Reynolds-Blended Weights for BDDC in Applications to Incompressible Flows

Martin Hanek, Jakub Šístek, and Marek Brandner

1 Introduction

We investigate the applicability of the Balancing Domain Decomposition by Constraints (BDDC) method to numerical solution of problems of incompressible flows. In particular, we use BDDC to solving linear systems with a nonsymmetric matrix arising from discretization of the Navier–Stokes equations by the finite element method.

The BDDC method was introduced by Dohrmann in [1] for the Poisson problem and linear elasticity. The underlying theory for the condition number bound of $O\left(\log^2(1 + H/h)\right)$ was presented by Mandel and Dohrmann in [5]. By discretizing and linearizing the Navier-Stokes equations, we get saddle-point systems with nonsymmetric matrices. An application of the BDDC method to nonsymmetric matrices arising from advection-diffusion problems was presented by Tu and Li [9], where the method was formulated without building and solving an explicit coarse problem. Finding explicit coarse basis functions and forming an explicit coarse problem of BDDC was presented by Yano for nonsymmetric problems arising from the Euler equations in [10]. A three-level extension of BDDC was presented by Tu [8], while a general multilevel method was introduced and analysed for symmetric positive definite problems by Mandel et al. [6]. We have extended the multilevel BDDC method

Martin Hanek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic,
Czech Technical University in Prague, Technická 4, Prague, Czech Republic,
e-mail: martin.hanek@fs.cvut.cz

Jakub Šístek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic,
e-mail: sistek@math.cas.cz

Marek Brandner
University of West Bohemia in Pilsen, Faculty of Applied Sciences, Univerzitní 22, Pilsen, Czech Republic, e-mail: brandner@kma.zcu.cz

to nonsymmetric matrices in [3]. A theoretically supported approach for handling continuous pressure in the Stokes problem was introduced in [4].

An important building block of BDDC as well as other nonoverlapping domain decomposition methods is the choice of weights used for averaging a discontinuous solution at the interface between subdomains. Standard types of weights include an arithmetic average (also known as cardinality scaling), or weighted average based on diagonal entries of subdomain matrices. In [3], we have also presented a novel averaging operator tailored to Navier-Stokes equations. The main idea behind it is using the current approximation of velocity for preferring information opposite the flow. Due to the similarity of this idea with numerical methods for convection dominated flows, we called this choice as the upwind scaling.

In this contribution, we present a modification of the upwind scaling. While the upwind scaling is superior for flows at higher Reynolds numbers, the simple arithmetic scaling tends to perform better for flows at lower Reynolds numbers. For this reason, we 'blend' the arithmetic and upwind scalings with the ratio based on the local Reynolds number, and we call the proposed method as *Reynolds-blended (Re-blended)* weights.

The rest of the paper is organized as follows. In Section 2, we recall the basics of iterative substructuring and BDDC for the nonsymmetric saddle-point systems arising from the finite element method (FEM). The new weights are proposed in Section 3. Section 4 presents results of numerical experiments showing the benefits of the Re-blended weights, while Section 5 is devoted to the summary.

2 FEM and BDDC for Navier-Stokes equations

We consider a stationary incompressible flow in a bounded three-dimensional domain $\Omega \subset \mathbb{R}^3$ with its boundary $\partial\Omega$ consisting of two disjoint parts $\partial\Omega_D$ and $\partial\Omega_N$, governed by the Navier-Stokes equations (see e.g. [2]),

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{2}$$

where $\boldsymbol{u}$ is the velocity vector of the fluid, ν is the kinematic viscosity of the fluid, p is the kinematic pressure, and $\boldsymbol{f}$ is the vector of body forces. In addition, we consider the following boundary conditions: prescribed velocity on $\partial\Omega_D$ and $-\nu(\nabla\boldsymbol{u})\boldsymbol{n} + p\boldsymbol{n} = \mathbf{0}$ on $\partial\Omega_N$, with $\boldsymbol{n}$ being the unit outer normal vector of $\partial\Omega$.

We consider Taylor-Hood Q2-Q1 elements, and after substituting linear combinations of the basis functions, we get the following system of algebraic equations

$$\begin{bmatrix} \nu\mathbf{A} + \mathbf{N}(\mathbf{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{3}$$

Details can be found in [3].

