Efficient Solvers for Large-Scale Saddle Point Systems Arising in Feedback Stabilization of Multi-field Flow Problems

Peter Benner(✉), Jens Saak, Martin Stoll, and Heiko K. Weichelt

Max Planck Institute for Dynamics of Complex Technical Systems,
Sandtorstr. 1, 39106 Magdeburg, Germany
{benner,saak,stoll,weichelt}@mpi-magdeburg.mpg.de
http://www.mpi-magdeburg.mpg.de

Abstract. This article introduces a block preconditioner to solve large-scale block structured saddle point systems using a Krylov-based method. Such saddle point systems arise, e.g., in the Riccati-based feedback stabilization approach for multi-field flow problems as discussed in [2]. Combining well known approximation methods like a least-squares commutator approach for the Navier-Stokes Schur complement, an algebraic multigrid method, and a Chebyshev-Semi-Iteration, an efficient preconditioner is derived and tested for different parameter sets by using a simplified reactor model that describes the spread concentration of a reactive species forced by an incompressible velocity field.

Keywords: Coupled flow control · Large-scale saddle point systems · Preconditioned GMRES · Least-squares commutator approach · Algebraic multigrid · Chebyshev-Semi-Iteration

1 Introduction

In this paper we investigate the solution of large-scale saddle point systems arising in control problems for coupled partial differential equations (PDEs). The starting points are recent publications concerning the boundary feedback stabilization of non-coupled flows like the linear Stokes flow in [3] and the non-linear Navier-Stokes flow in [1]. The analytic approach for this feedback stabilization is given by Raymond in, e.g., [13].

Using the projection idea proposed by Heinkenschloss et al. [8], Benner et al. [1,3] show that the solution of certain saddle point systems is the key ingredient to ensure that the numerical solution lies on the correct solution manifold, i.e., the space of discretely divergence-free velocity fields, without performing an explicit projection.

Applying these ideas to a coupled flow problem, namely the Navier-Stokes equations combined with a diffusion-convection equation, leads to saddle point systems with a more complicated block structure [2]. Solving these systems efficiently requires the use of appropriate preconditioners. This paper investigates

C. Pötzsche et al. (Eds.): CSMO 2013, IFIP AICT 443, pp. 11–20, 2014.
DOI: 10.1007/978-3-662-45504-3_2

an efficient iterative solution strategy via the use of preconditioned Krylov subspace methods based on the framework derived in [3]. Here we consider the full feedback system for the coupled multi-field flow problem, while in [3], only the linear Stokes case was treated without coupling to another field equation. Moreover, this paper complements [2] in the sense that there, we have focused on presenting results on the convergence of the Newton-ADI method for solving the algebraic Riccati equation determining the stabilizing feedback control for the coupled system, where the saddle point problems in the innermost step of the Newton-ADI iteration were solved by sparse direct methods, while here, we study preconditioned iterative solvers for this step.

The remainder of this paper is organized as follows. Section 2 briefly recalls the feedback stabilization approach for multi-field flow problems from [2] that leads to large-scale saddle point systems. Afterwards, we discuss properties of these saddle point systems to derive a suitable preconditioner in Sect. 3. Section 4 shows numerical results before we conclude the paper and give a short outlook to further investigations in Sect. 5.

2 Derivation of Saddle Point Systems

The derivation of the block structured saddle point systems in [2] starts with the linearized coupled flow problem defined for $t \in [0, \infty)$ and $\boldsymbol{x} \in \Omega \subset \mathbb{R}^2$. The linearized Navier-Stokes equations that describe, up to first order, the difference between actual and desired velocity and pressure are given as

$$\left.\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{z} - \frac{1}{\mathrm{Re}}\Delta\boldsymbol{z} + (\boldsymbol{w}\cdot\nabla)\boldsymbol{z} + (\boldsymbol{z}\cdot\nabla)\boldsymbol{w} + \nabla p &= \boldsymbol{f}_l, \\ \operatorname{div}\boldsymbol{z} &= 0, \end{aligned}\right. \quad \text{on } [0,\infty)\times\Omega. \tag{1}$$

