Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

ON SELECTION OF INTERFACE WEIGHTS IN DOMAIN DECOMPOSITION METHODS

Marta Čertíková[1], Jakub Šístek[2,1], Pavel Burda[1]

[1] Czech Technical University
Technická 4, Prague, Czech Republic
marta.certikova@fs.cvut.cz, pavel.burda@fs.cvut.cz
[2] Institute of Mathematics AS CR
Žitná 25, Prague, Czech Republic
sistek@math.cas.cz

Abstract

Different choices of the averaging operator within the BDDC method are compared on a series of 2D experiments. Subdomains with irregular interface and with jumps in material coefficients are included into the study. Two new approaches are studied along three standard choices. No approach is shown to be universally superior to others, and the resulting recommendation is that an actual method should be chosen based on properties of the problem.

1. Introduction

In many domain decomposition methods, an important role is played by the operator of averaging of a discontinuous function at the interface between adjacent subdomains. Two standard approaches commonly used in literature are: (i) arithmetic average, based simply on counting number of subdomains at an interface unknown, and (ii) weighted average, with weights derived from diagonal stiffness of subdomain Schur complements with respect to the interface. Its simplification presents approximation of the diagonal of the Schur complement by the diagonal of the original matrix, also known as the *stiffness scaling* [3]. The applicability of the so called *ρ-scaling* (see e.g. [3] or [4] for theoretical analysis) is limited to the case of material coefficients constant on each subdomain, which is not preserved in our examples. It also relies on knowledge of coefficients often not available in the solver. In the case of homogeneous material, it simplifies to arithmetic average. Consequently, it is not analyzed separately in this study.

In this paper, we study performance of these standard choices on a series of two-dimensional numerical experiments with the Poisson equation. These were selected to test the performance on regular and irregular subdomains, and in presence of jumps in material coefficients with different alignment with respect to interface. The

Balancing Domain Decomposition by Constrains (BDDC) method [2] is used for this study. In addition to the standard approaches, two new methods are included – averaging based on a unit jump on the interface described in [1], and a new approach based on a unit load applied on boundary of a subdomain. These approaches are shown to be competitive or even preferable in certain situations.

2. Reduction of the problem to the interface

Consider a boundary value problem with a self-adjoint operator defined on domain $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$. If we discretize the problem by means of the standard finite element method (FEM), we arrive at the solution of a system of linear equations in the matrix form

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is a large, sparse, symmetric positive definite (SPD) matrix and $\mathbf{f}$ is a vector of the right-hand side.

Let us decompose domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \ldots, N$. Unknowns common to at least two subdomains are called *interface unknowns* and the union of all interface unknowns form the *interface*. Remaining unknowns belong to subdomain *interiors*.

The first step used in many domain decomposition methods including BDDC is the reduction of the problem to the interface. Without loss of generality, suppose that unknowns are ordered so that interior unknowns form the first part and the interface unknowns form the second part of the solution vector, i.e. $\mathbf{u} = \begin{bmatrix} \mathbf{u}_\mathrm{o} & \widehat{\mathbf{u}} \end{bmatrix}^T$, where $\mathbf{u}_\mathrm{o}$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at the interface. System (1) can now be formally rewritten to the block form

$$\begin{bmatrix} \mathbf{K}_\mathrm{oo} & \mathbf{K}_\mathrm{or} \\ \mathbf{K}_\mathrm{ro} & \mathbf{K}_\mathrm{rr} \end{bmatrix} \begin{bmatrix} \mathbf{u}_\mathrm{o} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_\mathrm{o} \\ \widehat{\mathbf{f}} \end{bmatrix}. \tag{2}$$

The hat symbol ($\widehat{\ }$) is used to denote global interface quantities. If we suppose the interior unknowns are ordered subdomain after subdomain, then the submatrix $\mathbf{K}_\mathrm{oo}$ is block diagonal with each diagonal block corresponding to one subdomain.