System (3) is nonlinear due to the matrix $\mathbf{N}(\mathbf{u})$, and we consider the Picard iteration for its linearization. This leads to solving a sequence of linear systems of equations in the form

$$\begin{bmatrix} \nu\mathbf{A} + \mathbf{N}(\mathbf{u}^p) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{p+1} \\ \mathbf{p}^{p+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{4}$$

Linear system (4) is solved by means of iterative substructuring (see, e.g., [7]). In order to use the BDDC method, we decompose the solution domain Ω into N nonoverlapping subdomains. Then we reduce the system (4) to the interface to get

$$S \begin{bmatrix} \mathbf{u}_\Gamma \\ \mathbf{p}_\Gamma \end{bmatrix} = g, \tag{5}$$

where S is the Schur complement of the interior unknowns and g is the reduced right-hand side.

Problem (5) is solved by the BiCGstab method using one step of BDDC as the preconditioner. In each action of the BDDC preconditioner, a coarse problem and independent subdomain problems are solved. Before solving it in each iteration, we need to set-up the preconditioner. This is performed by solving two saddle-point systems

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad \begin{bmatrix} S_i^T & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i^* \\ \Lambda_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{6}$$

where S_i is the Schur complement with respect to the interface of the i-th subdomain, C_i is the matrix defining coarse degrees of freedom, which has as many rows as is the number of coarse degrees of freedom defined at the subdomain. The solution Ψ_i is the matrix of *coarse basis functions* with every column corresponding to one coarse unknown on the subdomain. These functions are equal to one in one coarse degree of freedom, and they are equal to zero in the remaining local coarse unknowns. The solution Ψ_i^* is the matrix of *adjoint coarse basis functions* which is needed for nonsymmetric problems as was shown in [10]. The coarse problem matrix is assembled in the setup of the BDDC preconditioner as $S_C = \sum_{i=1}^N R_{Ci}^T \Psi_i^{*T} S_i \Psi_i R_{Ci}$.

One step of the BDDC preconditioner $M_{BDDC} : r^l \to u_\Gamma^l$ proceeds as follows:

$$r_i^l = W_i R_i r^l$$

coarse problem subdomain problems

$$r_C^l = \sum_{i=1}^N R_{Ci}^T \Psi_i^{*T} r_i^l$$

$$S_C u_C = r_C^l$$

$$u_{Ci} = \Psi_i r_{Ci}^l u_C$$

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} u_i \\ \lambda \end{bmatrix} = \begin{bmatrix} r_i^l \\ 0 \end{bmatrix}$$

$$u_\Gamma^l = \sum_{i=1}^N R_i^T W_i (u_i + u_{Ci}),$$

where R_i is an operator restricting a global interface vector to the i-th subdomain, R_{Ci} is the restriction of the global vector of coarse unknowns to those present at the i-th subdomain, and matrix W_i applies weights to satisfy the partition of unity, which will be elaborated in the next section. Details of the application of this method to Navier-Stokes equations can be found in [3].

3 Weight operators

Let us now discuss several particular choices of the matrix of weights W_i. An important class of these matrices is represented by diagonal matrices

$$W_i = \begin{pmatrix} W_{iN}^1 & & \\ & W_{iN}^2 & \\ & & \ddots \end{pmatrix}, \tag{7}$$

where W_{iN}^k denotes the weight matrix for the unknowns in the k-th (with respect to the subdomain interface) node of the i-th subdomain. These matrices differ for nodes with just velocity unknowns and those containing also a pressure unknown ordered after the velocity ones. For example, in 3D the former and latter looks respectively as

$$W_{iN}^k = \begin{pmatrix} w_i^k & & \\ & w_i^k & \\ & & w_i^k \end{pmatrix}, \qquad W_{iN}^k = \begin{pmatrix} w_i^k & & & \\ & w_i^k & & \\ & & w_i^k & \\ & & & \frac{1}{N_S} \end{pmatrix}, \tag{8}$$

where N_S is the number of subdomains sharing the node.

