Martin Hanek; Jakub Šístek; Pavel Burda

An application of the BDDC method to the Navier-Stokes equations in 3-D cavity

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. 77--85.

Persistent URL: http://dml.cz/dmlcz/702666

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

AN APPLICATION OF THE BDDC METHOD TO THE NAVIER-STOKES EQUATIONS IN 3-D CAVITY

Martin Hanek[1], Jakub Šístek[2], Pavel Burda[1]

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

Abstract

We deal with numerical simulation of incompressible flow governed by the Navier-Stokes equations. The problem is discretised using the finite element method, and the arising system of nonlinear equations is solved by Picard iteration. We explore the applicability of the Balancing Domain Decomposition by Constraints (BDDC) method to nonsymmetric problems arising from such linearisation. One step of BDDC is applied as the preconditioner for the stabilized variant of the biconjugate gradient (BiCGstab) method. We present results for a 3-D cavity problem computed on 32 cores of a parallel supercomputer.

1. Introduction

The Balancing Domain Decomposition by Constraints (BDDC) was developed by Dohrmann in [1] as an efficient method to solve large systems of linear equations arising from partial differential equations discretised by the finite element method. In [1], the method was applied to elliptic problems, namely Poisson problem and linear elasticity. BDDC was extended to the incompressible Stokes problem in [4] considering finite elements with discontinuous approximation of pressure. In [6], the BDDC method was applied to the Stokes problem discretised by Taylor-Hood finite elements with continuous pressure approximation. The interface problem in this monolithic approach contains both velocity and pressure unknowns. An alternative approach was presented in [3]. A generalisation of the BDDC method for systems with nonsymmetric matrix was proposed in [12] and applied to Euler equations of inviscid compressible flows.

In our contribution, we combine the approach to building the interface problem from [6] with the extension to nonsymmetric problems from [12]. The algorithm is applied to nonsymmetric linear systems obtained by Picard's linearisation of the steady Navier-Stokes equations using Taylor-Hood finite elements. Numerical results for flow inside a 3-D lid driven cavity are presented.

2. Navier-Stokes equations and the finite element method

We consider stationary flow of incompressible fluid in three spatial dimensions, governed by the Navier-Stokes equations without body forces (see e.g. [2])

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{0} \quad \text{in } \Omega, \tag{1}$$

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

where $\boldsymbol{u} = (u_1, u_2, u_3)^T$ is an unknown velocity vector, p is an unknown pressure normalised by (constant) density, ν is a given kinematic viscosity, and Ω is the solution domain. In addition, the following boundary conditions are considered

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \Gamma_D, \tag{3}$$

$$-\nu(\nabla\boldsymbol{u})\mathbf{n} + p\mathbf{n} = 0 \qquad \text{on } \Gamma_N, \tag{4}$$

where Γ_D and Γ_N are parts of the boundary $\partial\Omega$, $\overline{\Gamma_D} \cup \overline{\Gamma_N} = \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\mathbf{n}$ is the outer unit normal vector of the boundary, and $\boldsymbol{g}$ is a given function.

2.1. Weak formulation

In deriving the weak mixed formulation, we multiply equations (1)–(2) by test functions and integrate over the solution domain. Then using the divergence theorem, we get the final weak formulation

We seek $\boldsymbol{u} \in V_g$ *and* $p \in L^2(\Omega)$*, satisfying*

$$\int_\Omega (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{v} \mathrm{d}\Omega + \nu \int_\Omega \nabla\boldsymbol{u} : \nabla\boldsymbol{v} \mathrm{d}\Omega - \int_\Omega p\nabla \cdot \boldsymbol{v} \mathrm{d}\Omega = 0 \quad \forall \boldsymbol{v} \in V, \tag{5}$$

$$\int_\Omega q\nabla \cdot \boldsymbol{u} \mathrm{d}\Omega = 0 \quad \forall q \in L^2(\Omega). \tag{6}$$

Here the spaces are

$$V_g := \left\{ \boldsymbol{u} \in H^1(\Omega)^3, \boldsymbol{u} = \boldsymbol{g} \text{ on } \Gamma_D \right\},$$
$$V := \left\{ \boldsymbol{v} \in H^1(\Omega)^3, \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_D \right\}.$$

