NUMERICAL COMPARISON OF DIFFERENT CHOICES OF INTERFACE WEIGHTS IN THE BDDC METHOD

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

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

Abstract

Balancing Domain Decomposition by Constraints (BDDC) belongs to the class of primal substructuring Domain Decomposition (DD) methods. DD methods are iterative methods successfully used in engineering to parallelize solution of large linear systems arising from discretization of second order elliptic problems.

Substructuring DD methods represent an important class of DD methods. Their main idea is to divide the underlying domain into nonoverlapping subdomains and solve many relatively small, local problems on subdomains instead of one large problem on the whole domain. In primal methods, it has to be specified how to distribute interface residuals among subdomains and how to obtain global, interface values of solution from local values on adjacent subdomains. Usually a weighted average is used with some simple choice of weights.

In our paper we present numerical comparison of three different choices of interface weights on test problem of 2D Poisson equation, with and without jumps in coefficients.

1. Introduction

The BDDC method introduced in [1] is one of the most popular substructuring DD methods. It belongs to the wide class of the primal Neumann-Neumann domain decomposition methods. As it has been recently shown in [4], a primal preconditioner of such type is determined by the choice of two operators: the injection R and the averaging E. These two operators appear also in the estimate of the condition number of the preconditioned operator (see (4) bellow).

The choice of the operator R can be formulated as the choice of continuity conditions across the interface (coarse unknowns). It is well-known that the choice of

the operator R has a strong influence on the quality of the preconditioning. A lot of work has been invested into research of relations between the choice of coarse unknowns and the quality of preconditioning, and significant results were obtained (e.g. in [2, 4]).

The averaging operator E seems to be aside from the main effort of the investigation so far. Standard choices of E found already in [1] are arithmetic average and average weighted by diagonal entries of matrices of local problems. Another recommended choice of weights is a ratio of corresponding diagonal entries of local and global Schur complements. However, often these values are not available, because Schur complements are not computed explicitly in the most efficient implementations of substructuring methods.

A more general framework for derivation of the averaging operator was introduced in [5]. The standard choice of using diagonal entries of Schur complements mentioned above can be regarded as one special case of it. In [5], two new variants derived from the general framework were preliminarily tested on a test problem with two subdomains, which led to promising results. However, in the case of two subdomains only, there is no difference between global and local interface, so these results cannot be regarded as typical. In this paper we use a test problem with four subdomains and one cross-point.

We start with brief introduction into primal substructuring methods and BDDC (detailed description with many references can be found in [3]) and description of different methods for averaging. Then numerical comparison of three averaging methods follows: one of the new variants tested in [5] and two standard choices – arithmetic average and average weighted by diagonal entries of Schur complements.

2. Reduction of the problem to the interface

Let us consider a boundary value problem with a self-adjoint operator defined on a domain $\Omega \subset \mathbf{R}^2$ or $\mathbf{R}^3$. 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

$$\mathbf{Ku} = \mathbf{f}, \tag{1}$$

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

Let us decompose the domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \ldots N$. Unknowns common to at least two subdomains form the *global interface*. Remaining unknowns are classified as belonging to subdomain *interiors*. The global interface can be expressed as union of *local interfaces*, containing interface unknowns involved just in subdomain Ω_i.

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. $\mathbf{u} = \left[\begin{array}{cc} \mathbf{u}_\mathrm{o} & \widehat{\mathbf{u}} \end{array} \right]^T$,

where $\mathbf{u}_o$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at interface. Now, system (1) can be formally rewritten to block form

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

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 $\mathbf{K}_{oo}$ is block diagonal with each diagonal block corresponding to one subdomain.

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

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

where $\widehat{\mathbf{S}} = \mathbf{K}_{rr} - \mathbf{K}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{K}_{or}$ is the *Schur complement* of (2) with respect to the interface and $\widehat{\mathbf{g}} = \widehat{\mathbf{f}} - \mathbf{K}_{ro}\mathbf{K}_{oo}^{-1}\mathbf{f}_o$ is sometimes called *condensed right-hand side*. Interior unknowns $\mathbf{u}_o$ are determined by interface unknowns $\widehat{\mathbf{u}}$ via the system of equations $\mathbf{K}_{oo}\mathbf{u}_o = \mathbf{f}_o - \mathbf{K}_{or}\widehat{\mathbf{u}}$, 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

Historically, primal DD methods were used so that in every iteration a residual $\widehat{\mathbf{r}}$ of (3) was computed and split into subdomains, then local problems were solved and then the global solution on the interface was computed as a weighted average of the local solutions. This can be written as Richardson iterations $\widehat{\mathbf{u}}_{i+1} = \widehat{\mathbf{u}}_i + \mathbf{M}\widehat{\mathbf{r}}_i$, where $\mathbf{M}$ stands for operator $M = ES^{-1}E^T$, with E^T representing splitting of the residual to subdomains, S^{-1} representing solution of subdomain problems, and E representing projection of subdomain solutions back to the global problem by some averaging.