After eliminating all the interior unknowns from (2), we arrive at the *Schur complement problem* for the interface unknowns

$$\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}, \tag{3}$$

where $\widehat{\mathbf{S}} = \mathbf{K}_\mathrm{rr} - \mathbf{K}_\mathrm{ro}\mathbf{K}_\mathrm{oo}^{-1}\mathbf{K}_\mathrm{or}$ is the *Schur complement* of (2) with respect to interface and $\widehat{\mathbf{g}} = \widehat{\mathbf{f}} - \mathbf{K}_\mathrm{ro}\mathbf{K}_\mathrm{oo}^{-1}\mathbf{f}_\mathrm{o}$ is sometimes called *condensed right-hand side*. Interior unknowns $\mathbf{u}_\mathrm{o}$ are determined by interface unknowns $\widehat{\mathbf{u}}$ via the system of equations $\mathbf{K}_\mathrm{oo}\mathbf{u}_\mathrm{o} = \mathbf{f}_\mathrm{o} - \mathbf{K}_\mathrm{or}\widehat{\mathbf{u}}$, which represents N independent subdomain problems with Dirichlet boundary condition prescribed on the interface and can be solved in parallel. The main objective represents the solution of problem (3), which is solved by the preconditioned conjugate gradient method (PCG).

3. Primal DD methods and BDDC

Primal DD methods can be viewed as preconditioners for problem (3), when it is solved by the PCG method. In every iteration of the PCG method, a preconditioned residual $\mathbf{M}\widehat{\mathbf{r}}$ is computed, where $\widehat{\mathbf{r}}$ is the residual. The action of $\mathbf{M}$ is realized by one step of the DD method.

The main idea of the primal DD substructuring methods of Neumann-Neumann type can be expressed as splitting the given residual of the PCG method to subdomains, solving subdomain problems and projecting the result back to the global domain. The primal preconditioner can be written as

$$M = ES^{-1}E^T\,, \tag{4}$$

where operator E^T represents splitting of the residual to subdomains, S^{-1} stands for solution of subdomain problems, and E represents projection of subdomain solutions back to the global problem by some averaging [5]. In the case some subdomains are 'floating', i.e. do not touch a part of boundary with Dirichlet boundary conditions, S is only positive semidefinite, and a generalized inverse S^+ may be needed in (4). The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq ||RE||_S^2\,, \tag{5}$$

where operator R splits the global interface into subdomains and the energetic norm on the right-hand side is defined by the scalar product as $||u||_S^2 = \langle Su, u\rangle$. The relationship (5) was proved in [5] assuming that $ER = I$, which means that if the problem is split into subdomains and then projected back to the whole domain, the original problem is obtained.

If we used independent subdomain problems only (no continuity conditions across the interface), the operator S would be expressed by a block diagonal matrix $\mathbf{S}$ with diagonal blocks representing local Schur complements on subdomains. Relationship between global and local problems can be expressed in matrix form as $\widehat{\mathbf{S}} = \mathbf{R}^{\mathrm{T}}\mathbf{S}\mathbf{R}$.

The main idea of the BDDC method ([2]) is to introduce a global *coarse problem* in order to achieve better preconditioning and to fix 'floating subdomains' by making their local Schur complements invertible. The matrix $\mathbf{S}$ is then positive definite, but it is not block diagonal any more, R now represents splitting of the global interface into subdomains (outside of the coarse unknowns), and E^T distributes residual among neighbouring subdomains only in those interface unknowns which are not coarse. Thus in BDDC, only part of the global residual is split into subdomains; residual at the coarse unknowns is left undivided – it is processed by the global coarse problem.

4. Choice of the averaging operator E

Three standard choices of the averaging operator E recommended already in [2] are (i) the arithmetic average, or weighted average with weights at interface nodes given (ii) by the ratio of the corresponding diagonal entries of the local and global

Schur complement, or (iii) by the ratio of the corresponding diagonal entries of the local and global system matrix $\mathbf{K}$. These choices are denoted here as *aa* (*a*rithmetic *a*verage), *ds* (*d*iagonal of *S*chur complement) and *dk* (*d*iagonal of $\mathbf{K}$), respectively. Method *dk* can be regarded as an approximation of method *ds*, if Schur complements are not computed explicitly.

