Conference Applications of Mathematics 2012
in honor of the 60th birthday of Michal Křížek.
Institute of Mathematics AS CR, Prague 2012

SOME PRACTICAL ASPECTS OF PARALLEL ADAPTIVE BDDC METHOD

Jakub Šístek[1,2], Jan Mandel[3], Bedřich Sousedík[4]

[1] Institute of Mathematics AS CR
Žitná 25, Prague, CZ 115 67, Czech Republic
jakub.sistek@math.cas.cz
[2] Department of Mathematics, FME CTU
Karlovo náměstí 13, Prague, CZ 121 35, Czech Republic
[3] Department of Mathematical and Statistical Sciences, UCD
Denver, CO 80217-3364, USA
jan.mandel@ucdenver.edu
[4] Department of Aerospace and Mechanical Engineering, USC
Los Angeles, CA 90089-2531, USA
sousedik@usc.edu

Abstract

We describe a parallel implementation of the Balancing Domain Decomposition by Constraints (BDDC) method enhanced by an adaptive construction of coarse problem. The method is designed for numerically difficult problems, where standard choice of continuity of arithmetic averages across faces and edges of subdomains fails to maintain the low condition number of the preconditioned system. Problems of elasticity analysis of bodies consisting of different materials with rapidly changing stiffness may represent one class of such challenging problems. The adaptive selection of constraints is shown to significantly increase the robustness of the method for this class of problems. However, since the cost of the set-up of the preconditioner with adaptive constraints is considerably larger than for the standard choices, computational feasibility of the presented implementation is obtained only for large contrasts of material coefficients.

1. Introduction

The *Balancing Domain Decomposition by Constraints* (BDDC) was developed by Dohrmann [3] as a primal alternative to the *Finite Element Tearing and Interconnecting - Dual, Primal* (FETI-DP) by Farhat et al. [4]. Both methods use constraints to impose equality of new coarse variables on substructure interfaces, such as values at substructure corners or averages over edges and faces.

FETI-DP and BDDC are quite robust. However, the condition number may deteriorate in certain situations of practical importance. Typical difficulties include

rough interfaces of subdomains introduced by automatic partitioning of the mesh, or problems with strong discontinuities of coefficients. A recent comprehensive study of problems from the last group is presented in [10, 11]. It is shown there, that while some configurations of jumps do not present problems for robustness of the method, other configurations may lead to very poor performance. Such a troublesome configuration is typically faced when the domain of considerably different coefficient cuts through the faces of subdomains. For such complicated problems, a better selection of constraints is desirable. Enriching the coarse space so that the iterations run in a subspace devoid of 'difficult' modes has been a successful trick in earlier iterative substructuring methods as well as multigrid methods. For BDDC and FETI-DP, the adaptive enrichment was first proposed in [8]. In [9], generalization to three-dimensional problems formulated in terms of BDDC operators was given.

Here, we build on top of our results in [9]. We focus on study of the behaviour of the adaptive method for problems with jumps in coefficients. We consider a model problem of nonlinear elasticity analysis of a cube containing rods of much larger stiffness. Moreover, some of these rods are punching through faces of subdomains. It is shown, that performance of the standard BDDC method deteriorates quite fast with increasing contrast, that corresponds to the ratio of the coefficients. On the other hand, adaptive constraints reduce the condition number considerably and thus significantly improve the robustness of the method with respect to jumps in coefficients. We study the cost of this approach in connection with our current implementation of BDDC method. As the cost of computing these averages keeps rather large, it is shown that regarding computational time, this advanced selection of constraints is beneficial only for quite large contrasts compared to the standard BDDC method, provided the latter converge. It is also demonstrated, that the enhanced method is able to produce a solution also for such difficult cases, for which standard BDDC does not converge at all.

2. Iterative substructuring

As the main purpose of this contribution is to report on experience with using of adaptive BDDC rather than to provide a self contained description of the method, iterative substructuring and BDDC are only briefly recalled in this and the next section, respectively. Reader is kindly referred to [9] for details. We follow the notation introduced in that paper throughout this contribution.

Consider an elliptic boundary value problem defined on a bounded domain $\Omega \subset \mathbb{R}^3$ and discretized by conforming finite elements. The domain Ω is decomposed into N nonoverlapping *subdomains* Ω_i, $i = 1, \dots N$, also called *substructures*. The nodes contained in more than one substructure are called the *interface*, denoted by Γ, and $\Gamma_i = \Gamma \cap \Omega_i$ is the interface of substructure Ω_i. The interface Γ may also be classified as the union of three different types of nonoverlapping sets: *faces*, *edges*, and *corners*. A *face* contains all nodes shared solely by one pair of subdomains,

an *edge* contains all nodes shared by same set of more than two subdomains, and a *corner* is a degenerate edge with only one node.

We identify finite element functions with the vectors of their coefficients in the standard finite element basis. These coefficients are also called *variables* or *degrees of freedom*. We also identify linear operators with their matrices, in bases that will be clear from the context.