They are then coupled via the velocity field $\boldsymbol{z}(t, \boldsymbol{x})$ with the linearized diffusion-convection equation

$$\frac{\partial}{\partial t}c_{\boldsymbol{z}} - \frac{1}{\mathrm{ReSc}}\Delta c_{\boldsymbol{z}} + (\boldsymbol{w}\cdot\nabla)c_{\boldsymbol{z}} + (\boldsymbol{z}\cdot\nabla)c_{\boldsymbol{w}} = 0, \quad \text{on } [0,\infty)\times\Omega \tag{2}$$

that describes the concentration of a reactive species denoted by $c_{\mathbf{z}}(t, \boldsymbol{x})$. The stationary linearization points $\boldsymbol{w}(\boldsymbol{x})$ for the velocity and $c_{\boldsymbol{w}}(\boldsymbol{x})$ for the concentration are assumed to be given. The equations are scaled with the Reynolds number Re and the Schmidt number Sc. Using the mixed Taylor-Hood finite elements [9] for the velocity and pressure in Eq. (1) as well as linear ansatz functions for the concentration in Eq. (2), we end up with a system of discrete differential-algebraic equations (DAE) that can be written as the control system:

$$\begin{bmatrix} M_{\mathbf{z}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_{\mathbf{c}} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} A_{\mathbf{z}} & G & 0 \\ G^T & 0 & 0 \\ -R & 0 & A_{\mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{c} \end{bmatrix} + \begin{bmatrix} B_{\mathbf{z}} \\ 0 \\ 0 \end{bmatrix} \mathbf{u}, \tag{3a}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & C_{\mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{c} \end{bmatrix} \tag{3b}$$

with the first block row for velocity (of dimension $n_{\mathbf{z}}$), the second row for pressure (of dimension $n_{\mathbf{p}}$), and the third row for concentration (of dimension $n_{\mathbf{c}}$) [2].

The matrix pencil

$$\left(\underbrace{\begin{bmatrix} A_{\mathbf{z}} & G & 0 \\ G^T & 0 & 0 \\ -R & 0 & A_{\mathbf{c}} \end{bmatrix}}_{\mathbf{A}}, \underbrace{\begin{bmatrix} M_{\mathbf{z}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_{\mathbf{c}} \end{bmatrix}}_{\mathbf{M}} \right)$$

is of dimension $n \times n$ with $n = n_{\mathbf{z}} + n_{\mathbf{c}} + n_{\mathbf{p}}$ and has $2n_{\mathbf{p}}$ infinite eigenvalues [5].

In [2], a linear-quadratic regulator (LQR) approach is applied to system (3) for determining the stabilizing control function $\mathbf{u}$. The solution of this LQR problem is a linear feedback control $\mathbf{u}(t) = \mathbf{K}(\mathbf{z}(t), \mathbf{p}(t), \mathbf{c}(t))$, determined via the solution of an algebraic Riccati equation (ARE) defined on the subspace of discretely divergence-free vector fields. The resulting ARE is then solved using a Newton-ADI algorithm. This method yields a threefold nested iteration. In the innermost loop, saddle point systems of the form

$$\underbrace{\begin{bmatrix} A_{\mathbf{z}}^T + q_i M_{\mathbf{z}} & G & -R^T \\ G^T & 0 & 0 \\ 0 & 0 & A_{\mathbf{c}}^T + q_i M_{\mathbf{c}} \end{bmatrix}}_{=\mathbf{A}^T + q_i \mathbf{M} =: \mathbf{F}_i} \underbrace{\begin{bmatrix} \Lambda_{\mathbf{z}} \\ \Lambda_{\mathbf{p}} \\ \Lambda_{\mathbf{c}} \end{bmatrix}}_{\mathbf{\Lambda}} = \underbrace{\begin{bmatrix} \tilde{Y}_{\mathbf{z}} \\ 0 \\ \tilde{Y}_{\mathbf{c}} \end{bmatrix}}_{\mathbf{Y}}. \tag{4}$$

have to be solved for certain ADI shifts $q_i \in \mathbb{C}^-$ and a block right hand side $\mathbf{Y}$. The whole nested iteration is given in [2, Algorithm 1] and is omitted here due to space constraints.