We try to improve convergence of the BDDC method by choosing some more efficient weights. One of the proposed methods is to choose operator E so that it approximately minimizes the energy norm of the projection RE from estimate (5) for some suitable test vectors representing jumps across the interface. The method, described in more detail in [1], is denoted here as *uj* (*u*nit *j*umps). Here we numerically test just one choice of the test vectors: for every common face of two subdomains, one (local) test vector consisting of ones in the nodes belonging to the face and zeros elsewhere was chosen, corresponding to unit jump. Such choice results in the same weight for every node at the whole face. This, in a sense, makes this method similar to arithmetic average, where also just one weight is used for every node at the face (equal to 0.5).

The second proposed method, denoted as *ul* (*u*nit *l*oads), tries to exploit information of different values of local solution at corresponding interface nodes caused by constant (unit) load at the local interface.

Computation of the weights at interface nodes

For the sake of clarity, formulas are presented for the 2D case, where an interface node is either coarse (so there is no division into subdomains), or it belongs to a face (i.e. to exactly two adjacent subdomains). We also assume one degree of freedom per node, so that numbering of nodes and degrees of freedom coincide. It is straightforward to generalize these methods for 3D cases and more degrees of freedom at a node.

Notation for interface nodes:
j – number of the node in numbering with regard to interface
i – global number of the j-th node on interface
w_j^m – weight at the j-th node at the interface corresponding to the m-th subdomain

Formulas for individual methods:

$$aa: \qquad w_j^m = \frac{1}{2}$$

$$ds: \qquad w_j^m = \frac{s_{pp}^m}{s_{jj}}$$

$$dk: \qquad w_j^m = \frac{k_{qq}^m}{k_{ii}}$$

$$uj: \qquad w_j^m = \frac{\mathbf{d}^T \mathbf{S}^m \mathbf{d}}{\mathbf{d}^T \widehat{\mathbf{S}} \mathbf{d}}$$

$$ul: \qquad w_j^m = \frac{\mathbf{v}^m(j)}{\mathbf{v}^m(j)+\mathbf{v}^n(j)}$$

where:

s_{jj} – diagonal entry of the global Schur complement $\widehat{\mathbf{S}}$

s^m_{pp} – corresponding diagonal entry of the local Schur complement for the m-th subdomain; p is a local number (at the interface of the m-th subdomain) of the j-th node at the (global) interface

k_{ii} – diagonal entry of the (global) system matrix $\mathbf{K}$

k^m_{qq} – corresponding diagonal entry of the local matrix for the m-th subdomain; q is a local number (at the m-th subdomain) of the i-th node (in global numbering)

$\mathbf{d}$ – test vector equal to ones at the face which the j-th node belongs to and zeros otherwise (representing jump at that face)

$\mathbf{S}^m$ – local Schur complement for the m-th subdomain

$\mathbf{v}^m$, $\mathbf{v}^n$ – vectors of solution of the local (subdomain) Schur complement problems with zero values at coarse nodes and the right-hand side equal to one at every interface node that is not coarse, at the m-th and n-th subdomain respectively, where the n-th and m-th subdomain have common face which the j-th node belongs to.

5. Numerical results

The 2D problem of stationary heat conduction (Poisson equation) on a rectangular domain was used for testing. It was discretized by 59 x 59 bilinear finite elements of the same size and shape.

We compared two different divisions into subdomains: rectangular subdomains (Figure 1 left), as the usual choice for rectangular domain, and irregular subdomains (Figure 1 right), typical for domains with irregular shape or when some tool for automatic division into subdomains is used. For the coarse space, just the cross-points were used. Both homogeneous and nonhomogeneous materials were tested. The nonhomogenity was given by a 1:100 jump in conductivity. Nine different space arrangement of the jump was used, denoted as problems *p1–p9* and depicted in Figure 2 (white color represents the conductivity of 1 and black color represents the conductivity of 100).

Five different methods of weights for averaging between the subdomains in the BDDC method were compared, three standard ones (*aa*, *ds* and *dk*) and two new (*uj*, *ul*), all described in Section 4.

Number of PCG iterations for different methods are summarised for rectangular subdomains in Table 1 and for irregular ones in Table 2. The problem *p0* represents problem with constant conductivity on the whole domain, the problems *p1–p9* are problems with different locations of jumps in conductivity depicted in Figure 2. As a convergence criterion, norm of the residual less than 10^{-6} was used.