The space of all finite element functions on subdomain Ω_i is denoted by W_i, and let

$$W = W_1 \times \cdots \times W_N. \tag{1}$$

The space W is equipped with the standard $\mathbb{R}^n$ basis and the Euclidean inner product $\langle w, v\rangle = w^{\mathrm{T}} v$. For a symmetric positive semidefinite matrix M, $\langle u, v\rangle_M = \langle Mu, v\rangle$, and $\|u\|_M = \langle Mu, u\rangle^{1/2}$.

Let $A_i : W_i \to W_i$ be the local substructure stiffness matrix, obtained by the subassembly of element matrices only in substructure Ω_i. The matrices A_i are symmetric positive semidefinite for an elliptic problem. We can write vectors and matrices in the block form

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}, \quad w \in W, \quad A = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix} : W \to W. \tag{2}$$

Now let $U \subset W$ be the space of all functions from W that are continuous across substructure interfaces. We are interested in solving the problem

$$u \in U : \quad \langle Au, v\rangle = \langle f, v\rangle, \quad \forall\, v \in U\,, \tag{3}$$

where $f \in W$ is a given right-hand side. This problem corresponds to the standard solution of an elliptic partial differential equation discretized by the finite element method.

Remark 1. *Let matrix R be the global-to-local mapping that restricts the global vectors of degrees of freedom to local degrees of freedom on each Ω_i. Then $R^{\mathrm{T}} A R$ is the global stiffness matrix, and (3) is equivalent to the assembled system*

$$R^{\mathrm{T}} A R \mathbf{u} = R^{\mathrm{T}} f, \tag{4}$$

where $\mathbf{u}$ is the coefficient vector satisfying $u = R\mathbf{u}$.

Denote by $U_I \subset W$ the space of all (vectors of) finite element functions with nonzero values only in the interiors of substructures Ω_i. Then $U_I \subset U$, and the space W is decomposed as the A-orthogonal direct sum

$$W = U_I \oplus W_H, \quad U_I \perp_A W_H, \tag{5}$$

where the functions from W_H are called *discrete harmonic*. The A-orthogonal projection onto U_I is denoted by

$$P : W \to U_I. \tag{6}$$

The space of all discrete harmonic functions from W that are continuous at interface is denoted by $\widehat{W}$. We have

$$\widehat{W} = W_H \cap U = (I - P)\, U, \tag{7}$$

and the A-orthogonal decomposition

$$U = U_I \oplus \widehat{W}, \quad U_I \perp_A \widehat{W}. \tag{8}$$

The solution $u \in U$ of problem (3) is split as

$$u = u_I + w, \quad u_I \in U_I,\ w \in \widehat{W}. \tag{9}$$

Solving for the interior component $u_I \in U_I$ decomposes into N independent Dirichlet problems. We are interested in finding the discrete harmonic component $w \in \widehat{W}$, which is the solution of the reduced problem

$$w \in \widehat{W} : \quad \langle Aw, z\rangle = \langle f, z\rangle\,, \quad \forall z \in \widehat{W}. \tag{10}$$

Problem (10) is solved by a Krylov subspace method, e.g. preconditioned conjugate gradient method (PCG) in the case of symmetric positive definite matrix, and the BDDC serves as a preconditioner for this method.

Let us briefly summarize the main steps of iterative substructuring, details may be found e.g. in [14].

Algorithm 1. (Iterative substructuring)**.** *Problem (3) is solved indirectly by the following steps:*

1. *Prepare reduced problem (10), i.e. factorize the matrices of local Dirichlet problems on each subdomain. This formally corresponds to construction of* Schur complement *and* reduced right-hand *side with respect to the interface* Γ *except that the former is not constructed explicitly. This step is inherently parallel.*

2. *Solve problem (10) by a Krylov subspace method.*

3. *Reconstruct the whole solution of problem (3) on each subdomain by solving the local Dirichlet problems. This step is inherently parallel.*

3. Balancing domain decomposition by constraints

Let us further define an averaging operator

$$E : W \to U, \tag{11}$$

which is a projection from W onto U. Then the operator

$$(I - P)\, E : W \to \widehat{W} \tag{12}$$

is a projection from W onto $\widehat{W}$. Its evaluation consists of averaging between the substructures, followed by the discrete harmonic extension from the substructure boundaries. Note that

$$(I - (I - P)\, E)\, w = (I - P)\, (I - E)\, w, \quad \forall w \in W_H, \tag{13}$$

since $Pw = 0$ if $w \in W_H$.

Proper weights (e.g. proportional to the substructure stiffness) in the averaging given by E are important for the performance of BDDC (as well as other iterative substructuring methods) independent of different stiffness of substructures [5, 7]. The scaled operator E takes care of the case of jumps of coefficients, when these are constant on each subdomain. A detailed discussion of construction of the averaging operator E can be found in [1].

