T\widetilde{\mathbf{S}}\mathbf{w} = 0 \;\; \forall \mathbf{w} \in \widetilde{W}_i,\; i = 1, \ldots, N. \tag{7}$$

Functions from $\widetilde{W}_i$ can have nonzero values only in Ω_i except for coarse degrees of freedom. They have zero values at coarse degrees of freedom, and they are fully determined by degrees of freedom on Γ and the discrete harmonic extension to interiors of subdomains. Similarly, functions from $\widetilde{W}_C$ are fully determined by their values at coarse degrees of freedom (where they are continuous) and by the discrete harmonic extension to interiors of subdomains

and on the rest of the interface (i.e. everywhere apart from the coarse nodes). Functions from the spaces $\widetilde{W}_C$ and $\widetilde{W}_i$ are generally discontinuous across Γ outside corners.

According to decomposition (6), solution of the problem (4) can now be split into solution of N local subdomain problems on the spaces $\widetilde{W}_i$ and one global coarse problem on the coarse space $\widetilde{W}_C$. All these problems are mutually independent and so can be naturally parallelized.

Coarse degrees of freedom have to be selected so that stable invertibility of both the coarse problem and the local problems is assured. Important role of the coarse space is to assure scalability by global error propagation over the whole domain. It was shown that while for 2D elasticity problems the BDDC (or FETI-DP) preconditioner is scalable for coarse space defined by coarse nodes (corners) only, in 3D elasticity problems more general coarse degrees of freedom, such as (weighted) average values over edges and faces, need to be used in order to achieve scalability, see e.g. Toselli and Widlund [22].

Choice of the coarse degrees of freedom has a great impact on the performance of the preconditioner $\mathbf{M}$. The more coarse degrees of freedom are chosen, the more difficult it is to obtain the solution of (4), which, on the other hand, is then closer to the solution of the original problem (3). In the extreme case of selecting all interface nodes as coarse, $\widetilde{W}_C \equiv \widetilde{W} \equiv \widehat{W}$, coarse problem becomes the original problem (3) and $\mathbf{M} \equiv \mathbf{S}^{-1}$. In the opposite extreme, if no coarse degrees of freedom are selected, $\widetilde{W}_C$ is empty and solution of (4) splits to N local problems only, some of which might not be invertible. Thus, the optimal choice of the coarse space lies somewhere in-between.

4. Geometry and selection of the coarse space in 3D

The interface Γ in 3 dimensions can be specified as a set of nodes belonging to at least two subdomains (subdomains are considered as closed sets). It consists of subdomain faces, edges and vertices. While there is an intuitive geometric notion what these three entities mean in a simple case of a cubic domain divided into cubic subdomains, there is no unique exact classification in more general case of domain with complicated geometry and subdomains obtained, e.g., by a graph partitioning tool. We adopt the classification presented by Klawonn and Rheinbach in [8] and use it in a slightly simplified form, which does not assume knowledge of boundary of the domain and is easy to implement:

Definition 1.

- a *face* contains all nodes shared by the same two subdomains,

- an *edge* contains nodes shared by the same more than two subdomains,

- a *vertex* is a degenerated edge with only one node.

Then every node of the interface belongs to just one of the entities defined above. Two subdomains are called *adjacent* if they share a face.

However, this classification does not reproduce our intuition in the case of cubic subdomains, as can be seen in Figure 2: for instance the interface of a domain consisting of two cubic subdomains has neither vertex nor edge, just one face (the left case in Figure 2). Different definitions of faces and edges are discussed by Klawonn and Rheinbach in [7, Section 2].

In practice, there are often not enough vertices, edges, or faces for satisfactory number of constraints. We have found it useful to introduce one additional entity:

Definition 2.

- a *corner* is any interface node selected as coarse.

In implementations of the BDDC method, it is often customary to distinguish between the following two kinds of constraints on continuity across interface.

Node constraints - corners
The most obvious choice of coarse degrees of freedom are node constraints (at corners). The basic choice is a set of corners, that assures invertibility of local subdomain problems and also the global coarse problem. This is often put as a requirement on their selection (e.g. in [4], [21]).

Although vertices provide a good initial set of corners, they often do not suffice for assuring invertibility of subdomain problems and/or of the coarse problem (cf. Figure 2), and other constraints need to be added. When other nodes are selected as corners, they have to be excluded from corresponding faces or edges, so that every interface node is either a corner, or belongs to a face or an edge.