2.2. Assembly of the system of algebraic equations

During the assembly of the system of algebraic equations, we substitute into the weak formulation (5)–(6) for $\boldsymbol{u}$, p, $\boldsymbol{v}$, and q their finite element counterparts

$$\boldsymbol{u}_h = \sum_{i=1}^{3n_{\boldsymbol{u}}} u_i \boldsymbol{\phi}_i, \quad p_h = \sum_{i=1}^{n_p} p_i \psi_i, \quad \boldsymbol{v}_h = \sum_{i=1}^{3n_{\boldsymbol{u}}} v_i \boldsymbol{\phi}_i, \quad q_h = \sum_{i=1}^{n_p} q_i \psi_i.$$

Here $\boldsymbol{\phi}_i$ are vector basis functions for velocity, ψ_i are scalar basis functions for pressure, $n_{\boldsymbol{u}}$ is the number of nodes with velocity unknowns, and n_p is the number of nodes with pressure unknowns. For the considered hexahedral Taylor-Hood finite elements (see e.g. [2]), $n_{\boldsymbol{u}}$ is approximately eight times larger than n_p.

We obtain the following system of algebraic equations

$$\begin{bmatrix} \nu\mathbf{A}+\mathbf{N}(\mathbf{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{7}$$

where $\mathbf{u}$ is the vector of unknown coefficients of velocity, $\mathbf{p}$ is the vector of unknown coefficients of pressure, $\mathbf{A}$ is the matrix of diffusion, $\mathbf{N}(\mathbf{u})$ is the matrix of advection which depends on the solution, B is the matrix from continuity equation, and $\mathbf{f}$ and $\mathbf{g}$ are discrete right-hand side vectors arising from Dirichlet boundary conditions. Each part of system (7) is assembled as (see [2])

$$\mathbf{A} = [a_{ij}],\ a_{ij} = \int_\Omega \nabla\boldsymbol{\phi}_i : \nabla\boldsymbol{\phi}_j \,\mathrm{d}\Omega, \tag{8}$$

$$\mathbf{N}(\mathbf{u}) = [n_{ij}],\ n_{ij} = \int_\Omega (\mathbf{u}\cdot\nabla)\boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_i \,\mathrm{d}\Omega, \tag{9}$$

$$B = [b_{lj}],\ b_{lj} = -\int_\Omega \psi_l \nabla\cdot\boldsymbol{\phi}_j \,\mathrm{d}\Omega, \tag{10}$$

$$\mathbf{f} = [f_i],\ f_i = -\sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega (\mathbf{u}\cdot\nabla)\boldsymbol{\phi}_j\cdot\boldsymbol{\phi}_i \,\mathrm{d}\Omega - \nu \sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega \nabla\boldsymbol{\phi}_j : \nabla\boldsymbol{\phi}_i \,\mathrm{d}\Omega, \tag{11}$$

$$\mathbf{g} = [g_l],\ g_l = \sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega \psi_l \nabla\cdot\boldsymbol{\phi}_j \,\mathrm{d}\Omega. \tag{12}$$

System (7) is nonlinear due to the matrix $\mathbf{N}(\mathbf{u})$, and for its linearisation, we use the Picard iteration. This leads to solving a sequence of linear systems of equations in the form

$$\begin{bmatrix} \nu\mathbf{A}+\mathbf{N}(\mathbf{u}^k) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{13}$$

where $\mathbf{N}(\mathbf{u}^k)$ means that we substitute a solution of velocity from the previous step to the matrix $\mathbf{N}$. This—already linear—nonsymmetric system is solved by means of iterative substructuring.

3. Iterative substructuring

For our calculations, we use decomposition of domain Ω into N nonoverlapping subdomains. In order to explain how the BDDC algorithm fits to problem (13), we assume reordering of unknowns within $\mathbf{u}$ and $\mathbf{p}$ such that the components corresponding to the nodes on the interface are at the end. This leads to the following blocking of the system

$$\begin{bmatrix} \nu\mathbf{A}_{11}+\mathbf{N}_{11} & \nu\mathbf{A}_{12}+\mathbf{N}_{12} & B_{11}^T & B_{21}^T \\ \nu\mathbf{A}_{21}+\mathbf{N}_{21} & \nu\mathbf{A}_{22}+\mathbf{N}_{22} & B_{12}^T & B_{22}^T \\ B_{11} & B_{12} & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{p_1} \\ \mathbf{p_2} \end{bmatrix} = \begin{bmatrix} \mathbf{f_1} \\ \mathbf{f_2} \\ \mathbf{g_1} \\ \mathbf{g_2} \end{bmatrix}, \tag{14}$$

where subscript $_1$ denotes the part with interior unknowns and subscript $_2$ denotes the part with interface unknowns. The whole blocks are now permuted to get an interface problem, similarly as it was done for the Stokes problem in [6],