The BDDC preconditioner is characterized by a selection of *coarse degrees of freedom,* such as values at corners and averages over edges or faces. We are mainly interested in construction of efficient weighted averages on faces of subdomains in the rest of this paper. For selection of corners, we use the face-based algorithm described in [12] and we use arithmetic averages on edges.

The action of the BDDC preconditioner is then defined in the space given by the requirement that the coarse degrees of freedom on adjacent substructures coincide, which is enforced in the algorithm by *constraints*. So, the design of the BDDC preconditioner is characterized by a selection of an intermediate space $\widetilde{W}$ satisfying these constraints,

$$\widehat{W} \subset \widetilde{W} \subset W_H. \tag{14}$$

We formally define the space $\widetilde{W}$ by enforcing the constraints on continuity weakly by a matrix D, in which each row defines one constraint,

$$\widetilde{W} = \{w \in W_H : \; Dw = 0\}\,. \tag{15}$$

An application of BDDC preconditioner can be described as solving the original variational problem in the space $\widetilde{W}$.

$$M_{BDDC} : r \mapsto u = (I - P)\, Ew, \quad w \in \widetilde{W} : \quad \langle Aw, z\rangle = \langle r, (I - P)\, Ez\rangle\,, \quad \forall z \in \widetilde{W}, \tag{16}$$

where r is the residual in the PCG method.

The following condition number bound for BDDC will play an essential role in our design of the adaptive method.

Theorem 1. ([7]). *The eigenvalues of the preconditioned operator of the BDDC method satisfy* $1 \leq \lambda \leq \omega_{BDDC}$, *where*

$$\omega_{BDDC} = \sup_{w \in \widetilde{W}} \frac{\|(I-(I-P)\,E)\,w\|_A^2}{\|w\|_A^2}. \tag{17}$$

4. Adaptive selection of constraints

The condition number bound ω_{BDDC} from Theorem 1. equals to the maximum eigenvalue λ_1 of the associated generalized eigenvalue problem

$$w \in \widetilde{W}: \quad \langle (I-(I-P)\,E)\,w, (I-(I-P)\,E)\,z\rangle_A = \lambda\,\langle w, z\rangle_A\,, \quad \forall z \in \widetilde{W}. \tag{18}$$

Since the bilinear form on the left-hand side of (18) is symmetric positive semi-definite and the bilinear form on the right-hand side is symmetric positive definite, application of the Courant-Fisher-Weyl minimax principle (cf. e.g. [2, Theorem 5.2]) leads to the following theorem.

Theorem 2.. *The generalized eigenvalue problem (18) has eigenvalues* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$. *Denote the corresponding eigenvectors* w_ℓ. *Then, for any* $k = 1, \ldots, n-1$, *and any linear functionals* L_ℓ *on* W, $\ell = 1, \ldots, k$,

$$\max\left\{ \frac{\|(I-(I-P)\,E)\,w\|_A^2}{\|w\|_A^2} : w \in \widetilde{W},\ L_\ell\,(w) = 0, \quad \forall \ell = 1, \ldots, k \right\} \geq \lambda_{k+1}, \tag{19}$$

with equality if

$$L_\ell\,(w) = \langle (I-(I-P)\,E)\,w_\ell, (I-(I-P)\,E)\,w\rangle_A\,. \tag{20}$$

It follows that the optimal decrease of the condition number bound (17) can be achieved by adding the rows d_ℓ^{T} defined by $d_\ell^{\mathrm{T}} w = L_\ell\,(w)$ to the constraint matrix D in the definition of $\widetilde{W}$ (15). However, solving the global eigenvalue problem (18) would be prohibitively expensive. For this reason, we replace (18) by a collection of local problems, each defined by considering only two adjacent subdomains Ω_i and Ω_j at a time. Subdomains are considered adjacent if they share an edge in 2D, or a face in 3D. All quantities associated with such pair will be denoted by the subscript $_{ij}$. Using also (13), the local generalized eigenvalue problem (18) becomes

$$\begin{aligned} &w_{ij} \in \widetilde{W}_{ij}: \\ &\langle (I-P_{ij})\,(I-E_{ij})\,w_{ij}, (I-P_{ij})\,(I-E_{ij})\,z_{ij}\rangle_{A_{ij}} = \lambda\,\langle w_{ij}, z_{ij}\rangle_{A_{ij}}\,, \forall z_{ij} \in \widetilde{W}_{ij}. \end{aligned} \tag{21}$$

The space $\widetilde{W}_{ij}$ is constructed with respect to an initial set of constraints on continuity between subdomains i and j. The starting point used in this paper is continuity at corners and of arithmetic averages on edges. We assume that initial constraints are already sufficient to prevent relative rigid body motions of any pair of adjacent substructures. The maximal eigenvalue of (21) is denoted $\omega_{ij} = \lambda_{ij,1}$.