Condition numbers of the preconditioned systems are presented in Tables 3 and 4, where the row k0 is added with the condition number of Schur complement system without preconditioning. Condition numbers were estimated using ratio of the largest and the smallest eigenvalue computed by Matlab function `eig`.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
aa	14	45	14	48	22	22	43	42	46	42
uj	14	6	14	60	21	23	49	37	49	29
ds	14	6	14	28	23	22	30	26	59	16
dk	14	6	14	28	22	22	31	25	59	16
ul	14	6	15	39	23	23	38	35	60	16

Table 1: Number of iterations of PCG, rectangular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
aa	13	51	46	65	35	52	58	54	68	83
uj	14	42	41	77	49	72	55	43	70	14
ds	19	23	28	37	37	55	30	33	50	16
dk	20	23	29	37	37	57	32	34	56	16
ul	15	21	24	54	46	64	47	34	64	15

Table 2: Number of iterations of PCG, irregular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
k0	5e2	1e3	3e3	2e3	4e4	2e4	2e4	3e4	4e3	3e3
aa	3.71	255	3.65	83	20	33	59	61	69	83
uj	3.72	1.15	3.22	73	18	30	80	50	39	19
ds	3.71	1.15	3.61	104	20	33	50	51	153	7
dk	3.71	1.15	3.65	105	20	33	53	55	160	7
ul	3.94	1.15	3.83	46	21	33	55	58	40	8

Table 3: Condition number of the preconditioned system, rectangular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
k0	6e2	4e3	3e3	2e3	5e4	3e4	3e4	4e4	6e3	4e3
aa	3.26	73	62	72	50	57	82	82	77	136
uj	3.31	91	49	166	147	157	90	131	116	4
ds	8.23	22	148	79	106	172	128	123	94	6
dk	8.79	22	161	80	114	188	141	132	113	7
ul	3.49	19	34	71	109	98	99	135	82	4

Table 4: Condition number of the preconditioned system, irregular subdomains.

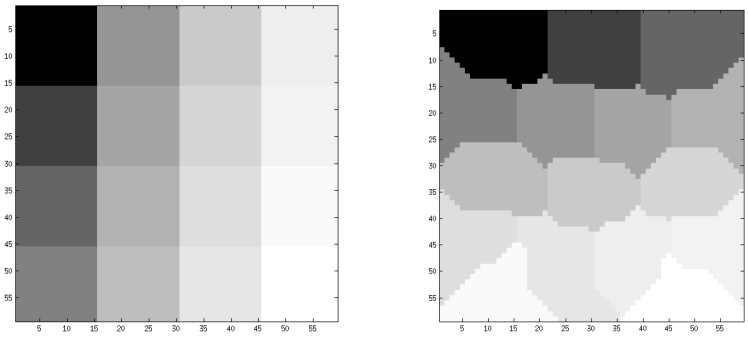

Figure 1: Division into rectangular (left) and irregular (right) subdomains.