$$S \begin{bmatrix} \mathbf{u_2} \\ \mathbf{p_2} \end{bmatrix} = g. \tag{15}$$

Here

$$S = \begin{bmatrix} \nu\mathbf{A}_{22} + \mathbf{N}_{22} & B_{22}^T \\ B_{22} & 0 \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{21} + \mathbf{N}_{21} & B_{12}^T \\ B_{21} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{11} + \mathbf{N}_{11} & B_{11}^T \\ B_{11} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nu\mathbf{A}_{12} + \mathbf{N}_{12} & B_{21}^T \\ B_{12} & 0 \end{bmatrix}$$

is the Schur complement with respect to the interface, and

$$g = \begin{bmatrix} \mathbf{f_2} \\ \mathbf{g_2} \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{21} + \mathbf{N}_{21} & B_{12}^T \\ B_{21} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{11} + \mathbf{N}_{11} & B_{11}^T \\ B_{11} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f_1} \\ \mathbf{g_1} \end{bmatrix}$$

is the reduced right-hand side.

Problem (15) is solved by the BiCGstab method [10], and one step of BDDC is used as a preconditioner. Thanks to domain decomposition, both the action of the BDDC preconditioner and of the matrix S are parallelised in each iteration. This is realised by the multilevel BDDC implementation in the *BDDCML* library[1] (version 2.4) [8] employed in our computations.

4. BDDC for nonsymmetric systems

The BDDC preconditioner works with a residuum r^k obtained from the k-th iteration of the BiCGstab algorithm

$$r^k = g - S \begin{bmatrix} \mathbf{u_2}^k \\ \mathbf{p_2}^k \end{bmatrix}. \tag{16}$$

The preconditioner provides an approximate solution to problem (15), and it is realised by one iteration of the BDDC method.

A key idea of BDDC is to choose suitable *coarse degrees of freedom*, and then seek solution on the interface in a space of functions that are continuous in these coarse degrees of freedom. Although more advanced choices were introduced for advection-diffusion problem in [9], we restrict ourselves in this study to continuity at coarse nodes, which are selected according to [7], and continuity of arithmetic averages over all faces and edges enforced independently for each component of velocity and for pressure.

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

In each action of the BDDC preconditioner, a coarse problem and independent subdomain problems are solved. First we look at one subdomain problem. It takes the total residuum r^k and extracts a local part on the subdomain as

$$r_i = W_i R_i r^k, \tag{17}$$

where R_i is an operator restricting a global interface vector to i-th subdomain, and matrix W_i applies weights to satisfy the partition of unity. Then we solve on each subdomain a saddle-point problem

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} u_i \\ \lambda \end{bmatrix} = \begin{bmatrix} r_i \\ 0 \end{bmatrix}, \tag{18}$$

where λ are Lagrange multipliers, S_i is the Schur complement with respect to the interface of the i-th subdomain, and 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. After solving this problem on each subdomain, we get the *subdomain correction*.

Let us now have a look at the coarse problem. Before solving it in each iteration, one needs to build it in the set-up phase of the preconditioner. This is performed by solving the saddle-point systems from (18) with several right-hand sides

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{19}$$

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 equal to zero in the remaining local coarse unknowns. As introduced in [12], also a set of *adjoint coarse basis functions* Ψ_i^* is needed for nonsymmetric problems. These are obtained by solving

$$\begin{bmatrix} S_i^T & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i^* \\ \Lambda_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{20}$$

By solving problem (19), we obtain the *local coarse matrix* as a side product,

$$S_{Ci} = \Psi_i^{*T} S_i \Psi_i = -\Lambda_i.$$

Local coarse matrices are then assembled into the global matrix of the coarse problem

$$S_C = \sum_{i=1}^{N} R_{Ci}^T S_{Ci} R_{Ci},$$

where R_{Ci} is the restriction of the global vector of coarse unknowns to those present at i-th subdomain.

In each action of BDDC, we first extract the residuum for the coarse problem as

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

solve the coarse problem

$$S_C u_C = r_C, \tag{21}$$

and distribute the coarse solution to individual subdomains

$$u_{Ci} = \Psi_i R_{Ci} u_C.$$

The complete action of the preconditioner $M_{BDDC} : r^k \to u^k$ is obtained by combining the subdomain corrections with the localised coarse corrections,