Definition 1. (Condition number indicator). *The heuristic* condition number indicator $\widetilde{\omega}$ *is defined as*

$$\widetilde{\omega} = \max \{\omega_{ij} : \Omega_i \text{ and } \Omega_j \text{ are adjacent}\} . \tag{22}$$

Considering two adjacent subdomains Ω_i and Ω_j only, we construct the added constraints $L_\ell(w) = 0$ from (20) as

$$\langle (I - P_{ij})(I - E_{ij}) w_{ij,\ell}, (I - P_{ij})(I - E_{ij}) w \rangle_{A_{ij}} = 0, \quad \forall \ell = 1, \ldots, k_{ij}, \tag{23}$$

where $w_{ij,\ell}$ are the eigenvectors corresponding to the k_{ij} largest eigenvalues from (21). These constraints then form additional rows of matrix D.

Algorithm 2. (Adaptive BDDC). *Find the smallest k_{ij} for each pair of adjacent substructures Ω_i and Ω_j to guarantee that $\lambda_{ij,k_{ij}+1} \leq \tau$, where τ is a given tolerance, and add the constraints (23) to the definition of $\widetilde{W}$.*

Remark 2. *The adaptive BDDC method assures that the condition number indicator $\widetilde{\omega} \leq \tau$ with the minimum number of added constraints. However, our theory does not cover the correspondence between the indicator $\widetilde{\omega}$ and actual condition number of the preconditioned operator ω_{BDDC} (due to the localization of the global eigenvalue problem). In [13], it has been shown on a number of experiments that the indicator agrees very well with the global eigenvalue. It is a subsequent goal also of this paper to study their relation on experiments.*

5. Parallel implementation

In this section, we describe some details of the approach to the parallel implementation of the adaptive selection of constraints, as it has been recently implemented as an optional feature of our publicly available solver based on BDDC method, the BDDCML[1] package. This package is written in Fortran 95 programming language and parallelized using Message Passing Interface (MPI) library.

As was described in the previous section, the main additional work compared to the standard BDDC method comprises solving a large number of generalized eigenproblems (21), one for each pair of subdomains sharing a face. From the point of view of parallel computing, this immediately presents a complication, because the layout of pairs can be quite different from the natural layout of parallel domain decomposition computation based on distribution by subdomains. This issue is addressed in the implementation by a separate assignment of eigenproblems to processors, which is independent of distribution of subdomains.

Note further, that computing the energy scalar products on both sides of problem (21) corresponds (note the space of discrete harmonic functions) to computing with Schur complement formed by assembling the Schur complements of subdomains i and j, the S_{ij}. The responsibility of computing with these two components

[1] `http://www.math.cas.cz/~sistek/software/bddcml.html`

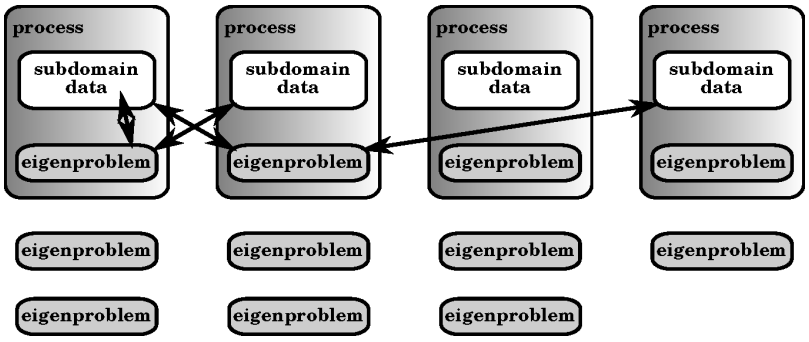

Figure 1: An example of a possible parallel layout of local eigenproblems with communication pattern marked for two of them.

is clearly on the processes which hold subdomains i and j. However, not only that we do not want to send these dense matrices to the process responsible for the ij–eigenproblem via network, but these Schur complements are not available explicitly. Instead, only multiplication of a vector by them is prepared in the form of local Dirichlet problems on subdomains i and j. This in turn implies that we need a method for solving generalized eigenproblems which requires only applications of the operators on both the left- and right- hand sides on vectors. A recent algorithm that meets these requirement is the LOBPCG method [6], and its C implementation is available in the BLOPEX package.

Thus, the process responsible for computing a local ij–eigenproblem sends parts of the vector for multiplication to processes responsible for subdomains i and j, which formally multiply these vectors by S_i and S_j by solving respective Dirichlet problems, and then send the results back to initial process. This scheme may lead to quite complicated communication pattern for general situations (see Figure 1 for a rather simple example). To somewhat simplify this pattern, the eigenproblems are solved in turns. In each turn, at most as many eigenproblems are solved as is the number of available processors. In the beginning of each turn, a new communication pattern is determined, so that each process knows if it is responsible for an eigenvalue computation and for which other processes it will compute the applications of local Schur complements. Then, the iterative solution is started. Although some eigenproblems may require lower number of iterations than others, no other eigenproblem is solved until all problems in the turn are finished, after which a new group of eigenproblems is set-up and processed.