3 Preconditioned Iterative Solvers for Block Structured Saddle Point Systems

The use of direct solvers in (4) is only suitable for moderate problem sizes and two-dimensional problems. Although iterative methods can handle much larger systems, their performance will deteriorate if the mesh-size decreases. To avoid this, a suitable preconditioner $\mathbf{P}_i \in \mathbb{C}^{n \times n}$ is introduced such that

$$\mathbf{P}_i^{-1} \mathbf{F}_i \mathbf{\Lambda} = \mathbf{P}_i^{-1} \mathbf{Y}$$

is solved instead of (4) (see [7,16]). Before we derive a suitable preconditioner $\mathbf{P}_i$ we need to describe the properties of the saddle point system and their influence on the chosen preconditioner.

3.1 Properties

The matrices $M_{\mathbf{z}}, M_{\mathbf{c}}$ are symmetric and positive definite, G, R are of full rank, and the ADI shift $q_i \in \mathbb{C}^-$ is contained in the convex hull of the finite spectrum of $(\mathbf{A}, \mathbf{M})$. The shifted system matrix $\mathbf{F}_i$ is indefinite $\forall q_i \in \mathbb{C}^-$. Due to the different q_i, the matrix $\mathbf{F}_i$ changes in each ADI step and, therefore, the preconditioner has to be adapted in each ADI step as well. Nevertheless, for the remainder of this section we assume a fixed ADI shift $q_i = q$ to omit the index i if it is obvious.

3.2 Derivation of Block Preconditioner

Adapting the ideas from [3, Sect. 3.2] we consider

$$\mathbf{F} = \begin{bmatrix} F_{\mathbf{z}} & G & -R^T \\ G^T & 0 & 0 \\ 0 & 0 & F_{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{NSE} & -\tilde{R}^T \\ 0 & F_{\mathbf{c}} \end{bmatrix} \quad \text{with} \quad \begin{array}{r} F_{\mathbf{z}} := A_{\mathbf{z}}^T + qM_{\mathbf{z}}, \\ F_{\mathbf{c}} := A_{\mathbf{c}}^T + qM_{\mathbf{c}}, \\ \tilde{R} := \begin{bmatrix} R \; 0 \end{bmatrix}, \end{array} \tag{5}$$

and $\mathbf{F}_{NSE}$ as the saddle point system for the non-coupled Navier-Stokes flow as it is used in [1]. Using the preconditioner $\mathbf{P}_{NSE}$ from [3], we define a block preconditioner for the use with GMRES [17] to solve with the block structured saddle point system (5) as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{NSE} & -\tilde{R}^T \\ 0 & P_{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} P_{\mathbf{z}} & 0 & -R^T \\ G^T & -P_{SC} & 0 \\ 0 & 0 & P_{\mathbf{c}} \end{bmatrix}$$

$$\Rightarrow \mathbf{P}^{-1} = \begin{bmatrix} P_{\mathbf{z}}^{-1} & 0 & P_{\mathbf{z}}^{-1} R^T P_{\mathbf{c}}^{-1} \\ P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} & -P_{SC}^{-1} & P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} R^T P_{\mathbf{c}}^{-1} \\ 0 & 0 & P_{\mathbf{c}}^{-1} \end{bmatrix}.$$

In contrast to the preconditioner derived in [3], we cannot achieve a block lower triangular matrix due to the coupling matrix R. Applying $\mathbf{P}^{-1}$ to $\mathbf{F}$ yields

$$\mathbf{P}^{-1}\mathbf{F} = \begin{bmatrix} P_{\mathbf{z}}^{-1} F_{\mathbf{z}} & P_{\mathbf{z}}^{-1} G & -P_{\mathbf{z}}^{-1} R^T + P_{\mathbf{z}}^{-1} R^T P_{\mathbf{c}}^{-1} F_{\mathbf{c}} \\ P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} F_{\mathbf{z}} - P_{SC}^{-1} G^T & P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} G & -P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} R^T + P_{SC}^{-1} G^T P_{\mathbf{z}}^{-1} R^T P_{\mathbf{c}}^{-1} F_{\mathbf{c}} \\ 0 & 0 & P_{\mathbf{c}}^{-1} F_{\mathbf{c}} \end{bmatrix} \tag{6}$$