Presently, primal DD methods are mostly used as preconditioners for problem (3) within the PCG method. In every iteration of the PCG method, a preconditioned residual is computed using the DD preconditioner M, which is realized by one step of the corresponding DD method. The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq \|RE\|_S^2, \tag{4}$$

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 (4) was proved in [4] 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 use independent subdomain problems only (no continuity conditions across the interface), the operator S is expressed by a block diagonal matrix $\mathbf{S}$ with diagonal blocks $\mathbf{S}_i$ representing local Schur complements on subdomains. Relationship between global and local problems can be expressed in matrix form as

$$\widehat{\mathbf{S}} \;=\; \mathbf{R}^{\mathrm{T}}\mathbf{S}\mathbf{R} \;=\; \sum_i \mathbf{R}^{i\mathrm{T}}\mathbf{S}^i\mathbf{R}^i\,, \quad \mathbf{u} = \mathbf{R}\widehat{\mathbf{u}}, \quad \widehat{\mathbf{u}} = \mathbf{E}\mathbf{u}, \tag{5}$$

where $\mathbf{R}^i$ represents prolongation operator from local (subdomain) interface to the global interface and $\mathbf{E}$ performs some averaging.

The main idea of the BDDC ([1]) 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 except 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 and it is processed by the global coarse problem. However, it still can be formally written like (5).

4. Choice of the averaging operator E

Standard choices of E are arithmetic average, average weighted by diagonal entries of matrices of local problems or a ratio of corresponding diagonal entries of local and global Schur complements.

General formula introduced in [5] for computing local weights α_k^i in i-th subdomain is

$$\alpha_k^i = \mathbf{d}_k^T \left(\mathbf{R}^{i\mathrm{T}}\mathbf{S}^i\mathbf{R}^i\right)\mathbf{d_k} \,/\, \mathbf{d}_k^T\,\widehat{\mathbf{S}}\,\mathbf{d}_k \tag{6}$$

for some choice of a set of test vectors $\mathbf{d}_k$. One option is to choose all the cartesian basis vectors $\mathbf{e}_k$, which leads to a standard choice of using diagonal entries of Schur complements. Our proposition is to choose several test vectors with nonzero values at some selected nodes only and obtain some average value for these nodes (typically face or edge, or only part of it). In this paper we tested only one variant described below.

Variant I (tailored for 2D problems)

Select all cross-points as coarse. Then for i-th subdomain, for every face F_{ij} between i-th and j-th subdomain, choose only one vector $\mathbf{d}_j$ in formula (6), with ones in positions corresponding to nodes belonging to face F_{ij} and zeros elsewhere. This leads to just one value of α_j^i assigned to every interface node of the face F_{ij} in i-th subdomain.

5. Numerical results

For numerical comparison of different methods of averaging, a 2D Poisson equation on a rectangular domain with homogeneous Dirichlet boundary condition on the opposite sides was chosen. The methods were tested first on the equation with constant coefficients and then on the equation with jump in the coefficients. Coefficients are prescribed as piecewise constant on subdomains with jump in the rate of 1:9 along the part of the interface parallel with y axis.

The problem was discretized by FEM using 35 bilinear elements. The FEM solution is depicted in the left side of Figure 1. The domain was divided into four rectangular subdomains different in size, all of which have one side formed by the part of the boundary with prescribed Dirichlet boundary condition. Only one continuity constraint across the interface was chosen: A coarse node in the cross-point of all four subdomains. Solution in the space of functions continuous in this coarse node only is depicted in the right side of Figure 1.
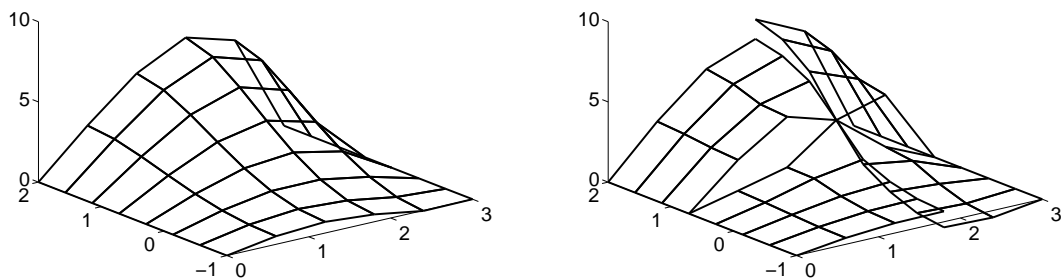

Figure 1: Test problem. 2D Poisson equation, bilinear elements, 4 subdomains, 1 coarse node chosen as the cross-point of all the subdomains. Continuous solution (left) and solution with discontinuities across the interface (right).