The purpose of *Adaptive BDDC* method is enhancing the robustness of BDDC for numerically complicated problems. However, it has turned out that for these problems, convergence of LOBPCG may be very slow without preconditioning. Thus, we apply a preconditioner for the LOBPCG method, which is nothing else than the BDDC preconditioner localized to the ij–pair. This preconditioner is build on top of existing components of BDDC on each subdomain and follows the same communication pattern as the multiplication by local Schur complements described above,

thus does not present much overhead. This strategy was suggested and studied in [13] on number of experiments.

The LOBPCG method computes only selected number of dominant eigenvalues and corresponding eigenvectors. As the computational cost quickly grows with adding these vectors, we have limited their value to 10 or 15 in our computations. Maximum number of LOBPCG iterations was limited to 15 and tolerance on residual set to 10^{-9}. See [13] for details of generation of adaptive constraints.

The BDDCML solver employs the serial MUMPS solver for factorizations of both subdomain problems (corresponding to Dirichlet problem and the saddle-point Neumann problem, see e.g. [3]). A parallel instance of the MUMPS solver is then applied to solution of the explicit global coarse problem.

6. Numerical results

To study robustness of the adaptive BDDC method, we have selected a model problem of elasticity analysis of a unit cube made of soft material with Young modulus E_1 which contains nine stiff rods with Young modulus E_2. We study the behaviour of the BDDC method with respect to value of 'contrast' of coefficients, which is defined here as the ratio E_2/E_1. In our test, we keep $E_2 = 2.1 \cdot 10^{11}$ fixed and compute E_1 based on the desired contrast.

The nonlinear elasticity model with St. Venant–Kirchhoff constitutive law is used. The considered loading of the problem was small enough so that a single nonlinear iteration was sufficient to converge to the final solution. A linearized problem (of one nonlinear iteration) is symmetric positive definite and is passed to the BDDCML solver.

The cube if fixed at one face orthogonal to the stiff rods and loaded by its own weight. The domain is discretized by 32^3, 64^3, and 128^3 tri-linear cubic elements. Regular divisions into 2^3 and 4^3 subdomains are used. This leads to the following four tested cases: (i) 8 subdomains, $H/h = 16$, (ii) 8 subdomains, $H/h = 32$, (iii) 64 subdomains, $H/h = 16$, (iv) 64 subdomains, $H/h = 32$. Due to the set-up, four stiff rods are punching through faces, which creates one of the least favourable situations from the point of view of the BDDC or FETI-DP method [11]. Other four stiff rods are attached to faces and one rod is placed along edges of subdomains in the centre of the cube.

Figure 2 shows a division of the cube into 8 subdomains and an example of a deformed shape of the cube with contrast 10. Presented computations were performed on the SGI Altix UV machine in the supercomputing centre in Prague. One core of a CPU was used per subdomain.

For all four tested configurations, we investigate dependence of convergence on the contrast of coefficients E_2/E_1. Dependence of condition number is presented in Figure 3, dependence of number of iterations is presented in Figure 4, and finally computational times are presented in Figure 5. Each of the cases was computed using standard BDDC with arithmetic averages on all edges and faces, and adaptive

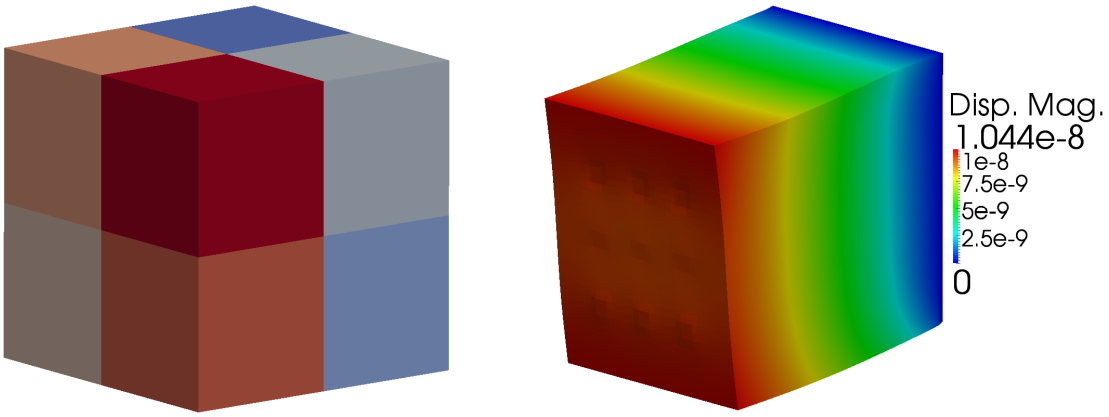

Figure 2: Example of division of the cube into 8 subdomains (left) and (magnified) deformed shape for contrast $E_2/E_1 = 10$ coloured by magnitude of displacement (right).

BDDC with arithmetic averages on edges. Maximum number of adaptive constraints on a face was set to 10 and 15 and tolerance τ in Algorithm 2. was set to 1.5.

