Ninth International Conference on Computational Fluid Dynamics (ICCFD9), Istanbul, Turkey, July 11-15, 2016

ICCFD9-0140

Application of the BDDC method to incompressible flows in hydrostatic bearings

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

[1] Faculty of Mechanical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, CZ - 121 35 Prague 2, Czech Republic
[2] Institute of Mathematics, Czech Academy of Sciences, Žitná 25, CZ - 115 67 Prague 1, Czech Republic
Corresponding author: martin.hanek@fs.cvut.cz

Abstract: We apply Balancing Domain Decomposition based on Constraints (BDDC) to stationary incompressible flow governed by the Navier-Stokes equations. This method solves large systems of linear equations arising from the finite element method. The algorithm is applied to nonsymmetric linear systems obtained by Picard's linearisation of the Navier-Stokes equations discretised by Taylor-Hood finite elements. Numerical results for an industrial problem of oil flow in a hydrostatic bearing are presented.

Keywords: BDDC, Navier-Stokes equations, FEM, hydrostatic bearing.

1 Introduction

We deal with numerical simulation of oil flow in hydrostatic bearings. These are parts of production machines that keep heavy moving parts of machines on a thin layer of oil to provide low friction. The thickness of the layer is controlled by the so called *throttling gap* and is only few tens of micrometers high, while the remaining dimensions are in millimeters. The domain where the oil flows, such as the one in Fig. 1, is called a *hydrostatic cell*. Oil enters the domain through the top face and flows out through the outer side of the *throttling gap*, in which the pressure drops from a high value in the main chamber maintained by an oil pump to the atmospheric value outside.

Our main goal is to simulate oil flow in whole *hydrostatic cell* during the move of the bearing wit real scale geometry. This is not an easy task and, up to our knowledge, such type of 3-D simulations has not been presented in literature. During our research we have dealt with several kinds of problems of hydrostatic bearings. Starting with a 2-D axially-symmetric problem which does not allow the sliding motion of the bearing, and ending with a real scale simulation of a moving bearing. This goes with the application of the finite element method to the Navier-Stokes equations and efficient method to solving the arising systems of algebraic equations.

We apply the Balancing Domain Decomposition based on Constraints (BDDC) method to the linear equation systems arising from the discretisation of the Navier-Stokes equations. The BDDC method was first introduced by Dohrmann in [1] and applied to Poisson equation and linear elasticity. We use the approach described by Hanek, Šístek and Burda in [2] combined with the domain partitioning strategy preferring straight subdomain interfaces recently described and investigated in [3].

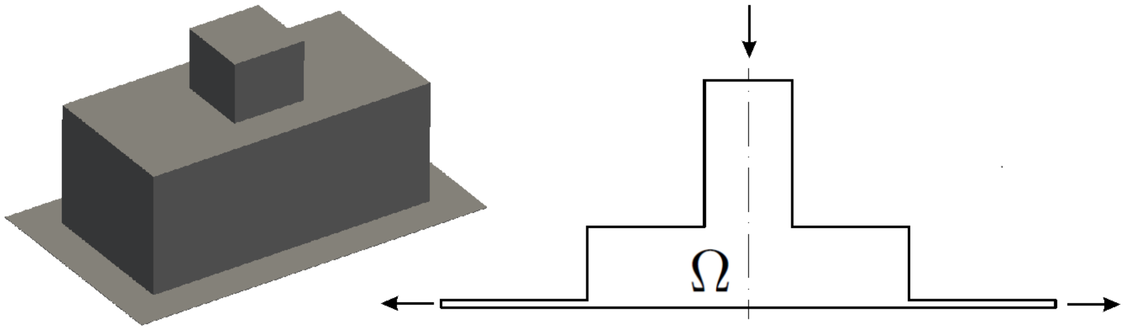

Figure 1: Hydrostatic cell (left) and projection of hydrostatic cell (right)