$$u^k = \sum_{i=1}^{N} R_i^T W_i (u_i + u_{Ci}).$$

5. Numerical results

As the benchmark problem, we consider the 3-D extension of the popular problem in cavity introduced in [11]. The computational domain is a unit cube. The mesh is divided into 32 subdomains using the METIS library (see Figure 1). The computations are performed by a parallel finite element package written in C++ and described in [5], and the *BDDCML* library [8] is used for solving the arising system of equations. Simulations were performed on an SGI Altix UV 100 supercomputer at the Supercomputer center of the CTU in Prague using 32 cores and the same number of subdomains. Our results are compared with [11]. Two directions of the unit tangential velocity vector are considered on the top wall, $\mathbf{u_{top1}} = (1, 0, 0)$ and $\mathbf{u_{top2}} = (1/\sqrt{3}, \sqrt{2}/\sqrt{3}, 0)$. Picard iteration is used for linearisation, with precision $\left\|u^k - u^{k-1}\right\|_2 \leq 10^{-5}$. In [11], FpGMRES method is used for the linearised systems with a block preconditioner. In our computations, the BiCGstab method preconditioned by the BDDC preconditioner is used. The linear iterations are terminated when $\left\|r^k\right\|_2 / \left\|g\right\|_2 \leq 10^{-6}$ or after reaching the maximum number of 100 iterations.

We compare the maximal numbers of linear iterations over all steps of the nonlinear method. These are considered for two equidistant meshes, with $n = 16$ and 32 elements per edge, corresponding respectively to 4096 and 32 768 elements, 35 937 and 274 625 nodes, and 112 724, 859 812 unknowns. Four different values of viscosity ν are tested. Results are presented in Tables 1 and 2. Number of linear iterations from [11] are denoted as 'FpGMRES + block prec.', while our current results are denoted as 'BiCGstab + BDDC'. Finally, numbers of nonlinear iteration required in our calculations are reported in Table 3.

From Tables 1 and 2, we can see that the number of linear iterations is growing with decreasing viscosity, while the dependence is similar for both methods. This

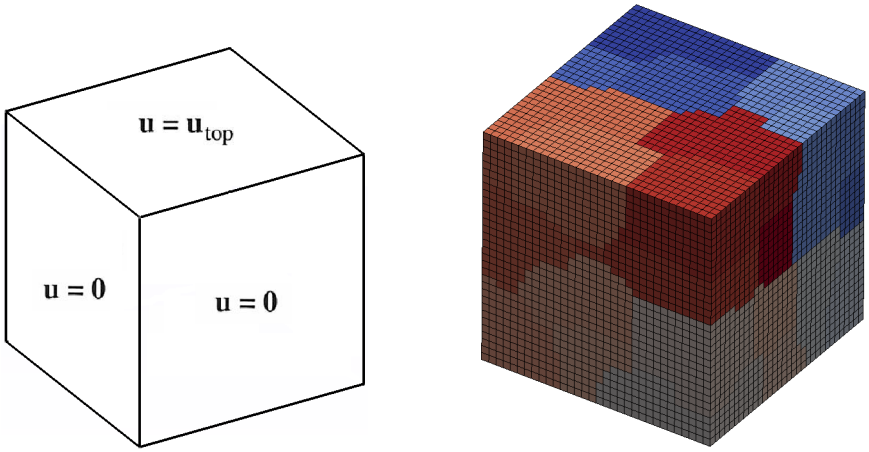

Figure 1: Solution domain with boundary conditions (left) and mesh with 32 subdomains for cavity problem (right)

ν		1/20	1/40	1/80	1/160
$n = 16$	FpGMRES + block prec.	29	32	43	68
	BiCGstab + BDDC	32	31	38	58
$n = 32$	FpGMRES + block prec.	28	32	42	69
	BiCGstab + BDDC	18	19	23	49

Table 1: Number of linear iterations for $\mathbf{u_{top1}} = (1, 0, 0)$

ν		1/20	1/40	1/80	1/160
$n = 16$	FpGMRES + block prec.	29	36	48	64
	BiCGstab + BDDC	29	31	35	53
$n = 32$	FpGMRES + block prec.	28	35	45	61
	BiCGstab + BDDC	18	19	23	49

Table 2: Number of linear iterations for $\mathbf{u_{top2}} = (1/\sqrt{3}, \sqrt{2}/\sqrt{3}, 0)$

confirms that for this problem, the BDDC preconditioner provides a comparable efficiency as the advanced block preconditioner from [11]. As shown in Table 3, a reasonable convergence of the Picard iteration has been obtained for most cases. However, skewing the velocity vector on the lid with respect to coordinate axes had an opposite effect than we expected, with significantly worse convergence for $\mathbf{u_{top1}}$ than for $\mathbf{u_{top2}}$ in the case $\nu = 1/160$. We do not have an explanation for this behaviour. The solution for $\mathbf{u_{top2}}$, $n = 32$, and $\nu = 1/160$ in the slice $x = 0.5$ is shown in Figure 2.