BDDC was used in two different ways: either as a preconditioner combined with PCG, or as a Richardson iteration method. The second technique is not the standard use of the BDDC method, we choose it just for testing the numerical properties of different methods of averaging.

Three different methods for choice of the averaging operator E were tested:

a) Arithmetic average.
b) Weighted average using ratio of corresponding diagonal entries of local and global Schur complements.
c) The variant of (6) described in section 4 as Variant I.

Table 1 contains results for BDDC used as a Richardson method, Table 2 for standard BDDC as a preconditioner combined with PCG. In both tables there are

without jumps in coefficients			with jump in coefficients		
arithm. av.	diag. Sch.	$d = (1, ..1)$	arithm. av.	diag. Sch.	$d = (1, ..1)$
1.52	1.50	1.44	2.31e+01	1.18	1.16
3.78e–01	3.71e–01	3.54e–01	6.73e+01	2.68e–01	2.59e–01
9.36e–02	9.10e–02	8.61e–02	1.96e+02	6.86e–02	6.59e–02
2.32e–02	2.24e–02	2.10e–02	5.66e+02	1.96e–02	1.84e–02
5.76e–03	5.50e–03	5.17e–03	1.63e+03	5.64e–03	5.37e–03

Table 1: Errors in the first 5 iterations, BDDC used as **Richardson method**. Poisson equation with constant coefficients (left) and with jump in coefficients along the interface (right).

without jumps in coefficients			with jump in coefficients		
arithm. av.	diag. Sch.	$d = (1, ..1)$	arithm. av.	diag. Sch.	$d = (1, ..1)$
1.27	1.26	1.22	6.32	1.05	1.03
1.36e–02	1.26e–02	1.83e–02	1.67	6.29e–02	6.10e–02
1.21e–03	1.13e–03	1.51e–03	1.21e–02	4.30e–04	5.05e–04
2.82e–06	2.48e–06	1.43e–06	2.60e–03	9.07e–07	1.07e–06
1.69e–09	1.51e–09	2.02e–09	5.64e–05	2.09e–10	4.24e–10

Table 2: Errors in the first 5 iterations, BDDC used as a preconditioner for **PCG**. Poisson equation with constant coefficients (left) and with jump in coefficients along the interface (right).

norms of errors (Euclidean norm of vector of differences from exact FEM solution at nodes) at first 5 iterations as a measure of the rate of convergence. In the left part of the tables there are results for equation without jump in coefficients, in the right part there are results for the jump.

Results show superiority of preconditioned PCG over Richardson iterations as expected.

As for comparison of the three methods of averaging, for equation without jump in coefficients (the first three columns of the tables), all three methods give very similar rate of convergence in both ways BDDC was used. If we look at the slight differences, we can see that if Richardson iteration is used, method c) gives the best results and method a) the worst ones in every iteration, while if PCG is used, rating of the methods changes in every iteration so it is difficult to decide which one is best.

However, for equation with jump in coefficients (the last three columns of the tables), there is remarkable difference between the arithmetic average a) and the other two methods. In Richardson iteration the method a) does not lead to convergence at all and in combination with PCG it leads to remarkably slower convergence than the other methods. The other two methods are mutually comparable, method c) gives slightly better results in Richardson iteration and method b) is better in combination with PCG.

6. Conclusions

We compared three different methods of averaging in BDDC method: two standard ones (arithmetic average and weighted average using ratio of corresponding diagonal entries of local and global Schur complements) and one based on the new proposition (6). A simple test of Poisson equation in 2D indicates that there is no essential difference among the methods if there are no jumps in coefficients. On the other hand, if there is a jump along the interface, arithmetic average performs significantly worse than either one of the other two methods, which are mutually comparable.

It seems that without presence of jumps in coefficients, the method of choice is using arithmetic averages, because it is the simplest and also cheapest approach to implementation. If the problem involves jumps in coefficients aligned to subdomain boundaries, the choice of weighted averages based on diagonals of the Schur complements and the new approach seems equally efficient. In the most efficient implementations, the Schur complements are not computed explicitly, and so the new approach seems preferable. However, numerical results presented in this paper are just preliminary and need to be confirmed by results for problems involving large number of subdomains. We plan to perform such experiments in future.

References

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

[2] 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.

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

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

[5] Čertíková, M., Burda, P., Novotný, J., Šístek, J.: Some remarks on averaging in the BDDC method. In: Vejchodský, T., Chleboun, J., Přikryl, P., Segeth, K., Šístek, J. (Eds.), *Proceedings of Seminar PANM'10*, pp. 28–34. Dolní Maxov, Czech Republic, June 6 – 11. Institute of Mathematics AS CR, Prague, 2010.