2 Finite element method for Navier-Stokes equations

In this section we recall our approach from [2] of using one step of BDDC method as a preconditioner for Navier-Stokes equation. We consider stationary flow of incompressible fluid in a 3-D domain governed by the Navier-Stokes equations without body forces (see e.g. [4])

$$
\begin{aligned}
(\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p &= \mathbf{0} \quad \text{in } \Omega, && (1)\\
\nabla\cdot\mathbf{u} &= 0 \quad \text{in } \Omega, && (2)
\end{aligned}
$$

with boundary conditions

$$
\begin{aligned}
\mathbf{u} = \mathbf{g} &\quad \text{on } \Gamma_D, && (3)\\
-\nu(\nabla\mathbf{u})\mathbf{n} + p\mathbf{n} = 0 &\quad \text{on } \Gamma_N, && (4)
\end{aligned}
$$

where $\mathbf{u} = (u_1, u_2, u_3)^T$ is an unknown vector of velocity, p is an unknown pressure normalized by (constant) density, ν is a given kinematic viscosity, Ω is the solution domain, 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 $\mathbf{g}$ is a given function.

After multiplying equations (1)–(2) by test functions, integrating over the solution domain and using the divergence theorem we get the following weak formulation

Seek $\mathbf{u} \in V_g$ *and* $p \in L^2(\Omega)$*, satisfying*

$$
\begin{aligned}
\int_\Omega (\mathbf{u}\cdot\nabla)\mathbf{u}\cdot\mathbf{v}\,\mathrm{d}\Omega + \nu\int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\,\mathrm{d}\Omega - \int_\Omega p\nabla\cdot\mathbf{v}\,\mathrm{d}\Omega = 0 \quad \forall\mathbf{v}\in V, && (5)\\
\int_\Omega q\nabla\cdot\mathbf{u}\,\mathrm{d}\Omega = 0 \quad \forall q \in L^2(\Omega). && (6)
\end{aligned}
$$

Here the spaces are

$$
\begin{aligned}
V_g &:= \left\{\mathbf{u}\in H^1(\Omega)^3, \mathbf{u} = \mathbf{g} \text{ on } \Gamma_D \text{ in the sense of traces}\right\},\\
V &:= \left\{\mathbf{v}\in H^1(\Omega)^3, \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D \text{ in the sense of traces}\right\}.
\end{aligned}
$$

During the assembly of the system of algebraic equations we substitute into the weak formulation for finite element functions of velocity and pressure linear combinations of the basis function to get nonlinear system of algebraic equations. This system is linearized using Picard iteration which leads to the following system

$$
\begin{bmatrix} \nu A + N(\mathbf{u}^k) & B^T \\ B & 0 \end{bmatrix}
\begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{bmatrix}
=
\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \qquad (7)
$$

where $\mathbf{u}^{k+1}$ is the vector of unknown coefficients of velocity at the $(k+1)$-th iteration, $\mathbf{p}^{k+1}$ is the vector of unknown coefficients of pressure at the $(k+1)$-th iteration, A is the matrix of diffusion, $N(\mathbf{u}^k)$ is the matrix of advection where we substitute velocity from the k-th iteration, B is the matrix from continuity equation, and $\mathbf{f}$ and $\mathbf{g}$ are discrete right-hand side vectors arising from Dirichlet boundary conditions. This linear nonsymmetric system is solved using the iterative substructuring.

3 Iterative substructuring for Navier–Stokes equations

In order to use iterative substructuring, we decompose the solution domain Ω into N nonoverlapping subdomains. This gives rise to dividing the unknown coefficients of the finite element functions to two classes – those present only in one subdomain (*interior*) and those shared by two or more subdomains (*interface*).

Let subscript I denote the interior unknowns and subscript Γ denote the interface unknowns. We can now reorder system (7) into the following block system

$$\begin{bmatrix} \nu A_{II}+N_{II} & \nu A_{I\Gamma}+N_{I\Gamma} & B_{II}^T & B_{\Gamma I}^T \\ \nu A_{\Gamma I}+N_{\Gamma I} & \nu A_{\Gamma\Gamma}+N_{\Gamma\Gamma} & B_{I\Gamma}^T & B_{\Gamma\Gamma}^T \\ B_{II} & B_{I\Gamma} & 0 & 0 \\ B_{\Gamma I} & B_{\Gamma\Gamma} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \\ \mathbf{p}_I \\ \mathbf{p}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \\ \mathbf{g}_I \\ \mathbf{g}_\Gamma \end{bmatrix}. \tag{8}$$