Corner constraints are not as efficient as constraints on averages on edges or faces, nevertheless they can be used as a substitute for these constraints, if enough corners are employed. Figure 7 left illustrates the typical dependence of the condition number of the preconditioned problem on number of corners randomly selected from the interface, starting from some basic set. For small numbers of corners, we can observe poor performance of the preconditioner even though all system matrices are invertible. Then, after a typical sudden drop, the condition number improves only slightly with adding more corners. Number of iterations reproduces this dependence, see Figure 7 (centre).

Improving convergence by adding more corners leads to a larger coarse problem than adding averages on faces or edges. On the other hand, its implementation is straightforward and its scaling is easy to maintain.

For 2D problems, the basic set of corner constraints already ensures good convergence properties. Although an efficient BDDC method for 3D elliptic problems requires also constraints on some generalized degrees of freedom, such as average values on edges or faces of subdomains described below, for many industrial problems this simple approach also leads to satisfactory results.

Constraints on averages over edges and faces
General coarse degrees of freedom can be constructed as any linear combinations

of function values at nodes belonging to one face or one edge. This type of constraints is required for both BDDC and FETI-DP methods in three dimensions, if one expects the optimal polylogarithmic bound on condition number κ of the preconditioned operator

$$\kappa(\mathbf{MS}) \leq const. \left(1 + \log \frac{H}{h}\right)^2, \tag{8}$$

where H is the subdomain size and h is the finite element size (see [11]).

One of the standard choices is an arithmetic average over unknowns separately for each component of displacement leading to three constraints for 3D elasticity. We have tested this standard choice applied to all edges, to all faces, or both. More sophisticated methods of weighted averaging were developed, e.g., by Klawonn and Widlund [10], by Mandel and Sousedík [15], or recently by Mandel, Sousedík, and Šístek [19].

5. Selection of the basic set of corners

In this section, we concentrate on the selection of the basic set of corners that leads to positive definiteness of matrix $\widetilde{\mathbf{S}}$ in (4). This task is equivalent to assuring invertibility of both local subdomain problems and the global coarse problem only by corner constraints, which is often required by implementations (cf. [4], [21]). Therefore, we investigate selection of corners independently of enforcing constraints on general coarse degrees of freedom.

From the mechanical point of view, the question of assuring invertibility of local subdomain problems corresponds to enforcing enough boundary conditions on a body to fix rigid body modes, with subdomain playing the role of the body. This goal is easily attained by selecting three nodes (not in a line) of the interface of a subdomain as corners.

It turns out, that assuring invertibility of the coarse matrix is the more difficult task, since selection with respect to subdomain problems only may still lead to *mechanisms* in the coarse problem (see [12]). To see this, one can simply think of a domain divided into subdomains in a linear fashion. Figure 3 illustrates this on a 2D case, where two corners for each subdomain are sufficient for invertibility of subdomain stiffness matrices.

An algorithm attempting to select the smallest set of coarse nodes to avoid coarse mechanisms was given by Lesoinne in [12]. Minimization of the number of corners is obtained mainly by favouring already selected corners. Thus, the approach is serial in its nature.

Another algorithm for selecting corners was described already by Dohrmann in [4]. It is based on the investigation of all possible neighbourings between substructures and selecting three corners from each such set, that maximise the area of a triangle with corners at its vertices. However, this algorithm is based on an incomplete classification of interface into vertices, edges, and faces, and it does not distinguish between the last two groups. Also this algorithm favours already selected corners by selecting vertices on the interface as the

initial vertices of the triangle to be maximised. Nevertheless, it has provided a good starting point for the new algorithm proposed here.

The third algorithm, which is based on selection of corners along edges, was described in [20]. This idea is inspired by the definition of corners as endpoints of edges by Klawonn and Widlund [9]. Although it was successfully used by our group in a number of practical computations, it may fail to produce a mechanism-free coarse problem in the case of divisions where no edges are present (cf. the leftmost case in Figure 2).

The aim to select a low number of corners inherent to all these algorithms is motivated by the fact, that low number of corners results in a small size of the matrix of the coarse problem and its cheap factorization. However, it has been observed on a number of experiments (e.g. [21], also Section 7 in this paper) that this motivation may be misleading, and in fact, larger sets are preferable for the performance of the preconditioner often resulting in much lower number of PCG iterations. It has been also shown, that using more corners may lead to a considerable reduction of the computational time in spite of the longer time spent in factorization of the larger matrix of the coarse problem, even in the case of considering averages on edges and faces.

