Marta Čertíková; Jakub Šístek; Pavel Burda

Different approaches to interface weights in the BDDC method in 3D

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 8-13, 2014. Institute of Mathematics AS CR, Prague, 2015. pp. 47–57.

Persistent URL: http://dml.cz/dmlcz/702662

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

DIFFERENT APPROACHES TO INTERFACE WEIGHTS IN THE BDDC METHOD IN 3D

Marta Čertíková[1], Jakub Šístek[2], 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

In this paper, we discuss the choice of weights in averaging of local (subdomain) solutions on the interface for the BDDC method (Balancing Domain Decomposition by Constraints). We try to find relations among different choices of the interface weights and compare them numerically on model problems of the Poisson equation and linear elasticity in 3D. Problems with jumps in coefficients of material properties are considered and both regular and irregular interfaces between subdomains are tested.

1. Introduction

An important ingredient of many domain decomposition methods is a technique used for determining a continuous approximation of solution at interface from discontinuous local solutions from adjacent subdomains. A standard approach described already in [3] is to compute global value of any given interface unknown as some weighted average of local (subdomain) values of the corresponding unknown only. Very often just arithmetic average is used, based simply on counting number of subdomains to which the interface unknown belongs. More sophisticated method is to derive the weights from diagonal stiffness of subdomain Schur complements with respect to the interface. As these complements are typically not computed explicitly in efficient implementations, the diagonal of the Schur complement is sometimes approximated by the diagonal of the original matrix (also known as the *diagonal stiffness scaling*). Another method is the so called *ρ-scaling* (see e.g. [5] for theoretical analysis). However, it is limited to the case of material coefficients constant on each subdomain, which is often too restrictive requirement for applications, and it is also not preserved in our examples. Nevertheless, we tried its modification, using material coefficients on individual elements instead. In any case, this approach requires access to material coefficients from the equations solver in an implementation.

A different recent method of evaluation of global values, called *deluxe scaling*, represents solving local problems containing two or more adjacent subdomains (see e.g. [4]). Implementation of this method is quite demanding, and it is not covered by our numerical experiments, although we involve it in our theoretical considerations.

Our method of *averaged unit jump* was originally derived by approximate minimization of the upper bound on the condition number of the preconditioned operator (see e.g. [1]).

In this paper, we analyze theoretically relationships among methods mentioned above. We use an abstract formulation of the BDDC preconditioner presented in [6] in order to obtain a clearer form of our results formulated in Lemma 1 that seems to be new. We also test the methods numerically on 3D Poisson and linear elasticity problems, together with two heuristic methods (called *unit jump* and *unit load*; *unit load* was proposed and tested on 2D Poisson problem in [2]).

2. Notation

Consider a system of linear equations $\mathbf{Ku} = \mathbf{f}$ obtained by discretization of boundary value problem with a self-adjoint operator defined on a domain Ω. Let us decompose the 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*. The first step is the reduction of the problem to the interface. We thus arrive at the *Schur complement problem* for the interface unknowns $\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}$, where $\widehat{\mathbf{S}}$ is a symmetric positive definite (SPD) matrix. This system is solved by the preconditioned conjugate gradient method (PCG). More detailed description of this reduction to the interface can be found in [2].

According to [6], let us interpret the matrix $\widehat{\mathbf{S}}$ as an operator $\widehat{S} : \widehat{W} \to \widehat{W}'$, where $\widehat{W}$ is a finite dimensional linear space. Let us introduce another space $\widetilde{W}$ such that $\widehat{W} \subset \widetilde{W}$ (in terms of subdomains, the space $\widetilde{W}$ represents functions which can be discontinuous across parts of the interface, it means they can have different values of interface unknowns on individual subdomains – there can be a jump across the interface). Let $\widetilde{S}$ be an extension of $\widehat{S}$ to $\widetilde{W}$. Denote R the natural injection from $\widehat{W}$ to $\widetilde{W}$, then we have $\widehat{S} = R^T \widetilde{S} R$. The space $\widetilde{W}$ has to be chosen so that the extended operator $\widetilde{S}$ is positive definite and its inversion can be applied efficiently. Then the BDDC preconditioner can be expressed as

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

where operator E^T represents splitting of the residual to subdomains, $\widetilde{S}^{-1}$ stands for solution on subdomains and the coarse level, and E represents averaging of subdomain solutions back to the global problem. The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq ||RE||^2_{\widetilde{S}}\,, \tag{2}$$

where the energetic norm on the right-hand side is defined by the scalar product as $||u||^2_{\widetilde{S}} = \langle \widetilde{S}u, u\rangle$. The relationship (2) was proved in [6] 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.