After an embarrassingly parallel elimination of the interior unknowns on each subdomain, we get the interface problem

$$S \begin{bmatrix} \mathbf{u}_\Gamma \\ \mathbf{p}_\Gamma \end{bmatrix} = g. \tag{9}$$

Here

$$S = \begin{bmatrix} \nu\mathbf{A}_{\Gamma\Gamma}+\mathbf{N}_{\Gamma\Gamma} & B_{\Gamma\Gamma}^T \\ B_{\Gamma\Gamma} & 0 \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{\Gamma I}+\mathbf{N}_{\Gamma I} & B_{I\Gamma}^T \\ B_{\Gamma I} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{II}+\mathbf{N}_{II} & B_{II}^T \\ B_{II} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nu\mathbf{A}_{I\Gamma}+\mathbf{N}_{I\Gamma} & B_{\Gamma I}^T \\ B_{I\Gamma} & 0 \end{bmatrix}$$

is the Schur complement of the interior unknowns, and

$$g = \begin{bmatrix} \mathbf{f}_\Gamma \\ \mathbf{g}_\Gamma \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{\Gamma I}+\mathbf{N}_{\Gamma I} & B_{I\Gamma}^T \\ B_{\Gamma I} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{II}+\mathbf{N}_{II} & B_{II}^T \\ B_{II} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_I \\ \mathbf{g}_I \end{bmatrix}$$

is the reduced right-hand side. Problem (9) is solved by the BiCGstab method with one step of BDDC as a preconditioner.

4 Balancing Domain Decomposition based on Constraints preconditioner

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

$$r^k = g - S \begin{bmatrix} \mathbf{u}_\Gamma^k \\ \mathbf{p}_\Gamma^k \end{bmatrix}, \tag{10}$$

and provides an approximate solution to problem (9). In each action of the BDDC preconditioner, a coarse problem and independent subdomains problems are solved. First, let us now have a look at the coarse problem. Before applying the preconditioner in each iteration, one needs to set it up. First, we select the so called coarse degrees of freedom. Values of individual components of velocity and pressure are used at corners of subdomains selected according to [5]. In addition, arithmetic averages over edges and faces of subdomains are also considered as coarse unknowns.

The coarse basis functions are found algebraically by solving the following local saddle-point systems on each subdomain

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{11}$$

where 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. The solution Ψ_i is the matrix of *coarse basis functions* with every column corresponding to one coarse unknown on the subdomain. These functions resemble finite element shape functions – they are equal to one in the corresponding coarse degree of freedom, and they equal to zero in the remaining local coarse unknowns.

As was show by Yano in [6], due to the nonsymmetry of the problem we also need to solve another saddle-point system

$$\begin{bmatrix} S_i^T & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i^* \\ \Lambda_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{12}$$

where the solution Ψ_i^* is the matrix of *adjoint coarse basis functions* with similar properties as the coarse basis functions.

Denoting the residual preconditioned by the BDDC preconditioner $u_\Gamma^k = M_{BDDC}^{-1}\, r^k$, an action of the BDDC preconditioner can be described by the following scheme

$$r_i^k = W_i R_i r^k$$

coarse problem

$$S_C = \sum_{i=1}^{N} R_{Ci}^T \Psi_i^{*T} S_i \Psi_i R_{Ci}$$

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

$$S_C u_C = r_C^k$$

$$u_{Ci} = \Psi_i R_{Ci}^k u_C$$

subdomain problems

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

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

Here R_i is an operator restricting a global interface vector to i-th subdomain, matrix W_i applies weights to satisfy the partition of unity and R_{Ci} is the restriction of the global vector of coarse unknowns to those present at the i-th subdomain. The weights in the diagonal matrix W_i are constructed simply as inverses of the number of subdomains at the interface degrees of freedom.