A general scheme for constructing these matrices satisfying the partition of unity can be described in the following way. Every subdomain first generates a nonnegative weight $\widetilde{w}_i^k$. These values are then shared with all neighbouring subdomains, and the normalized weight w_i^k satisfying the partition of unity is obtained by dividing the local weight with the sum of contributions from all neighbours,

$$w_i^k = \frac{\widetilde{w}_i^k}{\sum_{j=1}^{N_S} \widetilde{w}_j^k}. \tag{9}$$

The first type of weights is based on the cardinality (*card*) of the set of subdomains sharing the node. Hence, $\widetilde{w}_i^k = 1$, and

$$w_i^k = \frac{1}{N_S}. \tag{10}$$

For example, the weight is simply $w_i^k = 1/2$ if the node is shared by two subdomains.

The second type of weights was introduced in [3], and it is inspired by numerical schemes for flow problems, namely by upwinding. The underlying idea is that for dominant advection, it should be beneficial to consider the subdomain from which the fluid flows with a higher weight than for the one where the node is a part of an inflow boundary.

More specifically, these *upwind* weights are based on the inner product of the vector of velocity at the k-th interface node $\boldsymbol{u}^k$ and the unit vector of the outer normal to the i-th subdomain boundary $\boldsymbol{n}_i^k$, therefore

$$p_i^k = \frac{\boldsymbol{u}^k \cdot \boldsymbol{n}_i^k}{\left\|\boldsymbol{u}^k\right\|_2}.$$

The values of the p_i^k are from the interval $[-1, 1]$. To derive a nonnegative weight, these values are mapped to the interval $[0, 1]$ by taking $\widetilde{w}_i^k = \frac{p_i^k+1}{2}$, which is used for all velocity unknowns. More details, such as the discrete construction of $\boldsymbol{n}_i^k$, can be found in [3].

The third type is the new approach obtained by linear interpolation of the previous two weights. For this method, we choose a critical Reynolds number Re_C, and then the resulting Reynolds-blended ($\mathrm{Re}_{\mathrm{blended}}$) weight is defined according to the local Reynolds number $\mathrm{Re}_{\mathrm{loc}} = |\boldsymbol{u}^k| L/\nu$ as

$$\widetilde{w}_i^k = \begin{cases} \widetilde{w}^k_{\mathrm{card}} & \text{for } \mathrm{Re}_{\mathrm{loc}} \leq 1, \\ \frac{\mathrm{Re}_{\mathrm{loc}}}{\mathrm{Re}_C} \widetilde{w}^k_{\mathrm{upwind}} + \left(1 - \frac{\mathrm{Re}_{\mathrm{loc}}}{\mathrm{Re}_C}\right) \widetilde{w}^k_{\mathrm{card}} & \text{for } 1 < \mathrm{Re}_{\mathrm{loc}} < \mathrm{Re}_C, \\ \widetilde{w}^k_{\mathrm{upwind}} & \text{for } \mathrm{Re}_{\mathrm{loc}} \geq \mathrm{Re}_C. \end{cases} \tag{11}$$

Here L corresponds to the characteristic length of the problem. Thus for small local Reynolds numbers, the scaling behaves as cardinality weights and for high Reynolds number as upwind weights depending on the chosen critical Reynolds number Re_C. Note that these weights are updated after each nonlinear iteration.

4 Numerical results

In this section, we compare the behaviour of the 2-level BDDC method for different types of interface weights described in Section 3, namely the cardinality scaling (*card*), *upwind*, and the proposed $\mathrm{Re}_{\mathrm{blended}}$ weights. We assume two problems, namely the lid-driven cavity and the backward facing step problems. First we look at the cavity problem. We consider unit cube with unit velocity on the top wall as in [2]. For $\mathrm{Re}_{\mathrm{blended}}$, we consider two critical Reynolds numbers, $\mathrm{Re}_C = 100$ and $\mathrm{Re}_C = 200$. For these simulations, the number of subdomains is 125 with 8 elements per subdomain edge. The decomposed solution domain can be seen in Fig. 1. For this problem, Reynolds number is defined as $\mathrm{Re} = |\boldsymbol{u}_{\mathrm{top}}| L/\nu$, where $|\boldsymbol{u}_{\mathrm{top}}| = 1$ is the velocity at the lid, and $L = 1$ is the cube size. We compare $\mathrm{Re} = 1$ and $\mathrm{Re} = 200$

monitoring the number of nonlinear iterations, the minimal, maximal, and mean number of linear iterations over all nonlinear iterations, the mean setup time of the BDDC preconditioner, the mean time for the Krylov subspace method with the mean time for one linear iteration, the mean time for one nonlinear iteration, and the time for all nonlinear iterations.