ν		1/20	1/40	1/80	1/160
$\mathbf{u_{top1}}$	$n = 16$	8	11	19	198
	$n = 32$	8	12	21	114
$\mathbf{u_{top2}}$	$n = 16$	8	11	18	39
	$n = 32$	8	12	21	47

Table 3: Number of nonlinear iterations in our calculations

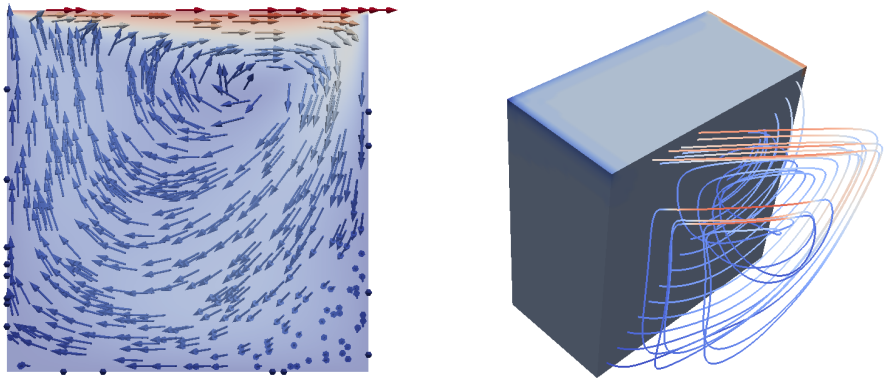

Figure 2: Cavity flow in the plane $x = 0.5$, velocity vectors with magnitude (left) and pressure with several streamtraces (right)

6. Conclusions

In this contribution, we have combined our previous developments on BDDC for the Stokes problem [6], with extensions of the BDDC method to nonsymmetric problems from [12]. An application of the BDDC preconditioner to nonsymmetric linear systems of equations obtained from linearisation of the incompressible Navier-Stokes equations by means of Picard iteration is presented. Taylor-Hood finite elements with continuous approximation of pressure are used for discretisation.

The parallel implementation of the method is employed for solving a 3-D problem of flow in a lid-driven cavity. The required numbers of linear iterations are compared with those by a block preconditioner published in [11], showing a comparable performance of this approach. The BiCGstab method is used for solution of the interface problem, which contains both velocity and pressure unknowns.

Larger tests of parallel scalability and applications to other problems will be the subject of future research.

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 through grant 14-02067S, by the Academy of Sciences of the Czech Republic through RVO:67985840, and by the Czech Technical University in Prague through the student project SGS13/190/OHK2/3T/12.

References

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

[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] Li, J. and Tu, X.: A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures. SIAM Journal on Numerical Analysis **51** (2013), 1235–1253.

[4] Li, J. and Widlund, O. B.: BDDC algorithms for incompressible Stokes equations. SIAM J. Numer. Anal. **44** (2006), 2432–2455.

[5] Šístek, J. and Cirak, F.: Parallel iterative solution of the incompressible Navier-Stokes equations with application to rotating wings, 2014. To be submitted.

[6] Šístek, J., Sousedík, B., Burda, P., Mandel, J., and Novotný, J.: Application of the parallel BDDC preconditioner to the Stokes flow. Comput. Fluids **46** (2011), 429–435.

[7] Šístek, J., Čertíková, M., Burda, P., and Novotný, J.: Face-based selection of corners in 3D substructuring. Math. Comput. Simulat. **82** (2012), 1799–1811.

[8] Sousedík, B., Šístek, J., and Mandel, J.: Adaptive-Multilevel BDDC and its parallel implementation. Computing **95** (2013), 1087–1119.

[9] Tu, X. and Li, J.: A balancing domain decomposition method by constraints for advection-diffusion problems. Commun. Appl. Math. Comput. Sci **3** (2008), 25–60.

[10] van der Vorst, H. A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **13** (1992), 631–644.

[11] Wathen, A. J., Loghin, D., Kay, D. A., Elman, H. C., and Silvester, D. J.: A new preconditioner for the Oseen equations. In: F. Brezzi, A. Buffa, S. Corsaro, and A. Murli (Eds.), *Numerical mathematics and advanced applications*. Springer-Verlag Italia, Milano, 2003 pp. 979–988. Proceedings of ENUMATH 2001, Ischia, Italy.

[12] Yano, M.: *Massively parallel solver for the high-order Galerkin least-squares method*. Master's thesis, Massachusests Institute of Technology, 2009.