3. Theoretical background of the averaging methods

In every step of PCG, for the given residual $r \in \widehat{W}'$, an approximation Mr of its preimage $e = \widehat{S}^{-1}r$ is to be computed. Our goal is to construct the averaging operator E so that good convergence of the PCG method is achieved, and its action is not too expensive. There is an upper bound on the distance of e from Mr:

Lemma 1. *Assume $ER = I$ and use the notation from Section 2. Denote $w = \widetilde{S}^{-1}E^{\mathrm{T}}r$. Then the following estimation holds:*

$$||e - Mr\ ||_{\widehat{S}} \leq ||R\,e - w\,||_{\widetilde{S}} + ||(I - RE)w\,||_{\widetilde{S}}\ , \tag{3}$$

where $||.||_{\widehat{S}}$ and $||.||_{\widetilde{S}}$ are the energetic norms in $\widehat{W}$ and $\widetilde{W}$, respectively.
Square of the first term on the right-hand side can be expressed as

$$||R\,e - w\ ||^2_{\widetilde{S}} = \langle Mr, r\rangle - ||e||^2_{\widehat{S}}\ . \tag{4}$$

Proof. The inequality (3) is obtained from triangle inequality and the fact, that the energetic norms in $\widehat{W}$ and $\widetilde{W}$ are equal:

$$||e - Mr\ ||_{\widehat{S}} = ||e - Ew\ ||_{\widehat{S}} = ||R\,e - REw\ ||_{\widetilde{S}} \leq ||R\,e - w\ ||_{\widetilde{S}} + ||w - REw\ ||_{\widetilde{S}}\ .$$

Rewriting square of the first term on the right-hand side of the inequality (3):

$$\begin{aligned}
||Re - w||^2_{\widetilde{S}} &= \langle Re - w, \widetilde{S}(Re - w)\rangle = \langle Re - \widetilde{S}^{-1}E^{\mathrm{T}}r, \widetilde{S}Re - \widetilde{S}\widetilde{S}^{-1}E^{\mathrm{T}}r\rangle \\
&= \langle Re, \widetilde{S}Re\rangle - \langle Re, E^{\mathrm{T}}r\rangle - \langle \widetilde{S}^{-1}E^{\mathrm{T}}r, \widetilde{S}Re\rangle + \langle \widetilde{S}^{-1}E^{\mathrm{T}}r, E^{\mathrm{T}}r\rangle \\
&= \langle e, R^{\mathrm{T}}\widetilde{S}Re\rangle - \langle ERe, r\rangle - \langle r, ERe\rangle + \langle E\widetilde{S}^{-1}E^{\mathrm{T}}r, r\rangle \\
&= \langle e, \widehat{S}e\rangle - 2\langle e, r\rangle + \langle Mr, r\rangle = \langle Mr, r\rangle - \langle e, r\rangle = \langle Mr, r\rangle - ||e||^2_{\widehat{S}}\ .
\end{aligned}$$

□

It seems that nearly all methods used so far for averaging have some connection with minimizing the second term $||(I - RE)w\,||_{\widetilde{S}}$ on the right-hand side of (3), as we show bellow. Moreover, there is also a connection with the upper bound (2), because norms of the complementary projections are equal: $||RE||_{\widetilde{S}} = ||I - RE||_{\widetilde{S}}$.