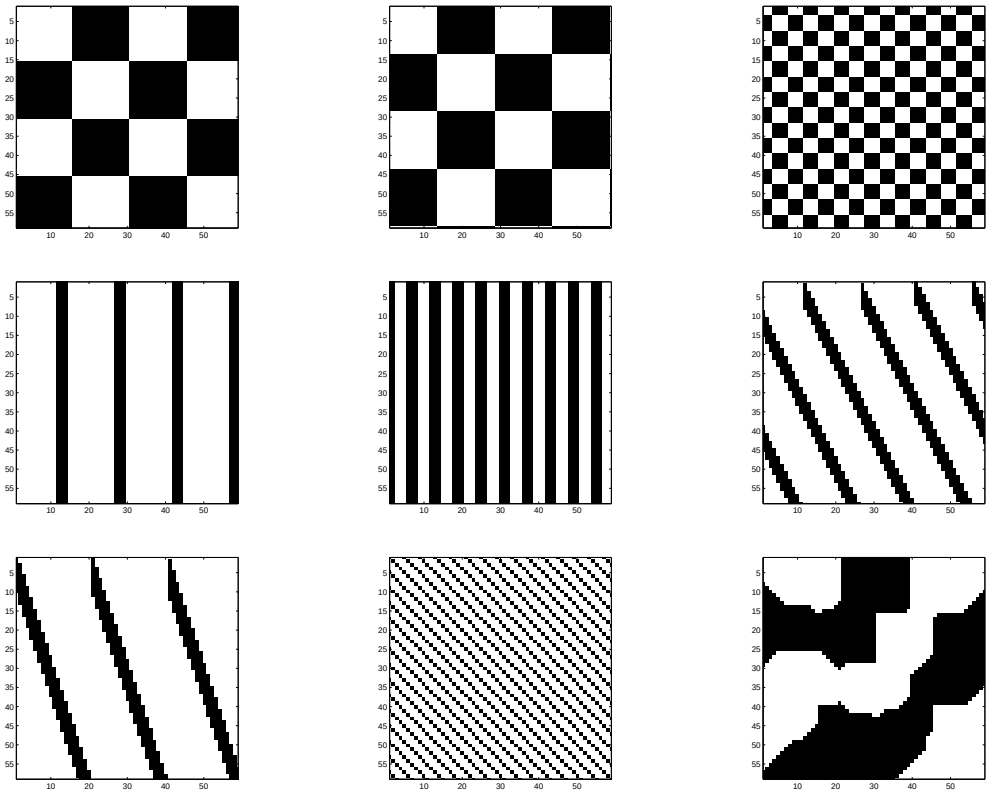

Figure 2: Different nonhomogeneous material properties for problems *p1–p9* (the first row *p1*, *p2*, *p3*, the second row *p4*, *p5*, *p6*, the last row *p7*, *p8*, *p9*).

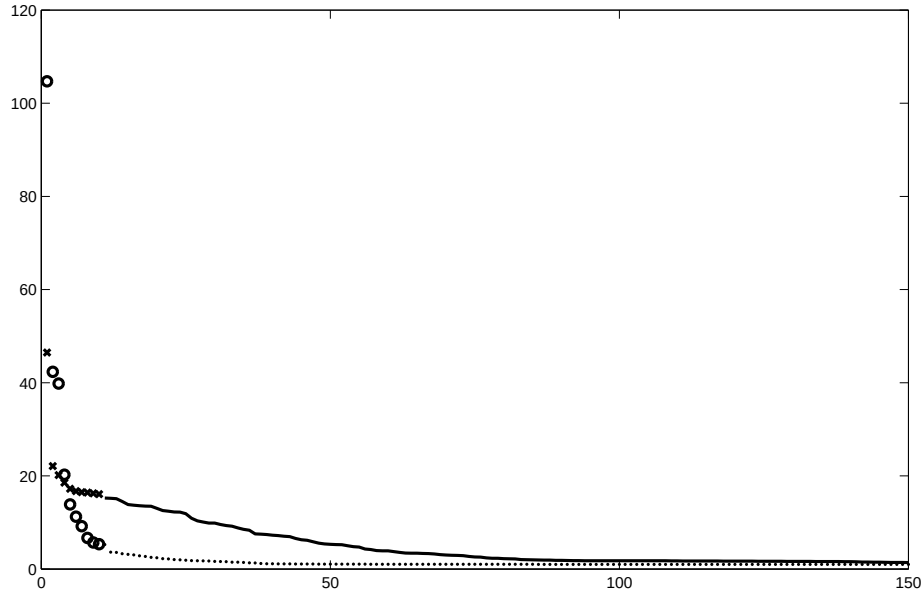

Figure 3: Comparison of the first 150 eigenvalues of $M\widehat{S}$ for methods *dk* ('$\circ$', dotted line), and *ul* ('$\times$', solid line), problem *p3*, regular subdomains.