The condition numbers reported in Fig. 3 are estimates from the Lanczos sequence in conjugate gradients. We can see how fast the condition number of the standard BDDC method grows with contrast–the growth is asymptotically linear with unit slope. The adaptive approach improves the condition number significantly. However the tolerance applies only for contrast around 10^3 (10 constraints limit) or 10^4 (15 constraints limit). Around this point, the adaptive algorithm saturates the limit of number of constraints on a face for all faces and, consequently, is unable to keep the condition number low. Until this point, the curves have a plateau, but tend to grow quickly when reaching the 'saturation point'. It is interesting to notice that while the indicator of condition number $\widetilde{\omega}$ (see (22)) approximately follows the estimated resulting condition number for case of eight subdomains, these curves departure for the case of 64 subdomains, the indicator being too pessimistic for 10 constraints, and too optimistic for 15. This effect is probably related to different positioning of the nine stiff rods with respect to the interface.

Similar conclusions can be done based on Fig. 4 with numbers of iterations. Adaptive method is able to keep these numbers independent up to the 'saturation point', from which a (delayed) growth begins, being milder for the case of 64 subdomains.

Perhaps the most interesting graphs are those for computational time presented in Fig. 5. Here we can see the typical behaviour of the standard BDDC method on this type of problems - while having a very cheap set-up of the preconditioner, the overall time is determined by the time spent in PCG iterations and grows hand in hand with this number. It is also worth noting, that we reached stagnation of PCG for contrast larger than 10^8 and 10^5 for 8 and 64 subdomains, respectively. The situations looks very different for the adaptive BDDC: we pay a large fixed cost when solving the local eigenproblems as part of the set-up. As the number of iterations in adaptive BDDC is kept quite low compared to the standard BDDC,

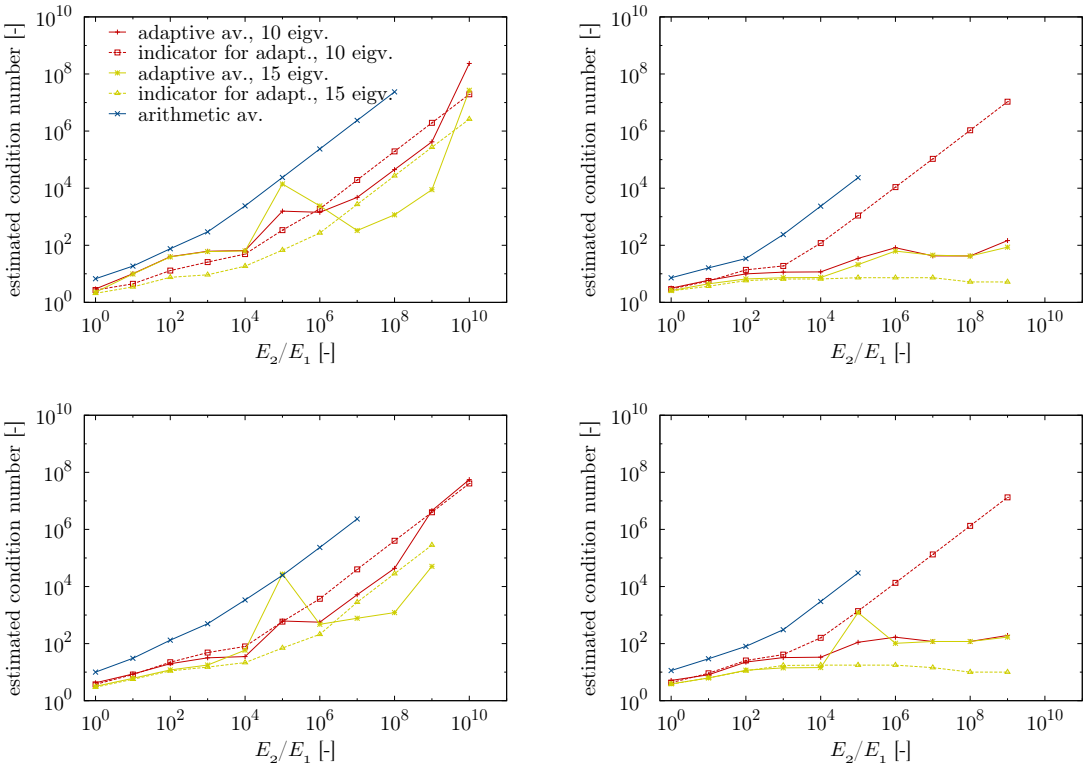

Figure 3: Estimated condition number with respect to contrast of coefficients; 32^3 elements, $H/h = 16$ (top left); 64^3 elements, $H/h = 16$ (top right); 64^3 elements, $H/h = 32$ (bottom left); 128^3 elements, $H/h = 32$ (bottom right); 'av.' = averages, '10 eigv.' = at most 10 eigenvectors are used per face, '15 eigv.' = at most 15 eigenvectors are used per face, 'indicator for adapt.' stands for the predicted condition number based on adaptivity indicator $\widetilde{\omega}$ from (22).