Based on these observations and experience with the algorithms, we see several ideas that the new proposed algorithm should reflect:

(i) selection with respect to faces (by Definition 1) as these are the basic building blocks of interface in 3D structures (Figure 2),

(ii) provide larger set of corners than the previous algorithms as this usually leads to much better preconditioning,

(iii) independence of selection subdomain by subdomain and of order of going through subdomains (better parallelization).

Points (ii) and (iii) are attained simply by not favouring already selected corners and selecting optimal distribution of at least three corners between each pair of substructures sharing a face, i.e. adjacent substructures, independently.

Let us now present an algorithm satisfying these requirements. For this, denote the set of faces of subdomain Ω_i as $\mathcal{F}(\Omega_i)$ and recall that N denotes the number of subdomains. A face $\mathcal{F}_{ij}$ between subdomains Ω_i and Ω_j is present in both sets $\mathcal{F}(\Omega_i)$ and $\mathcal{F}(\Omega_j)$.

Algorithm 1 (Selection of corners for 3D elasticity problems).

1. *Classify interface according to Definition 1 and use all vertices as corners.*
2. ***For** subdomain Ω_i, $i = 1, \ldots, N$,*
   ***For** face $\mathcal{F}_{ij} \in \mathcal{F}(\Omega_i)$, $j = 1, \ldots, \mathrm{size}(\mathcal{F}(\Omega_i))$,*

   - *find the set of* all *nodes shared with the adjacent subdomain (generally larger set than the face under consideration, as it may contain also edges and/or vertices),*

   - *construct a graph of nodes of this set with connections defined by elements, and detect components of this graph*

   - ***For** each such component, select (in 3D) three corners by:*

8

- (a) pick an arbitrary node of the subset,
- (b) find the first corner as the most remote node from the arbitrary node,
- (c) find the second corner as the most remote node from the first corner,
- (d) find the third corner as the node maximising the area of the triangle,

  end,

  end,

  end.

3. *Select corners as the union of vertices and face-based selection above.*
4. *Remove selected corners from edges and faces.*

The algorithm assures that at least three corners are selected in an optimal way with respect to each face. This situation is often not obtained by favouring already selected corners, since corners optimally distributed for one pair of subdomains may be far from optimal distribution with respect to another pair. Presented algorithm is also much simpler for parallelization than algorithms favouring already selected corners, since communication is needed only at the end of the selection to synchronise locally detected corners. It typically provides more corners than algorithms mentioned above, which we consider as an advantage rather than a drawback.

Remark 1. A modification of Algorithm 1 favouring already selected corners is simply possible by entering the face-based selection in any point (a), (b), (c), or (d), depending on how many corners are already selected between adjacent substructures. This modification leads to selection that is very similar to the algorithm by Dohrmann in [4]. In our experience, this modification, referred to as 'minimal', leads to lower number of corners, but also usually to worse results (some of them are presented in Section 7). Thus, we recommend using the ('full') version as stated by Algorithm 1.

Remark 2. A modification of Algorithm 1 for 2D problems (where no edges are present) or topologically 2D problems (such as for shell elements in 3D) is simply possible by finishing the face-based selection with point (c).

Remark 3. Detection of components is aimed at problems divided into subdomains by graph partitioners, such as METIS. These programs typically provide divisions well balanced with respect to size of subdomains, but often with some subdomains disconnected. Such divisions present a challenge for many existing domain decomposition methods. With the detection of components, the algorithm is able to detect many of such discontinuities, and fix each component independently. The BDDC method is then able to proceed with computation, keeping such subdomains disconnected, thus preserving the suggested balance of load.

We show the power of this detection on a simple problem of an elastic beam consisting of two subdomains, one of which is wedged in the other as in Figure 4. On the left-hand side of the figure, corners selected without the detection of components are presented. In this case, both cuts are handled as a single interface, and the search of triangle with maximal area does not succeed. Resulting configuration has a mechanism in the coarse problem and, consequently, BDDC method fails.

On the right-hand side of the figure, corners obtained with the component detection enabled are shown. Now the optimal triangle is sought at each of the cuts, which leads to a mechanism-free configuration of corners, and the BDDC method converges in four iterations.