If one assumes $P_{\mathbf{z}} = F_{\mathbf{z}}, P_{\mathbf{c}} = F_{\mathbf{c}}$, and $P_{SC} = G^T F_{\mathbf{z}}^{-1} G$ as ideal approximations in (6), this leads to

$$\begin{bmatrix} I_{\mathbf{z}} & F_{\mathbf{z}}^{-1} G & -F_{\mathbf{z}}^{-1} R^T + F_{\mathbf{z}}^{-1} R^T \\ P_{SC}^{-1} G^T - P_{SC}^{-1} G^T & P_{SC}^{-1} G^T F_{\mathbf{z}}^{-1} G & -P_{SC}^{-1} G^T F_{\mathbf{z}}^{-1} R^T + P_{SC}^{-1} G^T F_{\mathbf{z}}^{-1} R^T \\ 0 & 0 & I_{\mathbf{c}} \end{bmatrix}$$

$$= \begin{bmatrix} I_{\mathbf{z}} & * & 0 \\ 0 & I_{\mathbf{p}} & 0 \\ 0 & 0 & I_{\mathbf{c}} \end{bmatrix}$$

and our iterative method would converge within one step. The goal is to find good approximations for $P_{\mathbf{z}}, P_{\mathbf{c}}$, and P_{SC} that can be evaluated fast and still cluster the eigenvalues in a suitable way such that our iterative solver shows fast convergence [7]. Instead of calculating the inverse $\mathbf{P}^{-1}$ to apply the preconditioner $\mathbf{P}$, we consider the solution of a linear system

$$\begin{bmatrix} P_{\mathbf{z}} & 0 & -R^T \\ G^T & -P_{SC} & 0 \\ 0 & 0 & P_{\mathbf{c}} \end{bmatrix} \begin{bmatrix} x_{\mathbf{z}} \\ x_{\mathbf{p}} \\ x_{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} b_{\mathbf{z}} \\ b_{\mathbf{p}} \\ b_{\mathbf{c}} \end{bmatrix} \tag{7}$$

that can be solved in three steps:

$$\text{Step I:} \qquad x_{\mathbf{c}} = P_{\mathbf{c}}^{-1} b_{\mathbf{c}}, \tag{8a}$$

$$\text{Step II:} \qquad x_{\mathbf{z}} = P_{\mathbf{z}}^{-1}(R^T x_{\mathbf{c}} + b_{\mathbf{z}}), \tag{8b}$$

$$\text{Step III:} \qquad x_{\mathbf{p}} = P_{SC}^{-1}(G^T x_{\mathbf{z}} - b_{\mathbf{p}}). \tag{8c}$$

In conclusion, the coupling matrix R only leads to a matrix-vector multiplication. In steps I and II, one needs to solve with the shifted velocity and concentration system matrices as defined in (5). For both steps, an algebraic multigrid (AMG) method can be used as it is described below. But first, we discuss the more challenging step III that is handled as follows.

3.3 Approximation Methods

Schur Complement Approximation. P_{SC} is an approximation of the Navier-Stokes *Schur complement* $SC := G^T F_{\mathbf{z}}^{-1} G \in \mathbb{R}^{n_{\mathbf{p}} \times n_{\mathbf{p}}}$. Unfortunately, the matrix SC would be a dense matrix that includes the inverse of $F_{\mathbf{z}}$. To avoid the use of this matrix, we follow the approach in [3,18] and use a slightly modified variant of the least squares commutator approach as it is described in [7, Sect. 8.2]. Namely, we consider the shifted Oseen operator in the velocity space

$$\mathcal{F}_{\boldsymbol{z}} = -\frac{1}{\mathrm{Re}}\nabla^2 + \boldsymbol{w} \cdot \nabla + q\mathcal{I}.$$