The computations are performed on the *Karolina* supercomputer at the IT4I National Supercomputing Centre in Ostrava, Czech Republic. The computational nodes are equipped with two 64-core AMD 7H12 2.6 GHz processors, and 256 GB RAM. The values are presented in Tables 1 and 2.

Table 1 Re = 1. Number of nonlinear iterations, number of linear iterations (minimal, maximal, and mean), mean setup time, time for the BiCGstab iterations, time for one linear iteration, time for one nonlinear iteration, and the total time for solving the nonlinear problem.

weights type	nonl	linear solve			time [s]			
		min	max	mean	setup	BiCGstab iter (one iter)	nonl	total
card	4	13.5	13.5	**13.5**	4.50	4.07 (0.30)	8.57	32.28
upwind	4	13.5	18.5	**17.3**	4.53	5.20 (0.30)	9.73	38.92
Re-blended (Re_C = 100)	4	13.5	13.5	**13.5**	4.58	4.09 (0.30)	8.67	34.68
Re-blended (Re_C = 200)	4	13.5	13.5	**13.5**	4.51	4.10 (0.30)	8.61	34.44

Table 2 Re = 200. Number of nonlinear iterations, number of linear iterations (minimal, maximal, and mean), mean setup time, time for the BiCGstab iterations, time for one linear iteration, time for one nonlinear iteration, and the total time for solving the nonlinear problem.

weights type	nonl	linear solve			time [s]			
		min	max	mean	setup	BiCGstab iter (one iter)	nonl	total
card	29	14	91.5	**86.2**	5.01	27.99 (0.32)	33.0	1517.2
upwind	29	14	85.5	**34.2**	4.99	10.79 (0.32)	15.78	1029.1
Re-blended (Re_C = 10)	29	14	85.5	**34.0**	4.99	10.76 (0.32)	15.75	1028.6
Re-blended (Re_C = 100)	29	14	89	**137.8**	4.99	12.15 (0.32)	17.14	1069.5
Re-blended (Re_C = 200)	29	14	137.5	**58.1**	5.05	18.22 (0.31)	23.27	1240.44

From Tables 1 and 2, we can see that for small Reynolds numbers, the cardinality weight is slightly more efficient and for the high Reynolds number the same stands for the upwind weight. The critical Reynolds weight seems to benefit from both depending on the Reynolds number. For Re = 1, it inclines to the cardinality and for Re = 200 to the upwind weight.

Let us now explore the effect of the Reynolds-blended weight on the backward facing step problem. This problem was investigating in [2] in 2D. The solution domain is shown in Fig. 1 with prescribed unit inlet velocity, zero velocity on the top and bottom walls, and symmetry boundary condition on the side walls. With the x-axis aligned with the flow, the step occurs for $x = 1$, where the height changes from 1 to 2. The length of the domain is 5, and its width is 1. The solution domain consists of 37 thousand elements which correspond to 978 thousand unknowns. The mesh is decomposed into 32 subdomains using a vertical partitioner, which cuts the domain along the x direction (see Fig. 1). The Reynolds number is defined as $\mathrm{Re} = |\boldsymbol{u}_{\mathrm{inlet}}|L/\nu$, where $|\boldsymbol{u}_{\mathrm{inlet}}| = 1$ is the input velocity, and $L = 1$ is the size of the narrow part.

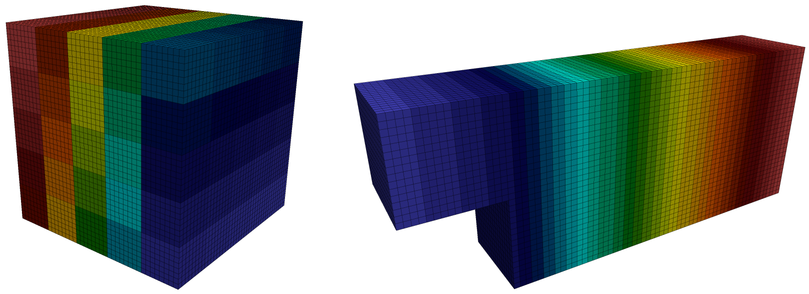

Fig. 1 Decomposed solution domain for the cavity problem (left) and for the backward facing step problem (right).