Our approach in [1] to find proper weights for averaging (in other words, to find the so called weight matrix) is to minimize the term $||(I - RE)u\,||_{\widetilde{S}}$ for some given $u \in \widetilde{W}$ under the standard assumption that the global value of any given interface unknown is computed as weighted average of subdomain values of the corresponding unknown only – it means that the weight matrix is supposed to be diagonal. For two

adjacent subdomains (with no coarse space) it leads to a system of linear equations for unknown diagonal of the weight matrix $\mathbf{A}$:

$$\mathbf{Ad} = \widehat{\mathbf{S}}^{-1}\mathbf{S_1 d} = (\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1 d}, \tag{5}$$

where $\mathbf{S_i}$ is the local Schur complement of the i-th subdomain, and the vector $\mathbf{d}$ represents a jump across the interface in some given test vector $\mathbf{u}$. For more details see [1]. This relationship can be interpreted as: for a given jump $\mathbf{d}$, find a diagonal representation $\mathbf{A}$ (dependent on $\mathbf{d}$) of some general matrix $(\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1}$ that is independent of $\mathbf{d}$. And so maybe we can use this general matrix for a construction of the averaging operator E independent of $\mathbf{d}$ and use the (full) matrix $(\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1}$ instead of the diagonal matrix $\mathbf{A}$. By this approach we arrive to *deluxe* scaling proposed in [4]. In this method, the system $(\mathbf{S_1} + \mathbf{S_2})\mathbf{v} = \mathbf{S_1 r}$ is solved for every pair (or, in some cases, group) of adjacent subdomains in every step of the PCG method, where $\mathbf{r}$ is a local residual on the appropriate part of the interface.

As we would like to avoid solving that large number of local systems, we look for some simplification of system (5). One option is to omit all off-diagonal entries of matrices $\mathbf{S_i}$, which leads to the choice of weights as ratios of diagonals of local and global Schur complements. Another option is to assume that all weights on the local interface are equal (as in the case of arithmetic average) and choose some suitable test jump $\mathbf{d}$. We have chosen a unit jump for numerical tests and call the method *averaged unit jump* method.

Note: The expression (4) has led us to the idea of trying to minimize the term $\langle Mr, r\rangle$. For the case of two adjacent subdomains, using the same assumption of diagonal weight matrix and using the same process of minimization as in [1], we arrive at equations

$$\mathbf{Ar} = (\mathbf{S}_1^{-1} + \mathbf{S}_2^{-1})^{-1}\mathbf{S}_2^{-1}\mathbf{r}, \tag{6}$$

where the vector $\mathbf{r}$ represents a local residual on the appropriate part of the interface. Nevertheless, this result does not seem to bring any practical advantage compared to system (5). Again, omitting all off-diagonal entries of matrices $\mathbf{S_i}$ leads to the well-known choice of weights as ratios of diagonals of local and global Schur complements.

4. Approaches for computation of weights at interface nodes

For the sake of clarity, formulas are presented again for 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 generalise these methods to more than two adjacent subdomains (on edges) or more degrees of freedom at a node. Notation:

j ... number of a node in numbering with regard to the interface

i ... global number of the j-th interface node with regard to the global numbering

w_j^1 ... weight at the j-th node at the interface corresponding to the first subdomain (the weight w_j^2 for the second subdomain is then $w_j^2 = 1 - w_j^1$)

• *aa* ... arithmetic average: $w_j^1 = \frac{1}{2}$

• *dk* ... fractions of diagonal entries of the system matrix $\mathbf{K}$: $w_j^1 = \frac{k_{qq}^1}{k_{ii}}$, where k_{ii} is a diagonal entry of the (global) system matrix $\mathbf{K}$, k_{qq}^1 is the corresponding diagonal entry of the local matrix for the first subdomain; q is a local number (at the first subdomain) of the i-th node (in global numbering)

• *rho* ... element-wise ρ-scaling: $w_j^1 = \frac{\alpha_j^1}{\alpha_j^1+\alpha_j^2}$, where α_j^k is a local material coefficient computed as an arithmetic average of material coefficients given on elements containing the j-th node and belonging to k-th subdomain

• *auj* ... averaged unit jump method: $w_j^1 = \frac{\mathbf{d}^T\mathbf{S}_1\mathbf{d}}{\mathbf{d}^T(\mathbf{S}_1+\mathbf{S}_2)\mathbf{d}}$, where $\mathbf{d}$ stands for test vector equal to ones at the common face of the two subdomains and zeros otherwise (representing jump at that face), and $\mathbf{S}_k$ is the local Schur complement for the k-th subdomain