Note that it is common practice to omit the reaction term $(\boldsymbol{z} \cdot \nabla)\boldsymbol{w}$ that appears in the linearized Navier-Stokes equations to derive preconditioners [7, Sect. 8]. Similar to [7, Sect. 8.2] and [6], we suppose that there exists an analogous operator on the pressure space defined as

$$\mathcal{F}_p = (-\frac{1}{\mathrm{Re}}\nabla^2 + \boldsymbol{w} \cdot \nabla + q\mathcal{I})_p.$$

The least squares commutator of the shifted Oseen operator with the gradient operator is defined as

$$\mathcal{E} = (\mathcal{F})\nabla - \nabla(\mathcal{F}_p)$$

and is supposed to become small in some sense [7]. Using the discrete versions of the operators, we end up with

$$E = (M_{\mathbf{z}}^{-1} F_{\mathbf{z}}) M_{\mathbf{z}}^{-1} G - M_{\mathbf{z}}^{-1} G (M_{\mathbf{p}}^{-1} F_{\mathbf{p}})$$

with $M_{\mathbf{p}}$ the mass matrix and $F_{\mathbf{p}} = A_{\mathbf{p}}^T + qM_{\mathbf{p}}$ the shifted system matrix, both defined on the pressure space. Premultiplying this by $G^T F_{\mathbf{z}}^{-1} M_{\mathbf{z}}$ and postmultiplying by $F_{\mathbf{p}}^{-1} M_{\mathbf{p}}$ yields [3]

$$G^T M_{\mathbf{z}}^{-1} G F_{\mathbf{p}}^{-1} M_{\mathbf{p}} \approx G^T F_{\mathbf{z}}^{-1} G = SC.$$

The large and dense matrix $G^T M_{\mathbf{z}}^{-1} G$ cannot be used explicitly, but it is shown in [7, Sect. 5.5.1] that this matrix is spectrally equivalent to the Laplacian $S_{\mathbf{p}}$ defined on the pressure space for an inf-sup stable discretization and an inflow-outflow problem [7, Sect. 8.2], as it is considered in this paper. Finally, we obtain

$$P_{SC} \approx S_{\mathbf{p}} F_{\mathbf{p}}^{-1} M_{\mathbf{p}} \quad \Rightarrow \quad P_{SC}^{-1} \approx M_{\mathbf{p}}^{-1} F_{\mathbf{p}} S_{\mathbf{p}}^{-1}.$$

In [4] the authors use a similar approach for the Navier-Stokes equations. In summary, the application of P_{SC}^{-1} requires to solve with $S_{\mathbf{p}}$ (step IIIa), multiply with $F_{\mathbf{p}}$ (step IIIb), and solve with $M_{\mathbf{p}}$ (step IIIc). The step IIIa can be solved with an AMG method, similar to the steps I and II.

Algebraic Multigrid. As it is described above, the steps I (8a), II (8b) and IIIa are solved using an AMG method [14]. Due to the possibly complex ADI shifts q in (8a) and (8b), we use the AGMG package developed by the group of Y. Notay [10–12]. In all three cases we use the MATLAB®-based implementation to solve systems of the form

$$Fx = b$$

with a sparse matrix $F \in \{F_{\mathbf{z}}, F_{\mathbf{c}}, S_{\mathbf{p}}\}$. Details about the used parameters for the function `agmg` are discussed in Subsect. 4.2. For more details about the internally used methods and the implemented syntax we refer the reader to [11]. Although the AGMG method can handle complex arithmetic, it needs significantly more steps to converge to the desired tolerance. Additionally, we note that `agmg` is a non-linear function, such that one should use a flexible iterative method, e.g., FGMRES [15]. However, our numerical experiments do not show any drawbacks using a standard GMRES implementation.