the overall time spent by adaptive BDDC is clearly dominated by the set-up phase, regardless of the contrast. In fact, the cost slightly decreases for increasing contrast which is probably related to faster convergence of LOBPCG method. We can also see, that for 15 constraints, the set-up is yet significantly more expensive than for 10 constraints, without a rewarding improvement of number of iterations. We can conclude a general statement, that while adaptive BDDC is far too costly for simple problems at lower contrast, there is a break-point, from which it becomes faster than the standard BDDC method. This break-point is around contrast 10^5–10^6 in presented computations. It should be also emphasised, that for higher contrasts, we were only able to obtain results by adaptive BDDC.

7. Conclusion

We have presented a description of a parallel implementation of the *Adaptive BDDC* method and a study of its robustness with respect to jumps in material

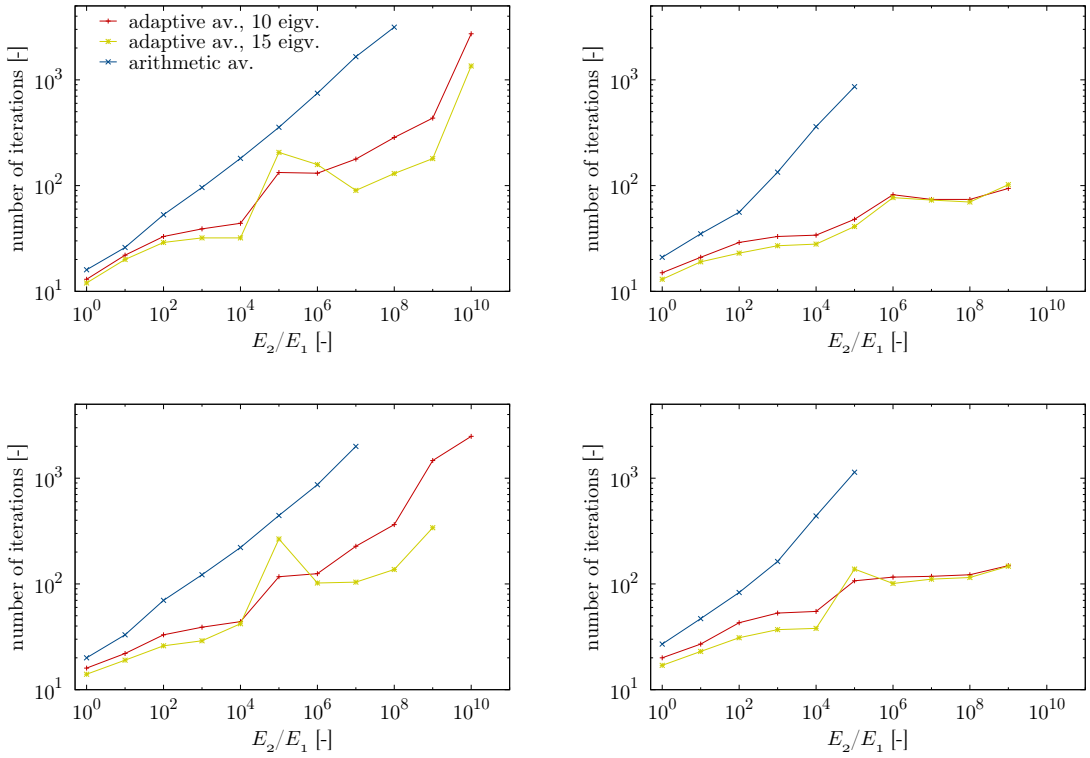

Figure 4: Number of iterations with respect to contrast of coefficients; 32^3 elements, $H/h = 16$ (top left); 64^3 elements, $H/h = 16$ (top right); 64^3 elements, $H/h = 32$ (bottom left); 128^3 elements, $H/h = 32$ (bottom right).

parameters. The parallel implementation faces some difficulties related mainly to the fact, that parallel layout of the faces among subdomains generally differs from the natural layout in domain decomposition with subdomains distributed among processors. This fact complicates an efficient parallel solution of local generalized eigenproblems used for construction of weighted averages as constraints in BDDC, and this part typically dominates the whole computational time of the method.

Nevertheless, despite its high cost, it has been shown on a model problem, that the method significantly improves robustness of the BDDC method, allowing for solution of problems with much larger contrasts than are manageable by the standard BDDC with arithmetic averages. For all tested problems, there has been a limiting contrast, from which computing by standard BDDC was either not possible, or was more expensive than computing by adaptive BDDC.

It has been shown, that while application of the adaptive BDDC to problems with low contrasts would be very inefficient and standard BDDC should be preferred, the method offers an interesting and competitive approach to handle problems with very large contrasts of coefficients.