• *uj* ... unit jump method: $w_j^1 = \frac{g_j^2}{g_j^1+g_j^2}$, where $\mathbf{g}^k = (g_1^k, g_2^k, \dots)^T$ is the local vector of reaction on the k-th subdomain induced by a unit jump: $\mathbf{g}^k = \mathbf{S}_k\,\mathbf{d}$

• *ul* ... unit load method: $w_j^1 = \frac{v_j^1}{v_j^1+v_j^2}$, where $\mathbf{v}^k = (v_1^k, v_2^k, \dots)^T$ is the vector of the local solution on the k-th subdomain under unit load, i.e. with the right-hand side equal to ones: $\mathbf{S}_k\,\mathbf{v}^k = \mathbf{d}$

5. Numerical results

Our aim is to numerically compare the robustness of the approaches to averaging listed in Section 4 with respect to two model aspects known to cause issues to domain decomposition methods, namely (i) roughness of the interface among subdomains, (ii) jumps in material coefficients inside the domain.

Following our preliminary 2D numerical tests of some of the methods for averaging (see [2]), we choose 3D **Poisson** and **linear elasticity** problems on a unit cube domain for testing (solutions of the problems are illustrated in Figure 3). The Poisson equation has unit right-hand side and homogeneous Dirichlet boundary conditions on the surface of the cube. For linear elasticity problem, the cube is mounted at a vertical face and loaded by its own weight.

Problems are discretized by the finite element method using trilinear cubic elements, all of them of the same size h. The domain was divided into 4×4×4 cubic subdomains of size H, and we test different numbers of elements per subdomain edge, H/h = 4, 8, 16, 32, and 60. The interface is either **regular**, i.e. consisting of plane sections only, or **jagged** (see Figure 1). Both homogeneous and nonhomogeneous materials are considered. For the Poisson problem, the low material constant is chosen as 1 and the high one as 10^6, for elasticity the low value of Young modulus is 10^5 and the high one is $2.1{\cdot}10^{11}$. Three nonhomogeneous material arrangements are designed (see Figure 2):

- Material 1 – *Random elements*: For each element, the value of the material

coefficient is chosen randomly with a uniform distribution between the low and high values.

- Material 2 – *Slices* along the interface: Only the low and high values of the material coefficient are used as depicted in Figure 2. The solution to the Poisson problem defined on this domain is in Figure 3.

- Material 3 – *Stiff rods* of material with the high coefficient arranged in a 4×4 lattice inside the material with the low coefficient.

These arrangements have been chosen to model several situations encountered in engineering, such as rapidly oscillating coefficients, layered structures, or reinforced composite structures.

Coarse nodes at all crosspoints and coarse averages are used. Quality of the preconditioner is measured by the number of iterations of PCG needed to reduce the relative residual below 10^{-6}.

For computations, the *BDDCML* library – a massively parallel implementation of the Adaptive-Multilevel BDDC method – is used. The Schur complements are not computed explicitly in this implementation, so the averaging by values on diagonals of the complements is only approximated by diagonals of the subdomain matrices.

For **homogeneous material** and **regular** interface, the same results have been obtained by all methods of averaging, only the *ul* method has performed little worse. For Poisson problems the number of iterations are depicted on Figure 4 (left), for linear elasticity the results are very similar and they are not reported.

For **jagged** interface, the results are summarised in Table 1, and for Poisson problems, they are also plotted in Figure 4 (right). The behaviour of the methods is again very similar for both Poisson and linear elasticity problems, the main difference is worse convergence for elasticity. The interesting observation is that the rate of worsening of the convergence with growing ratio of H/h is different for different methods: for instance, the *auj* method is much more stable than *dk* method, and although *auj* is the worst method for 4 elements per subdomain edge, it belongs to the best for the 32 and 60 elements per edge.