Chebyshev-Semi-Iteration. Although the solution of step IIIc with the symmetric positive definite mass matrix $M_{\mathbf{p}}$ is relatively cheap, this can still be accelerated by using the *Chebyshev-Semi-Iteration* as it is described, e.g., in [18]. Numerical tests showed that one needs only 4−6 steps to obtain a suitable result for the preconditioner, which results in a speedup that is shown in Subsect. 4.2.

The next section depicts selected results to show the performance of the preconditioned iterative method.

4 Numerical Examples

To test the efficiency of the preconditioned iterative method, the same data and configurations as in [2] are used. After refining the initial triangulation of the reactor model in Fig. 1, we end up with the variable dimensions as depicted in Table 1b. Furthermore, we define five parameter sets for different combinations of Reynolds and Schmidt numbers as shown in Table 1a. We use the MATLAB implementation of GMRES [17] to solve the saddle point systems (4) for selected ADI shifts q_i that appear during the Newton-ADI iteration. Each q_i is used for

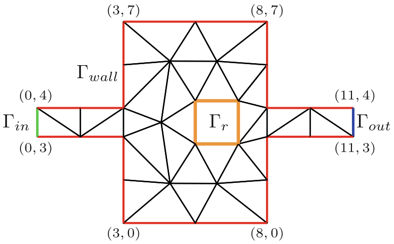

Fig. 1. Initial triangulation of the reactor model with coordinates and boundary conditions [2].

Table 1. Test parameter settings.

Set	Re	Sc
I	1	1
II	1	10
III	10	1
IV	1	100
V	10	10

(a) Different parameter settings.

Variable	Dimension
$n_{\mathbf{z}}$	9 092
$n_{\mathbf{p}}$	1 276
$n_{\mathbf{c}}$	1 187
n	11 555

(b) Different dimensions of FE space.

three ADI steps with four right hand sides every time. Thereby, the number of GMRES steps and the CPU times are measured and arithmetically averaged. The preconditioner $\mathbf{P}$ is evaluated as a MATLAB function handle that solves the linear system (7) using the steps (8). The GMRES tolerance is set to 10^{-10} to ensure the same convergence of the ADI iteration that a direct solve would imply [3]. Although a few complex ADI-shifts q_i appear for each parameter set during the Newton-ADI process, the pictures only show the real parts of q_i.

All computations were executed in MATLAB R2012a on a 64-bit server with 2×Intel® Xeon® X5650 @2.67 GHz, 12 Cores (6 Cores per CPU) and 48 GB main memory available.

4.1 Influence of ADI Shifts and Reynolds and Schmidt Numbers

The influence of the variation of the Reynolds and Schmidt numbers as given in Table 1 is depicted in Fig. 2. To obtain the best approximations for the preconditioning steps (8a)–(8c), a direct solver is used to solve with $F_{\mathbf{z}}, F_{\mathbf{c}}$, and $S_{\mathbf{p}}$. It can be observed that for ADI shifts $-10^5 < \mathrm{Re}\,(q_i) < -10^1$, between 20–25 GMRES steps are needed. As soon as the absolute value of q_i gets smaller then 10 the number of steps increases. This is a natural behavior, because the influence of the mass matrices $M_{\mathbf{z}}$ and $M_{\mathbf{p}}$ vanishes. Nevertheless, GMRES converges within at most 40–80 steps for all parameter configurations. An empirical test to set: $q_i = -10 \quad \forall |q_i| < 10$, during the Newton-ADI process showed similar ADI convergence behavior as for the original shift selection, without the drawback of