We set the critical Reynolds number for our new weight to 20 and plot the mean number of linear iterations and the average time for solving one linearized problem (4) depending on Reynolds number in Fig. 2.

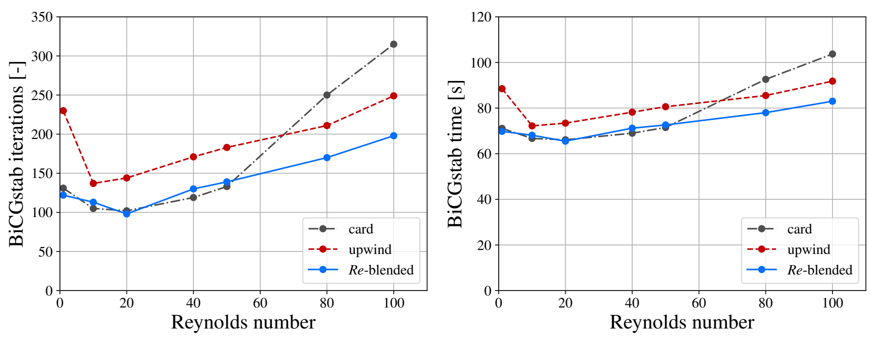

Fig. 2 Number of BiCGstab iterations (left) and average time for solving one linearized problem (right) for different Reynolds numbers for cardinality, upwind, and Re-blended operators.

From these plots we can see that up to a certain Re, cardinality performs better while for larger Re, the upwind is more effective. The Reynolds-blended weight operator with a suitably chosen critical Reynolds number Re_C provides the best results for almost every Re, and therefore it again combines advantages of cardinality and upwind weight operator. Interestingly, it even outperforms the upwind weight operator. This positive effect is attributed to the fact that the blending based on the local Reynolds number $\mathrm{Re}_{\mathrm{loc}}$ reduces the effect of upwinding in zones with reduced velocity such as in boundary layers.

5 Conclusions

We have presented a new scaling operator for the BDDC method in applications to saddle-point linear systems arising from discretization of the Navier-Stokes equations. It can be seen as a correction of the recent upwind operator when applied to flows with low Reynolds numbers, for which arithmetic scaling is superior. We have compared the relevant weight operators on the cavity and the backward facing step problems. The results demonstrate the intended behaviour of the new scaling, namely mimicking the arithmetic averaging for low Re and the upwind scaling for high Re. Although our simulations show promising results for the considered small and moderate Reynolds numbers, for larger Re some kind of stabilization of the discretization would be needed. Investigating the performance of the new method for other flow problems and the choice of the Re_C parameter will be a matter of our future research.

Acknowledgements This research was supported by the Czech Science Foundation through grants GAČR 20-01074S and GAČR 23-06159S, by the Czech Academy of Sciences through RVO:67985840, and by the Czech Technical University in Prague through RVO:12000. Computational time on the Karolina supercomputer has been provided thanks to the support of the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

References

1. Dohrmann, C. R. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.* **25**(1), 246–258 (2003).
2. Elman, H. C., Silvester, D. J., and Wathen, A. J. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2005).
3. Hanek, M., Šístek, J., and Burda, P. Multilevel BDDC for incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.* **42**(6), C359–C383 (2020).
4. Li, J. and Tu, X. A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures. *SIAM J. Numer. Anal.* **51**(2), 1235–1253 (2013).
5. Mandel, J. and Dohrmann, C. R. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.* **10**(7), 639–659 (2003).
6. Mandel, J., Sousedík, B., and Dohrmann, C. R. Multispace and multilevel BDDC. *Computing* **83**(2-3), 55–85 (2008).
7. Toselli, A. and Widlund, O. B. *Domain Decomposition Methods—Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005).
8. Tu, X. Three-level BDDC in three dimensions. *SIAM J. Sci. Comput.* **29**(4), 1759–1780 (2007).
9. Tu, X. and Li, J. A balancing domain decomposition method by constraints for advection-diffusion problems. *Commun. Appl. Math. Comput. Sci* **3**(1), 25–60 (2008).
10. Yano, M. *Massively Parallel Solver for the High-Order Galerkin Least-Squares Method*. Master's thesis, Massachusests Institute of Technology (2009).