6. Implementation

The BDDC method has been implemented on top of common components of existing finite element codes, namely the frontal solver and the element stiffness matrix generation. Such implementation requires only a minimal amount of additional code. In our case, most of the program is written in Fortran 77, with some parts in Fortran 90. The MPI library is used for parallelization.

The implementation relies on the separation of node constraints and enforcing the rest by Lagrange multipliers, as suggested already in Dohrmann [4]. One new aspect of the implementation is the use of reactions, which come naturally from the frontal solver, to avoid custom coding. An external parallel multifrontal solver MUMPS [1] is used for the solution of the coarse problem, instead of the serial frontal solver, as dimension of the coarse space could become a bottleneck.

Detailed description of the implementation can be found in [21], and some more experiments were presented in [20].

Recently, the proposed selection of corners has been implemented into the parallel solver, and the natural parallelism of the algorithm is fully exploited.

7. Numerical results

Presented numerical results were computed on SGI Altix 4700 computer with 1.5 GHz Intel Itanium 2 processors (OS Linux) in Czech Technical University Supercomputing Centre, Prague. For decompositions, we use the METIS graph partitioner [6].

Three different industrial problems have been tested. The first one is a problem of elasticity analysis of a turbine nozzle, through which the steam enters the turbine blades (Figure 5). The geometry is discretized using 2 696 quadratic elements, which leads to 40 254 unknowns. The second one is a problem of elasticity analysis of a hip joint replacement which is loaded by pressure from human body weight. This mesh consists of 33 186 quadratic elements resulting in 544 734 unknowns. Both meshes are divided into 36 subdomains by METIS. The turbine nozzle problem was computed using 12

processors, for hip joint replacement 36 processors were used. The third problem is stress analysis of a mine reel loaded by its own weight and the weight of the steel wire rope (Figure 6). The mesh consists of 140 816 quadratic elements and 1 739 211 unknowns. It was divided into 1 024 subdomains by METIS. Problem was computed using 32 processors. Decomposition characteristics of the three industrial problems are summarized in Table 1.

Three algorithms for selecting the basic set of corners are tested: Algorithm 1 from Section 5 referred to as *full*, modified Algorithm 1 described in Remark 1 in Section 5 referred to as *minimal*, and the edge-based algorithm mentioned in Section 5, inspired by [9] and described in [20], referred to as *edge*. The number of PCG iterations was chosen as a measure of quality of the BDDC preconditioning. Numbers of the basic sets of corners obtained by the three algorithms for the three problems are recorded in Table 2 and corresponding number of PCG iterations are summarized in Table 3. For the two smaller problems (turbine nozzle and hip joint replacement), either constraints on corners only (referred to as *C*), or constraints on corners and all averages (over all edges and faces) referred to as *C+E+F* are tested. For the larger problem of mine reel, corner constraints alone turned out to be too weak to achieve a reasonable convergence and the results are marked as 'n/a'. The edge-based algorithm did not work properly for hip joint replacement problem in the case of the basic set of corners only, so the results are missing too.

As the basic sets of corners selected by different algorithms have different numbers of corners, for a fair comparison of the algorithms we added more corners selected randomly from the interface to the smaller sets in order to achieve the same number of corners. Comparison of the algorithms using the same number of corner constraints is summarized in Table 4.

Interesting results are obtained by adding more randomly selected interface nodes as corners to the basic set in order to improve convergence (see Figures 8 – 10 left): it seems that the initial choice of the basic set influences the convergence properties even when many more randomly selected corners are added. Graphs on the right side of these figures show that the best computational time is achieved for higher numbers of corners than the basic sets for all problems tested and all algorithms for selecting the basic set used.

It can be observed especially on the most difficult problem of mine reel (Fig. 10), that the basic set of corners provided by the new algorithm in its full version is much more efficient than the basic sets provided by the earlier approaches and considerably reduces the computational time.

8. Conclusion

It has been observed on a number of practical computations by the BDDC method, that the effort to find the minimal set of corners might be misleading and selecting more corners often considerably improves the performance of the preconditioner and reduces the computational time. This behaviour can be explained for problems with complex interface by position of selected corners, which may be optimal with respect to one pair of subdomains, but may lead

to poorly conditioned problems for other subdomains. As a consequence, by favouring already selected corners, these subdomains are not given the freedom to select corners optimally distributed for their own fixation.