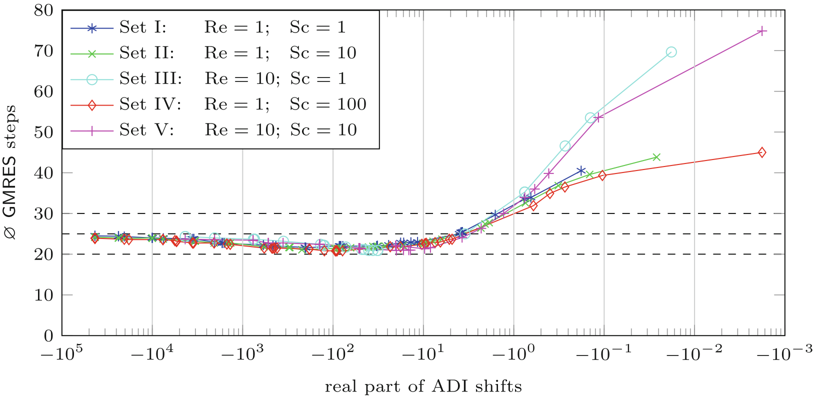

Fig. 2. Average number of GMRES steps for a representative selection of ADI shifts from the Newton-ADI iteration for the configuration sets in Table 1a.

higher GMRES cost for certain shifts. In summary, the derived preconditioner is suitable concerning different Reynolds and Schmidt numbers, as well as different ADI shifts.

4.2 Approximations Using AMG and Chebyshev-Semi-Iteration

As described in Subsect. 3.3, the different preconditioning steps should be solved by an easy to evaluate approximation that is accurate enough to ensure the convergence of GMRES, but avoids the use of sparse factorizations of large-scale sparse matrices. We exchanged the direct solver by its approximation step by step and depict the results in Fig. 3. At first, we use the MATLAB based function agmg [11] to solve with $F_{\mathbf{z}}$ and $F_{\mathbf{c}}$ in (8b) and (8a) with an accuracy of 10^{-10}. Depending on the used ADI shift, the function agmg needed 1–30 steps. Thus, the times to solve the whole saddle point system with the same number of GMRES steps increased a little bit compared to the direct solver. At second, we approximately solved with $S_{\mathbf{p}}$ in step IIIa using agmg as a preconditioner. This was sufficient enough to achieve the GMRES accuracy and, furthermore, decreased the time. Finally, we applied a Chebyshev-Semi-Iteration to approximately solve with $M_{\mathbf{p}}$ in step IIIc. The obtained speedup finally decreased the times below the time used by the direct solver in each step without the loss of any accuracy in GMRES. Due to the above addressed problems with complex ADI shifts in agmg, we restrict our comparison in Fig. 3 to a selection of real ADI shifts. The selection has been performed such that the span of all ADI shifts appearing in the entire Newton-ADI process is covered. Where those shifts clustered we only chose one representative per cluster.

At the end of this section it should be noted that the suggested preconditioned GMRES method for the considered class of saddle point problems would show its full strength in comparison to a direct solver when using finer discretizations,

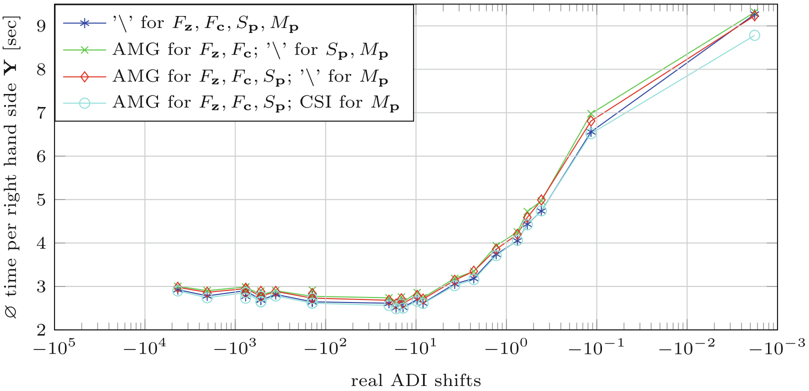

Fig. 3. Average time to solve Eq. (4) with GMRES for a representative selection of real ADI shifts from the Newton-ADI iteration for different approximations of the preconditioning steps (8).

leading to larger dimensions, and in particular when moving to 3D problems. This will be addressed in future work.