5 Numerical results

In this section we briefly recall our results from [3], where the importance of keeping straight interfaces of subdomains and low aspect ratios of element faces on the interface was shown. The graph and geometric partitioning strategies from [3] are applied here to our driving application – simulations of oil in hydrostatic bearings. In [3] we investigated the effect of aspect ratio of the faces of finite elements at the interface on convergence. As a benchmark problem, we considered a channel narrowing along one or two coordinates to increase the aspect ratio of the finite elements. Detail of the interface between two subdomains for each partitioner can be seen in Fig. 2. The results show that with increasing aspect ratio of finite elements, the number of iterations for the graph partitioner rapidly increases while for the partitioner favouring straight interfaces, it remains the same, and robustness with respect to the element aspect ratios can be achieved. More details can be found in [3].

The computations have been performed by a parallel finite element package written in C++ and described by [7], with the *BDDCML* library being used for solving the arising systems of linear equations. The Picard iteration is terminated when $\left\|u^k - u^{k-1}\right\|_2 \leq 10^{-5}$ or after performing 100 iterations. Here $u^k = \begin{bmatrix} \mathbf{u}_\Gamma^k \\ \mathbf{p}_\Gamma^k \end{bmatrix}$. The BiCGstab method

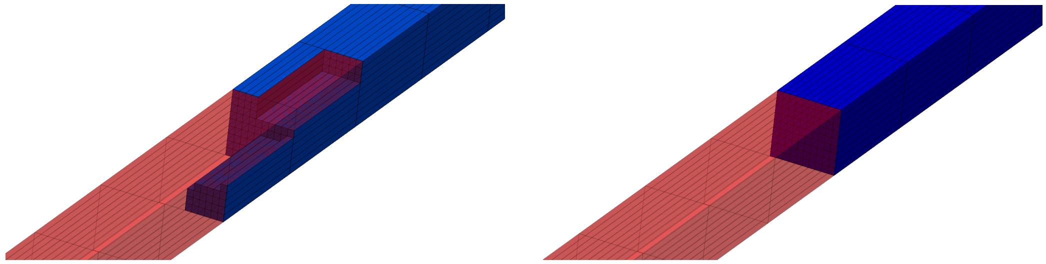

Figure 2: Detail of the interface between two subdomains for the graph (left) and the geometric (right) partitioner (from [3]).

is stopped if $\left\|r^k\right\|_2 / \|g\|_2 \leq 10^{-6}$, with the limit of 1000 iterations.

In our calculation we consider that the bearing is sliding on the lower wall with velocity $\mathbf{u} = (1,0,0)$ and kinematic viscosity is set to $\nu = 0.1$. In order to resolve the flow in the *throttling gap*, several layers of elements are placed along its thickness. This results in very bad aspect ratio of elements in the gap, easily reaching 200. The mesh contains approximately 18 thousand elements, 159 thousand nodes, and 500 thousand unknowns. Computations are performed on 32 processors.

We compare the two approaches to mesh partitioning described in [3]. Solution of the problem is presented in Fig. 4. Picard iteration using the mesh by the graph partitioner did not converge in 100 iterations. On the other hand, solution of the problem decomposed by the geometric partitioner required only 3 Picard iterations, each solved in the average by 162 BiCGstab iterations.

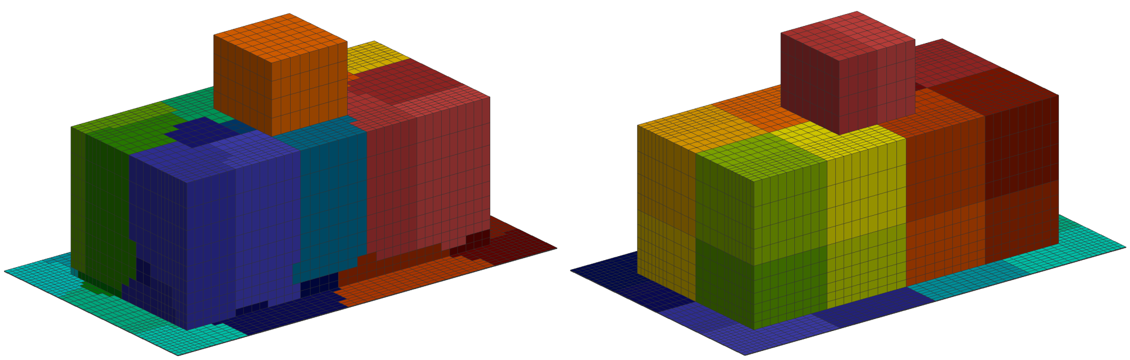

Figure 3: Computational mesh partitioned into 32 subdomains by the graph partitioner (left) and by the geometric partitioner (right).