This has been the main motivation for presenting a new approach to selecting the basic set of corners, which is proposed in Section 5. It attempts to combine advantages of previous algorithms, and it is based on selection of corners independently for each face, so it can be naturally parallelized. It does not aspire to minimize the number of selected corners that assure the invertibility of all problems in BDDC and typically produces a larger initial set of coarse nodes than the other algorithms. We have seen this to be beneficial for all performed computations.

Numerical experiments on three industrial problems show that for basic sets of corners, this approach gives better results than the other two algorithms used for comparison in all three tested problems. When more corners are added, better results are obtained in two of the problems (turbine nozzle and mine reel) and comparable results in the third case (hip joint replacement).

We are aware that for very large problems the solution of the coarse problem might eventually dominate the computation and another approach than a (parallel) direct solver could be necessary. In such cases, multilevel extension of the BDDC method (e.g. [18]) seems to be a promising way. However, we observed even for the largest test problem of the mine reel, that we did not reach this computational bottleneck when adding more corners into the coarse problem, and the curve of computational time with respect to the number of corners was still decreasing. The expected bottleneck is also pushed farther by the everlasting advances in parallel direct solvers.

Acknowledgement

References

[1] Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., 2000. Multifrontal parallel distributed symmetric and unsymmetric solvers. Comput. Methods Appl. Mech. Engrg. 184, 501–520.

[2] Brenner, S. C., Sung, L.-Y., 2007. BDDC and FETI-DP without matrices or vectors. Comput. Methods Appl. Mech. Engrg. 196 (8), 1429–1435.

[3] Brož, J., Kruis, J., 2009. Heuristic algorithm for searching corner nodes in FETI-DP method. In: Modelling 2009, Rožnov pod Radhoštěm, June 22–26. Institute of Geonics AS CR, Ostrava, Czech Republic, pp. 30–31, (book of abstracts).

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

[5] Farhat, C., Lesoinne, M., Pierson, K., 2000. A scalable dual-primal domain decomposition method. Numer. Linear Algebra Appl. 7, 687–714.

[6] Karypis, G., Kumar, V., 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20 (1), 359–392.

[7] Klawonn, A., Rheinbach, O., 2006. A parallel implementation of dual-primal FETI methods for three dimensional linear elasticity using a transformation of basis. SIAM J. Sci. Comput. 28 (5), 1886–1906.

[8] Klawonn, A., Rheinbach, O., 2007. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. Comput. Methods Appl. Mech. Engrg. 196 (8), 1400–1414.

[9] Klawonn, A., Widlund, O. B., 2004. Selecting constraints in dual-primal FETI methods for elasticity in three dimensions. In: Kornhuber, R., Hoppe, R., Périaux, J., Pironneau, O., Widlund, O., Xu, J. (Eds.), Domain Decomposition Methods in Science and Engineering. Lecture Notes in Computational Science and Engineering. Springer, pp. 67–81, proceedings of the 15th International Conference on Domain Decomposition Methods in Berlin, Germany, 2003.

[10] Klawonn, A., Widlund, O. B., 2006. Dual-primal FETI methods for linear elasticity. Comm. Pure Appl. Math. 59 (11), 1523–1572.

[11] Klawonn, A., Widlund, O. B., Dryja, M., 2002. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. SIAM J. Numer. Anal. 40 (1), 159–179.

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

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

[14] Mandel, J., Dohrmann, C. R., Tezaur, R., 2005. An algebraic theory for primal and dual substructuring methods by constraints. Appl. Numer. Math. 54 (2), 167–193.

[15] Mandel, J., Sousedík, B., 2007. 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.

[16] Mandel, J., Sousedík, B., 2007. BDDC and FETI-DP under minimalist assumptions. Computing 81, 269–280.

[17] Mandel, J., Sousedík, B., 2011. Coarse space over the ages. In: Huang, Y., Kornhuber, R., Widlund, O., Xu, J. (Eds.), Domain Decomposition Methods in Science and Engineering XIX, Lecture Notes in Computational Science and Engineering 78, Part 2. Springer-Verlag, pp. 213–220.

[18] Mandel, J., Sousedík, B., Dohrmann, C. R., 2008. Multispace and multilevel BDDC. Computing 83 (2-3), 55–85.

[19] Mandel, J., Sousedík, B., Šístek, J., 2009. Adaptive BDDC in three dimensions. Submitted to Math. Comput. Simulation.

[20] Šístek, J., 2008. The finite element method in fluids: stabilization and domain decomposition. Ph.D. thesis, Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Mathematics.