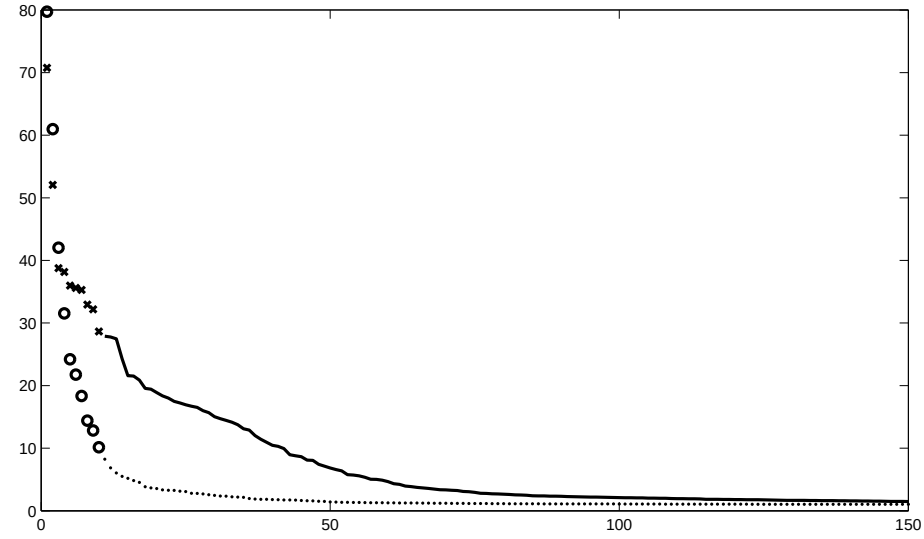

Figure 4: Comparison of the first 150 eigenvalues of $M\widehat{S}$ for methods *dk* ('$\circ$', dotted line), and *ul* ('$\times$', solid line), problem *p3*, irregular subdomains.

6. Conclusions

Our numerical results lead to several observations:

- Arithmetic average (method *aa*) is surprisingly robust even if jumps in coefficients of the equation occur, as long as the jumps do not exactly coincide with the interface (for instance see problem *p2*, where the jumps are shifted only one row of elements from the interface).

- Weights computed as the ratio of the corresponding diagonal entries of local and global Schur complements can be very successfully approximated using the original system matrix K instead of the Schur complements.

- For irregular shape of interface without jumps in coefficients (problem *p0*), using either *ds* or *dk* method instead of arithmetic averages (*aa*) can lead to worse convergence.

- Method *ul* seems to give promising results: it is usually better than arithmetic average, often it is comparable or better than *ds* or *dk*, and it does not seem to have difficulties with irregular shape of interface. However in some cases it leads to worse convergence than all of the standard methods.

- Both proposed methods, *uj* and *ul*, lead very often to lower condition number of the preconditioned system than all standard methods, *aa*, *ds* and *dk*. However, they often give worse convergence results. The reason for this seems to be the distribution of eigenvalues, as illustrated for problem *p3* with rectangular and irregular subdomains in Figures 3 and 4, respectively. For both cases, the first 150 eigenvalues for methods *dk* (circles) and *ul* (crosslines) are compared. For the first few largest eigenvalues, the values for the *dk* method are larger than the values for the *ul* method, which leads to larger condition number (the smallest eigenvalue is allways equal to one). However, following values for the *dk* method quickly drop down and cluster around 1, and they are much lower than the values for the *ul* method. As is well known, clustering of eigenvalues is another important aspect influencing the rate of convergence of PCG.

For equation without jumps in coefficients, the method of choice seems to be the arithmetic averaging. It can lead to very good convergence even if there are jumps in coefficients, except the case where jumps exactly coincide with the interface or some part of it.

If there are jumps in coefficients, the best choice is usually choosing weights as the ratio of corresponding diagonal entries of local and global Schur complements (method *ds*). As these numbers typically are not in hand, a very good substitute is using diagonal entries of local and global original system matrices (method *dk*).

Interesting results are obtained by the method *ul*, which deserves further investigation. Method *uj* does not lead to better convergence than the standard methods.

Acknowledgements

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by the Czech Science Foundation under project 106/08/0403, and by the Academy of Sciences of the Czech Republic through RVO: 67985840.

References

[1] Čertíková, M., Burda, P., Novotný, J., and Šístek, J.: Some remarks on averaging in the BDDC method. In: T. Vejchodský, J. Chleboun, P. Přikryl, K. Segeth, J. Šístek (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 15*, pp. 28–34. IM ASCR, Prague, 2010.

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

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

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

[5] Mandel, J. and Sousedík, B.: BDDC and FETI-DP under minimalist assumptions. Computing **81** (2007), 269–280.