5 Conclusions and Outlook

We have recalled the formation of block structured saddle point systems as they arise within the Riccati-based feedback stabilization approach for coupled flow problems that avoids any explicit projection [2]. We were able to extend the results from [3], developed for uncoupled Stokes flow, to the coupled flow described by incompressible Navier-Stokes and a diffusion-convection equation. For that reason, the least-squares commutator approach in [7] has been modified to approximate the shifted Navier-Stokes Schur complement. Exploiting the block structure of the arising preconditioner guarantees a fast evaluation within GMRESĖach of the blocks can either be approximated by an AMG method or a Chebyshev-Semi-Iteration. Several numerical experiments showed that the derived preconditioning method is able to solve the arising saddle point systems efficiently independent of the different parameter settings. Only the use of complex ADI shifts during the Newton-ADI process is not yet optimally covered by this approach and will be investigated in the future.

References

1. Bänsch, E., Benner, P., Saak, J., Weichelt, H.K.: Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flow. Preprint SPP1253-154, DFG-SPP1253 (2013)
2. Bänsch, E., Benner, P., Saak, J., Weichelt, H.K.: Optimal control-based feedback stabilization of multi-field flow problems. In: Leugering, G., Benner, P., Engell, S., Griewank, A., Harbrecht, H., Hinze, M., Rannacher, R., Ulbrich, S. (eds.) Trends in PDE Constrained Optimization. International Series of Numerical Mathematics. Birkhäuser (2014, to appear)

3. Benner, P., Saak, J., Stoll, M., Weichelt, H.K.: Efficient solution of large-scale saddle point systems arising in Riccati-based boundary feedback stabilization of incompressible Stokes flow. SIAM J. Sci. Comput. **35**(5), S150–S170 (2013)
4. Benzi, M., Olshanskii, M.A., Wang, Z.: Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. Int. J. Numer. Meth. Fluids **66**(4), 486–508 (2011)
5. Cliffe, K.A., Garratt, T.J., Spence, A.: Eigenvalues of block matrices arising from problems in fluid mechanics. SIAM J. Matrix Anal. Appl. **15**(4), 1310–1318 (1994)
6. Elman, H., Howle, V., Shadid, J., Shuttleworth, R., Tuminaro, R.: Block preconditioners based on approximate commutators. SIAM J. Sci. Comput. **27**(5), 1651–1668 (2006)
7. Elman, H., Silvester, D., Wathen, A.: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. Oxford University Press, Oxford (2005)
8. Heinkenschloss, M., Sorensen, D.C., Sun, K.: Balanced truncation model reduction for a class of descriptor systems with application to the Oseen equations. SIAM J. Sci. Comput. **30**(2), 1038–1063 (2008)
9. Hood, P., Taylor, C.: Navier-Stokes equations using mixed interpolation. In: Oden, J.T., Gallagher, R.H., Taylor, C., Zienkiewicz, O.C. (eds.) Finite Element Methods in Flow Problems, pp. 121–132. University of Alabama in Huntsville Press, Huntsville (1974)
10. Napov, A., Notay, Y.: An algebraic multigrid method with guaranteed convergence rate. SIAM J. Sci. Comput. **34**, A1079–A1109 (2012)
11. Notay, Y.: An aggregation-based algebraic multigrid method. ETNA, Electron. Trans. Numer. Anal. **37**, 123–146 (2010)
12. Notay, Y.: Aggregation-based algebraic multigrid for convection-diffusion equations. SIAM J. Sci. Comput. **34**, A2288–A2316 (2012)
13. Raymond, J.P.: Feedback boundary stabilization of the two-dimensional Navier-Stokes equations. SIAM J. Control Optim. **45**(3), 790–828 (2006)
14. Ruge, J., Stüben, K.: Algebraic multigrid (amg). In: McCormich, E. (ed.) Multigrid Methods. Frontiers in Applied Mathematics, vol. 5, pp. 73–130. SIAM, Philadelphia (1987)
15. Saad, Y.: A flexible inner-outer preconditioned gmres algorithm. SIAM J. Sci. Comput. **14**(2), 461–469 (1993)
16. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia (2003)
17. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
18. Stoll, M., Wathen, A.: All-at-once solution of time-dependent Stokes control. J. Comp. Phys. **232**, 498–515 (2013)