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

[22] Toselli, A., Widlund, O. B., 2005. Domain Decomposition Methods— Algorithms and Theory. Vol. 34 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin.

[23] Widlund, O. B., 2009. The development of coarse spaces for domain decomposition algorithms. In: Bercovier, M., Gander, M., Kornhuber, R., Widlund, O. (Eds.), Domain Decomposition Methods in Science and Engineering XVIII. Vol. 70 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, proceedings of 18th International Conference on Domain Decomposition, Jerusalem, Israel, January 2008.
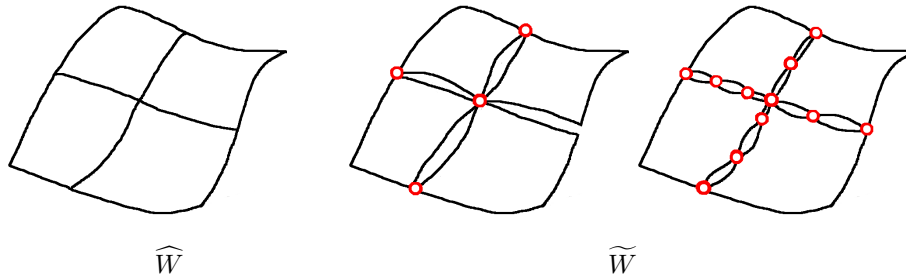
Figure 1: A schematic illustration of the continuity constraints: functions from $\widehat{W}$ are continuous across the interface (left), functions from $\widetilde{W}$ are continuous only at corners, marked by circles (centre and right, for two different choices of $\widetilde{W}$).
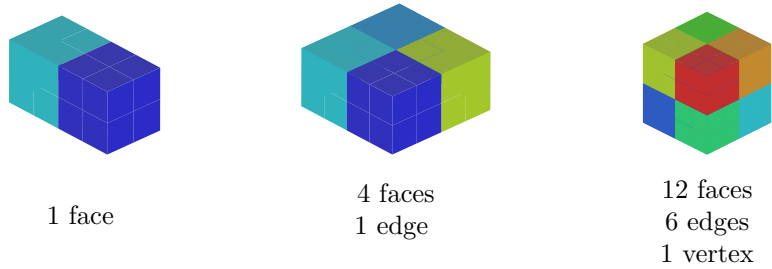

1 face

4 faces
1 edge

12 faces
6 edges
1 vertex

Figure 2: Examples of classification of the interface nodes as faces, edges and vertices according to *Definition 1*.

problem	subs.	vertices	edges	faces	intf. nodes	all nodes
Turbine nozzle	36	6	60	101	2 714	13 418
Hip replacement	36	1	19	78	9 222	181 578
Mine reel	1 024	2 451	1 209	4 164	117 113	579 737

Table 1: Decomposition characteristics of the tested problems.
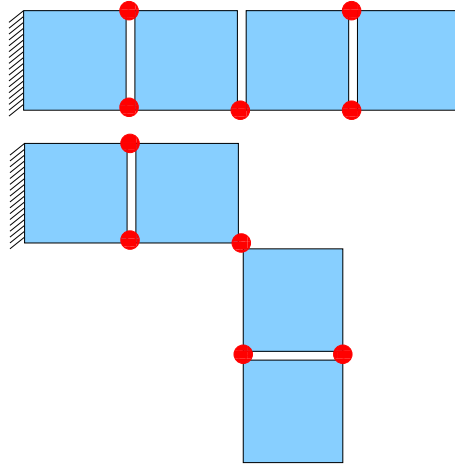
Figure 3: A 2D example of mechanism in the coarse problem for serial division into four subdomains, red dots denote corners.
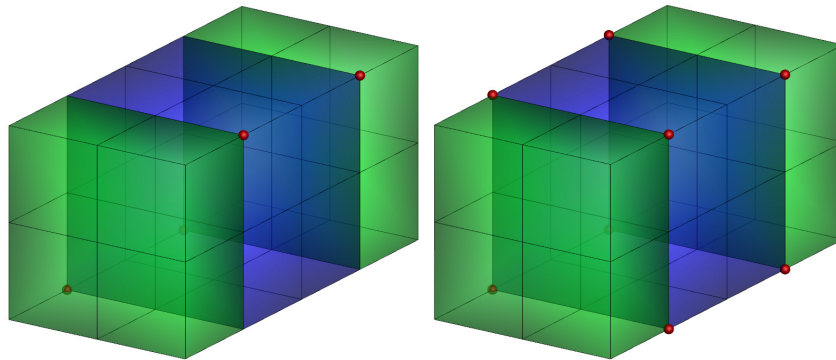