	Poisson problem					linear elasticity			
H/h	4	8	16	32	60	4	8	16	32
aa	11	14	15	16	18	28	35	37	39
dk	6	8	11	17	24	13	18	28	44
rho	11	14	15	16	18	28	35	37	39
auj	12	14	15	16	18	32	40	41	42
uj	7	9	14	22	32	19	30	41	68
ul	10	13	17	22	27	21	37	51	61

Table 1: Number of iterations: homogeneous material, jagged interface

The results for **nonhomogeneous materials** are illustrated by graphs only. For Material 1 (random elements), the behavior is again very similar for both Poisson and linear elasticity problems, so only results for the Poisson problem are depicted in Figure 5 for regular (left) and jagged (right) interface. Note the different scale of the vertical axes. We can see that the jagged interface worsens dramatically the behaviour of both aa and auj methods, which do not adapt locally to the jumps along the interface and use a single weight for the whole part of the interface.

In the case of Material 2 (slices), for regular interface, all methods perform equally well with the exception of aa. For the Poisson problem, see Figure 6 (left). For jagged interface (Figure 8), methods aa and uj did not converge in 1000 and for others, there is a difference between Poisson problems (left) and linear elasticity (right): for the former, the dk method worsens quite rapidly with growing H/h, for the latter dk it is the best even for $H/h = 32$.

Material 3 (stiff rods) leads to quite challenging problems: the results are diverse and difficult to predict. Methods behave differently for Poisson problems and linear elasticity even for regular interface (see Figure 9). For jagged interface, convergence was achieved only for Poisson problems (Figure 7 right).

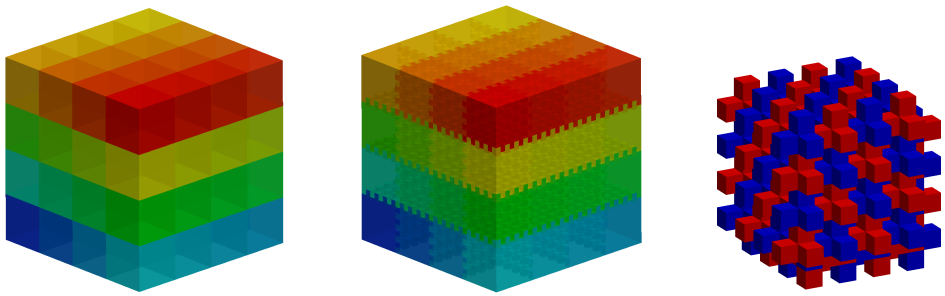

Figure 1: Regular (left) and jagged (centre) interface, and a detail of an interior jagged subdomain (right)

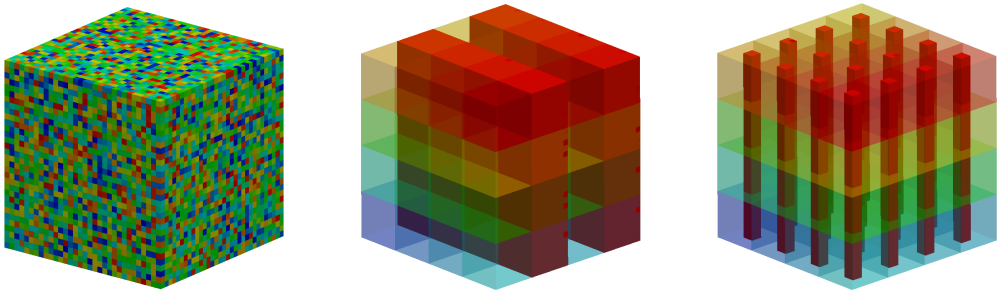

Figure 2: Material 1 – random elements (left), Material 2 – slices (centre), and Material 3 – stiff rods (right)