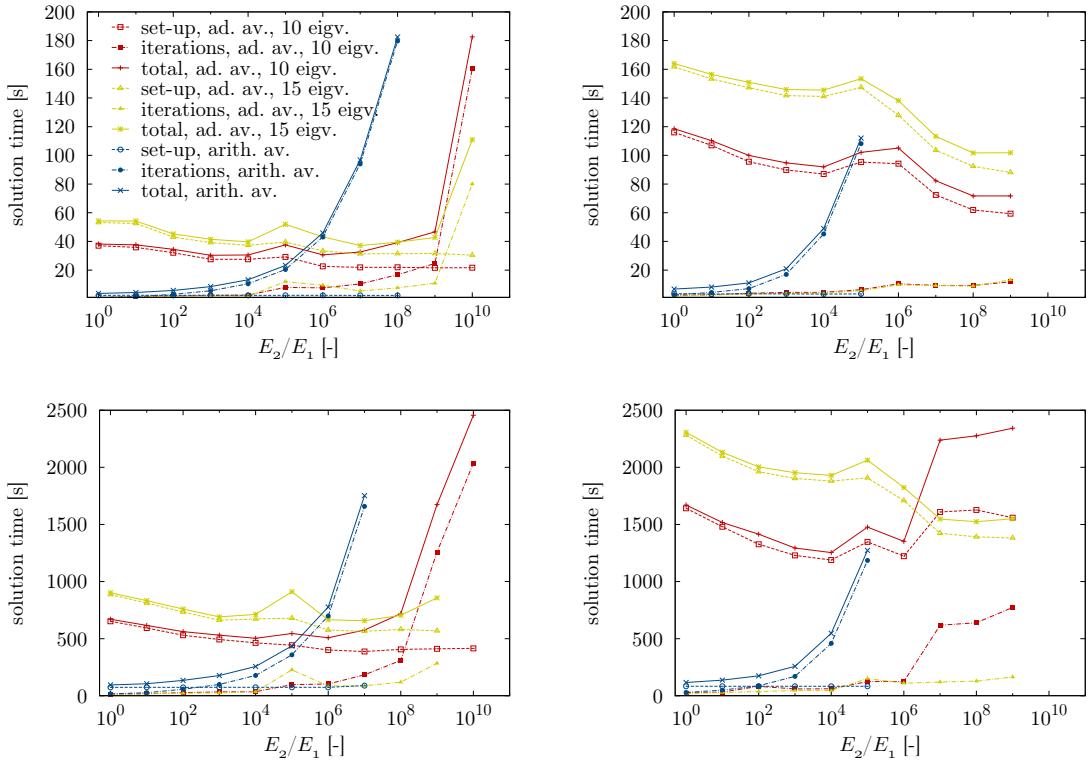

Figure 5: Computational time with respect to contrast of coefficients; 32^3 elements, $H/h = 16$ (top left); 64^3 elements, $H/h = 16$ (top right); 64^3 elements, $H/h = 32$ (bottom left); 128^3 elements, $H/h = 32$ (bottom right); 'ad. av.' = adaptive averages, 'arith. av.' = arithmetic averages, '10 eigv.' = at most 10 eigenvectors are used per face, '15 eigv.' = at most 15 eigenvectors are used per face.

Acknowledgements

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by Czech Science Foundation under project 106/08/0403, and by National Science Foundation under grant DMS-0713876. Bedřich Sousedík acknowledges support from DOE/ASCR. The research was started during two visits of Jakub Šístek at the University of Colorado Denver.

References

[1] Čertíková, M., Burda, P., Novotný, J., and Šístek, J.: Some remarks on averaging in the BDDC method. In: *Proceedings of Programs and Algorithms of Numerical Mathematics 15, Dolní Maxov, Czech Republic, June 6–11, 2010*, pp. 28–34. Institute of Mathematics AS CR.

[2] Demmel, J. W.: *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

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

[4] Farhat, C., Lesoinne, M., and Pierson, K.: A scalable dual-primal domain decomposition method. Numer. Linear Algebra Appl. **7** (2000), 687–714.

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

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

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

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

[9] Mandel, J., Sousedík, B., and Šístek, J.: Adaptive BDDC in three dimensions. Math. Comput. Simulation, doi:10.1016/j.matcom.2011.03.014. Available online.

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

[11] Pechstein, C. and Scheichl, R.: Analysis of FETI methods for multiscale PDEs - Part II: interface variations. Numer. Math. **118** (2011), 485–529.

[12] Šístek, J., Čertíková, M., Burda, P., and Novotný, J.: Face-based selection of corners in 3D substructuring. Math. Comput. Simulation, doi: 10.1016/j.matcom.2011.06.007. Available online.

[13] Sousedík, B.: *Adaptive-multilevel BDDC*. Ph.D. thesis, University of Colorado Denver, Department of Mathematical and Statistical Sciences, 2010.

[14] Toselli, A. and Widlund, O. B.: Domain decomposition methods–algorithms and theory. *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin, 2005.