Figure 4: Example of a problem with one subdomain disconnected. Four corners obtained by algorithm without detection of components (left), and eight corners obtained with detection of components of interface (right).

problem	full	min	edge
Turbine nozzle	218	145	115
Hip replacement	227	189	66
Mine reel	7 864	6 183	4 152

Table 2: Number of corners in the basic set selected by different algorithms.
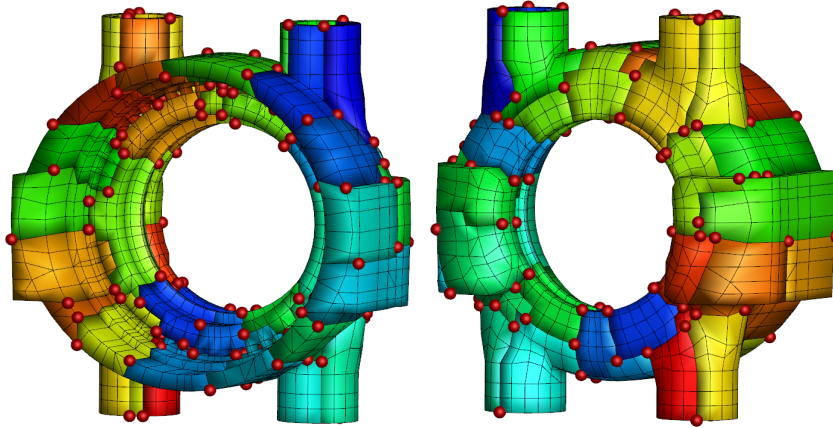
Figure 5: Turbine nozzle problem, 36 subdomains, initial set of 218 corners selected by the full version of Algorithm 1 marked by balls.
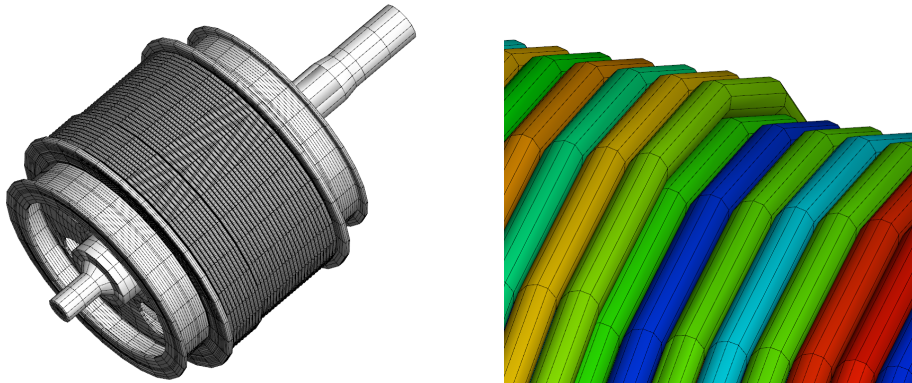

Figure 6: Mine reel problem, finite element mesh (left) and a detail of the steel rope with division into subdomains (right).
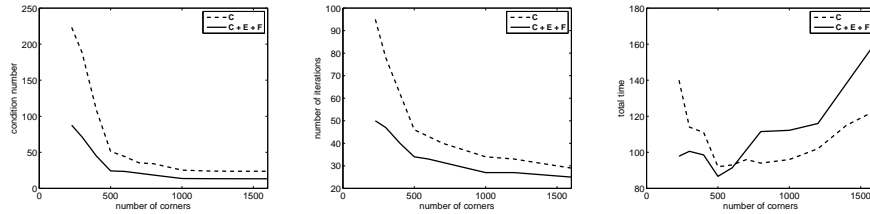

Figure 7: Typical dependence of the condition number (left), the number of iterations of the PCG (centre), and the total computational time (right) on the number of corner constraints. Dashed line - corner constraints only, full line - corner constraints and all face and edge averages. Hip joint replacement, 33 186 quadratic elements, 36 subdomains.