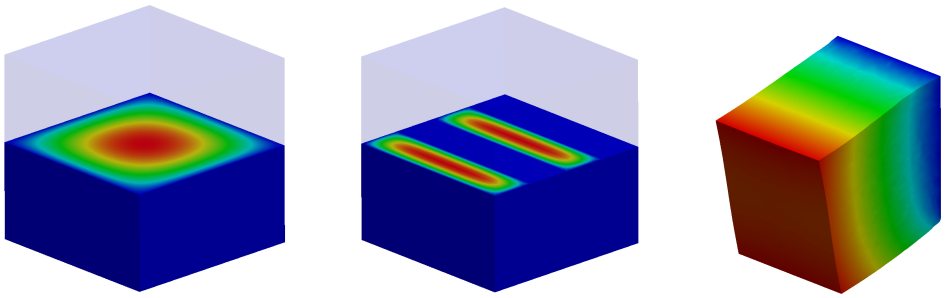

Figure 3: Solution to the Poisson problem with homogeneous material (left), for the Poisson problem with Material 2 – slices (centre), and magnified displacement of the linear elasticity problem with homogeneous material (right)

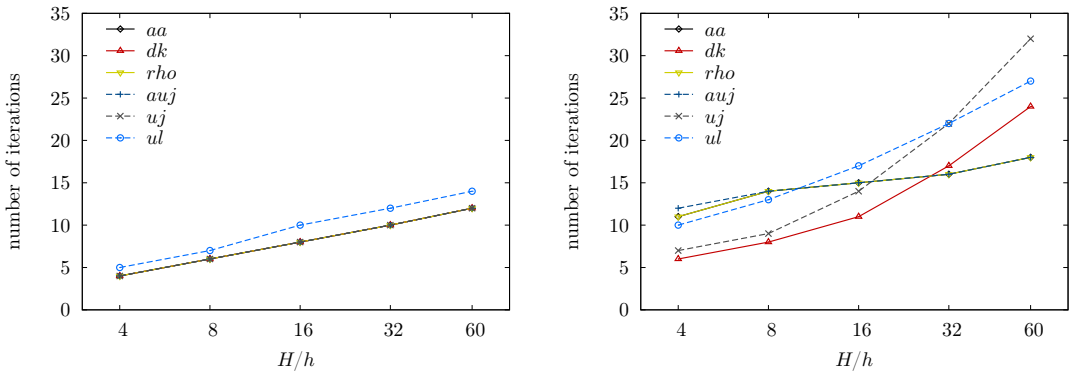

Figure 4: Homogeneous material, Poisson problem, regular (left) and jagged (right) interface

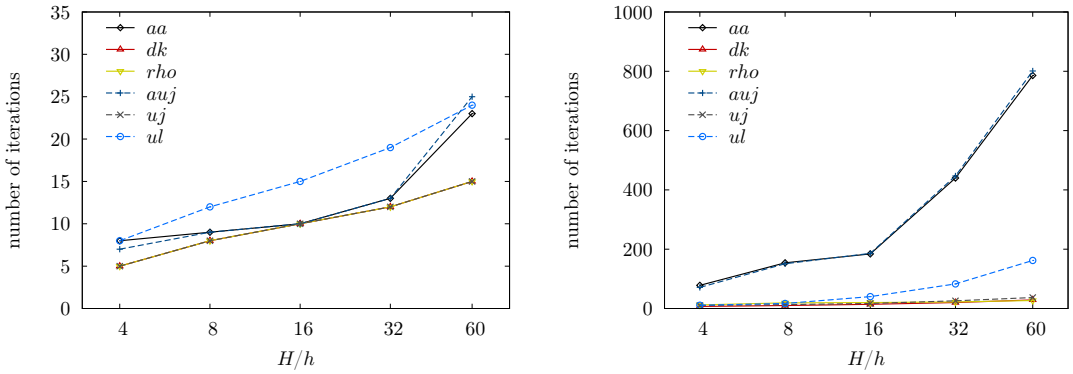

Figure 5: Material 1 (random elements), Poisson problem, regular (left) and jagged (right) interface. Note the different scale on the vertical axes