6 Conclusions

We have applied the Balancing Domain Decomposition based on Constraints (BDDC) method to Navier-Stokes equations. The goal is to simulate an industrial problem of oil flow in hydrostatic bearings. We have combined approach of using BDDC preconditioner to nonsymmetric linear equation systems from [2] with partitioning strategies described in [3]. In [3] we have shown that domain decomposition using the graph partitioner could be problematic for meshes containing elements with high aspect ratios, while the geometric partitioner can drastically improve robustness for those meshes.

In this contribution we compare computational possibilities using partitioners from [3] in application to a complicated geometry of hydrostatic bearing. Using the graph partitioner to decompose the solution domain, we could not

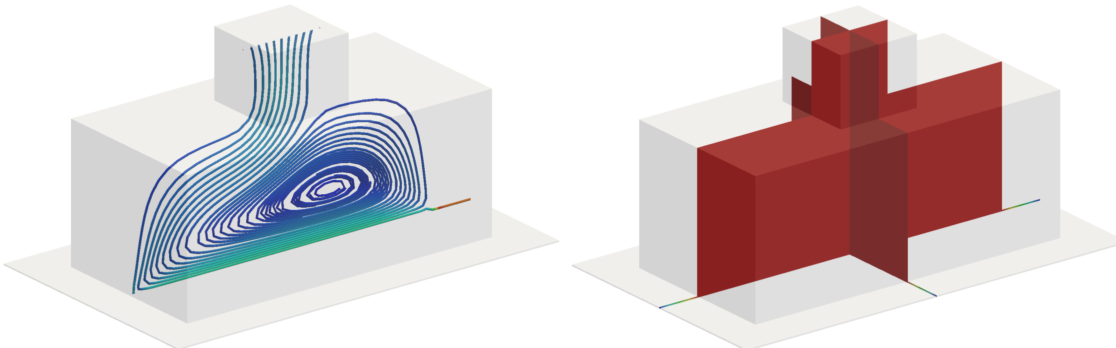

Figure 4: Solution of the hydrostatic bearing problem. Streamlines coloured by the magnitude of velocity (left) and a plot of pressure (right).

achieve the required precision for linear problems. On the other hand, using the geometric partitioner has given us the possibility to simulate flow of oil in hydrostatic bearings with real scale geometries of the throttling gap.

Acknowledgements

This work has been supported by the Grant Agency of the Czech Technical University in Prague through the grant No. SGS16/220/OHK2/3T/12, by the Czech Science Foundation through grant 14-02067S, and by the Czech Academy of Sciences through RVO:67985840.

References

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

[2] M. Hanek, J. Šístek, and P. Burda. An application of the BDDC method to the Navier–Stokes equations in 3–D cavity. In J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, and T. Vejchodský, editors, *Programs and algorithms of numerical mathematics 17*, pages 77–85. Institute of Mathematics AS CR, 2015.

[3] M. Hanek, J. Šístek, and P. Burda. The effect of irregular interfaces on the BDDC method for the Navier-Stokes equations. In *Proceedings of the International Conference on Domain Decomposition Methods (DD) XXIII, Jeju Island, Korea, July 6–10, 2015*. To appear.

[4] D. J. Silvester H. C. Elman and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005.

[5] J. Šístek, M. Čertíková, P. Burda, and J. Novotný. Face-based selection of corners in 3D substructuring. *Math. Comput. Simulat.*, 82(10):1799–1811, 2012.

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

[7] J. Šístek and F. Cirak. Parallel iterative solution of the incompressible Navier-Stokes equations with application to rotating wings. *Comput. & Fluids*, 122:165–183, 2015.