17

	C			C+E+F		
	full	min	edge	full	min	edge
Turbine nozzle	38	49	73	24	27	29
Hip replacement	95	99	n/a	50	52	n/a
Mine reel	n/a	n/a	n/a	935	1 841	4 637

Table 3: Number of PCG iterations needed for convergence for different algorithms of selecting the basic set of corners and different constraint type.

	C			C+E+F		
	full	min	edge	full	min	edge
Turbine nozzle	38	41	42	24	25	26
Hip replacement	95	91	> 138	50	50	61
Mine reel	n/a	n/a	n/a	935	1 674	≈1 800

Table 4: Number of PCG iterations needed for convergence for different algorithms of selecting the basic set of corners and different constraint type. For every problem, different basic sets were completed to the same number of corners by adding randomly selected corners.
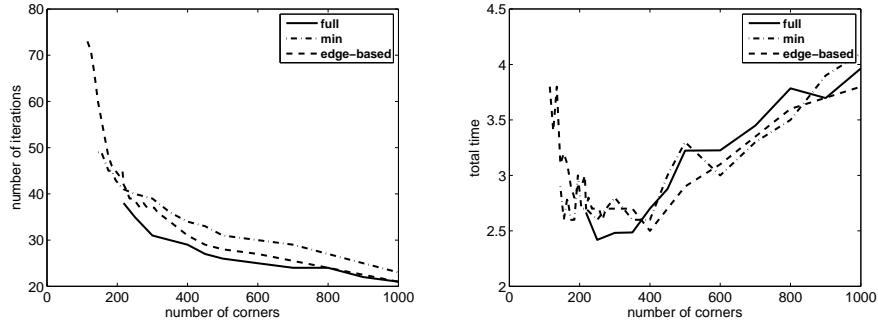

Figure 8: Turbine nozzle problem, 36 subdomains, corner constraints only. Dependence of the number of iterations (left) and the total computational time (right) on the number of corner constraints. Full line - full version of the Algorithm 1, dash-dotted line - minimalistic version, dashed line - the edge based algorithm.
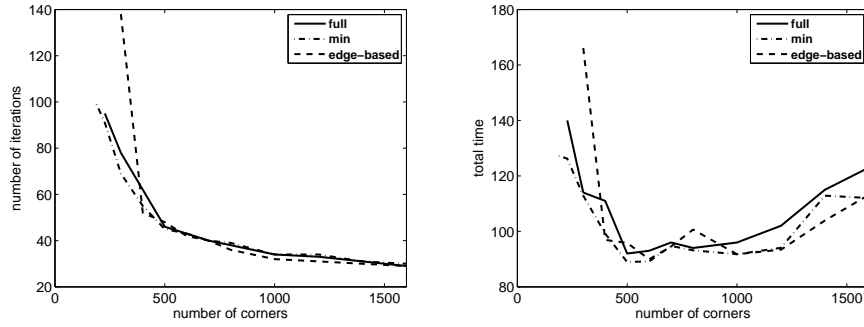
Figure 9: Hip joint replacement problem, 36 subdomains, corner constraints only. Dependence of the number of iterations (left) and the total computational time (right) on the number of corner constraints. Full line - full version of the Algorithm 1, dash-dotted line - minimalistic version, dashed line - the edge based algorithm.
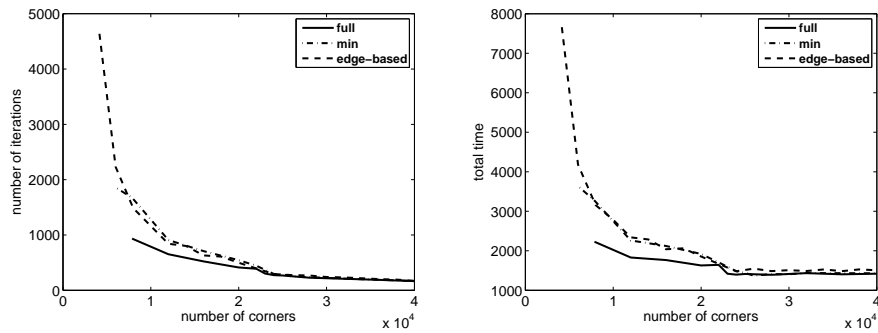

Figure 10: Mine reel problem, 1024 subdomains, corner and all edge and face constraints. A dependence of the number of iterations (left) and the total computational time (right) on the number of corner constraints. Full line - full version of the Algorithm 1, dash-dotted line - minimalistic version, dashed line - the edge based algorithm.

19