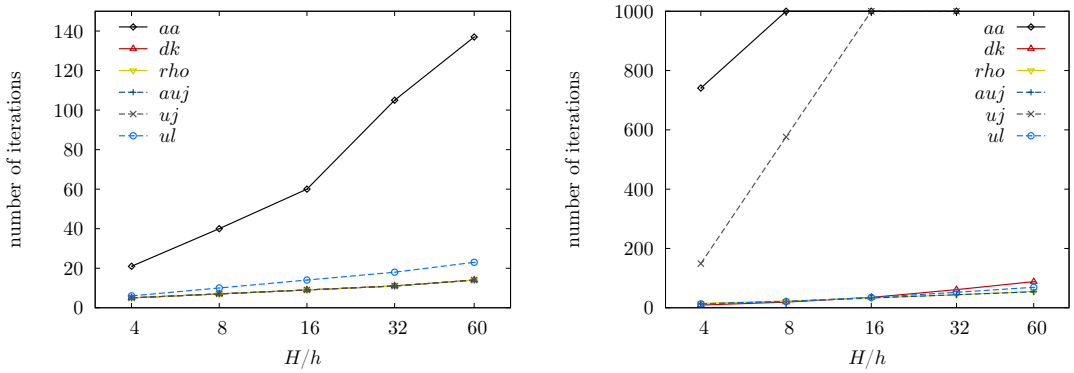

Figure 6: Material 2 (slices), Poisson problem, regular (left) and jagged (right) interface

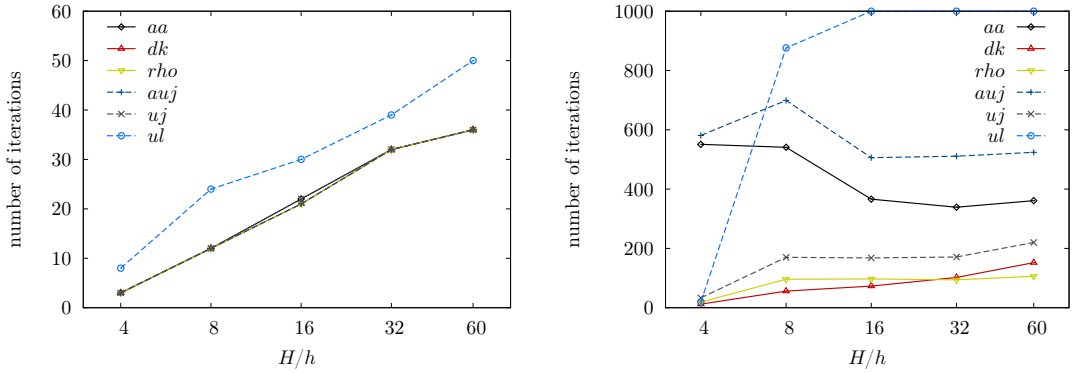

Figure 7: Material 3 (stiff rods), Poisson problem, regular (left) and jagged (right) interface. Note the different scale on the vertical axes.

6. Conclusions

Three new forms of the averaging operator (auj, uj, ul) have been numerically compared with three standard ones (aa, dk, rho) on several challenging test problems. We have found that the choice of the method of averaging has a significant influence not only on the convergence of the BDDC method, but also on the rate of worsening of the convergence with growing ratio of H/h. The main conclusion one can draw from our numerical results is that there is no single universal method for averaging that would perform well for all cases; the performance of the methods depends on the problem, on the H/h ratio as well as on the profile of the interface (regular or jagged). Moreover, it is usually not clear in advance, which method would be the best one for the given problem. It seems that a robust and efficient implementation of the BDDC method should offer a flexible choice from several different averaging methods, and it is worth trying several of them before a production computations are performed.

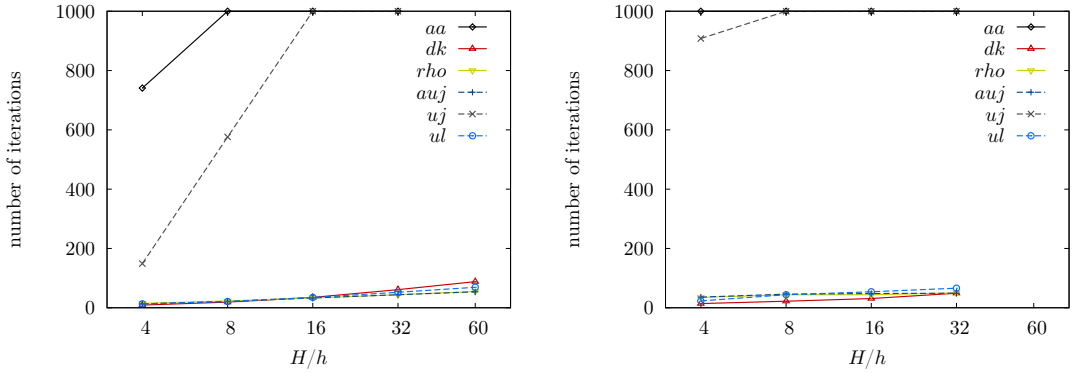

Figure 8: Material 2 (slices), jagged interface, Poisson (left) and linear elasticity (right) problems. Note the different scale on the vertical axes.

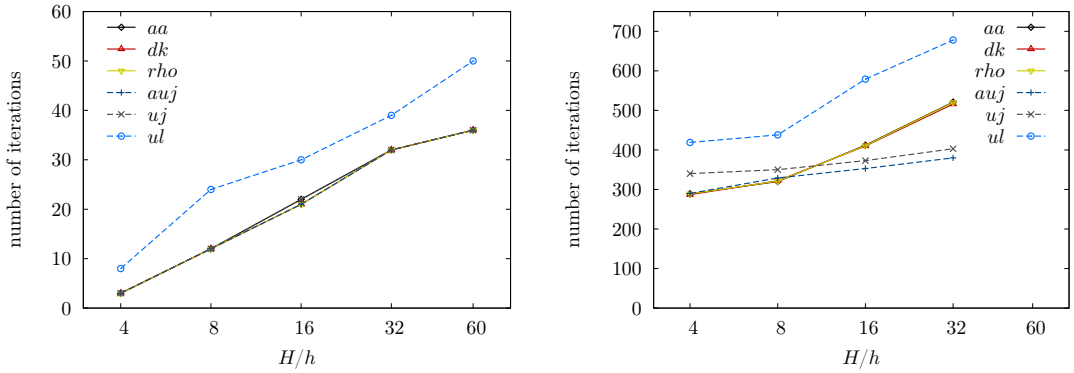

Figure 9: Material 3 (stiff rods), regular interface, Poisson (left) and linear elasticity (right) problems. Note the different scale on the vertical axes.

Nevertheless, some recommendations based on our results can still be made: For homogeneous problems, the simplest method *aa* is sufficient for both regular and irregular interface. It seems also less sensitive to growing H/h ratio than other methods. However *aa* should not be applied to nonhomogeneous materials for which the convergence can be disastrous. For nonhomogeneous problems, *dk* or, if the solver has access to material data, the *rho* scaling perform well. Moreover, *rho* seems more reliable, as convergence for *dk* deteriorates more rapidly with growing H/h ratio, especially for jagged interface. For several complicated cases combining jumps and irregular interface, the newly developed methods, *auj*, *uj*, and *ul* noticeably superseded the standard approaches, especially for linear elasticity problems.

In Lemma 1, some relationships between preconditioned residual and its preimage in both the spaces $\widehat{W}$ and $\widetilde{W}$ for the BDDC preconditioner have been presented. However, they have not led us to any new practical method for averaging so far.

Acknowledgements

This work was supported by the Czech Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by the Czech Science Foundation under project 14-02067S, and by the Academy of Sciences of the Czech Republic through RVO: 67985840. The work was also supported by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033). We are grateful to Jan Mandel for fruitful discussions on the topic.

References

[1] Čertíková, M., Burda, P., Novotný, J., and Šístek, J.: Some remarks on averaging in the BDDC method. In: T. Vejchodský et al. (Eds.), *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, 2010.

[2] Čertíková, M., Šístek, J., and Burda, P.: On selection of interface weights in domain decomposition methods. In: J. Chleboun et al. (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 16, Dolní Maxov, Czech Republic, June 3–8, 2012*, pp. 35–44. Institute of Mathematics AS CR, 2013.

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

[4] Dohrmann, C. R. and Widlund, O. B.: Some recent tools and a BDDC algorithm for 3D problems in H(curl). In: R. Bank et al. (Eds.), *Domain Decomposition Methods in Science and Engineering XX*, *Lecture Notes in Computational Science and Engineering*, vol. 91, pp. 15–25. Springer, 2013.

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

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