Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems. Stokes control

Owe Axelsson[a,b], Shiraz Farouq[b], Maya Neytcheva[b]

[a] Institute of Geonics AS CR, Ostrava, The Czech Republic
[b] Uppsala University, Uppsala, Sweden

Abstract

The governing dynamics of fluid flow is stated as a system of partial differential equations referred to as the Navier-Stokes system. In industrial and scientific applications, fluid flow control becomes an optimization problem where the governing partial differential equations of the fluid flow are stated as constraints. When discretized, the optimal control of the Navier-Stokes equations leads to large sparse saddle point systems in two levels.

In this paper we consider distributed optimal control for the Stokes system and test the particular case when the arising linear system can be compressed after eliminating the control function. In that case, a system arises in a form which enables the application of an efficient block matrix preconditioner that previously has been applied to solve complex-valued systems in real arithmetic. Under certain conditions the condition number of the so preconditioned matrix is bounded by 2. The numerical and computational efficiency of the method in terms of number of iterations and execution time is favorably compared with other published methods.

Keywords: PDE-constrained optimization problems, finite elements, iterative solution methods, preconditioning

1 Introduction

Optimal control problems constrained by a partial differential equations (PDEs) arise in many applications, see for instance [9, 18, 16] and the references therein. Due to the increased dimension of the problems involving both state and control functions as well as Lagrange multipliers for the constraints, efficient numerical solution methods are required. To get an acceptable runtime, these methods have to be of iterative type. The matrices,

arising after discretization have a rich block structure. It is then crucial to write those matrices in a form for which efficient preconditioners can be constructed.

Recently, construction of efficient solution strategies for the Stokes control problem has drawn attention. In the work [26], a parameter-robust block-diagonal preconditioner is derived and its analysis is based on nonstandard norm argument which in some sense follows the idea of operator preconditiong discussed in [6] and the references therein. Subsequently, another work in [11] is based on the fundamental saddle point theory, which then in turn utilizes block approximations based on the work [3] and a variation of a *commutator* argument discussed in the book [20]; four preconditioners are suggested - two of block-diagonal and two of block lower-triangular form. One of the preconditioners is a re-derivation of the one in [26]. The four preconditioners are tested and their numerical and computational efficiencies are compared. The outcome shows that the block-diagonal preconditioners are performing best and therefore we compare our preconditioner with those two only.

We mention two more works, where Stokes control problems are studied, [22] and [17]. There, the cost functional differs from that considered in [26, 11], namely, it includes also control of the pressure variable. The preconditioning techniques proposed there rely on effective approximations of the (1,1) pivot block and the Schur complement. While the quality of the Schur complement approximation is mesh size (h) independent, it is not regularity parameter (β) independent. Some discussion on that case is included in [11]. In this work we do not consider them any further.

In [10], a technique, previously used to solve linear systems with complex-valued matrices in real arithmetic, is utilized in the context of optimal control problems, constrained by the Poisson and the convection-diffusion equations. In the present paper the latter technique is applied also for Stokes control problems. It is shown that the very favorable parameter-independent condition number 2 of the corresponding preconditioned matrix holds then also.

The remainder of the paper is structured as follows. In Section 2 we briefly formulate the control problems with a PDE constraint and present with more detail the control problem for Stokes equation. In Section 3 we state our preconditioner and derive condition number bounds. Section 4 contains a summary of some other preconditioning methods and a summary of computational complexity of the preconditioners considered is given in Section 5. Section 6 presents a performance comparisons between the different preconditioning techniques, referred to in this paper, in terms of total solution time and number of iterations. The paper ends with some concluding remarks.

2 Optimal control problems, constrained by PDEs. Stokes control

We recall the general form of a distributed control problem, minimized subject to a PDE, posed on some domain $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$:

$$\begin{aligned} \min_{y,u} \mathcal{J}(y,u) &= \frac{1}{2}\|y - y_d\|^2_{L_2(\Omega)} + \frac{1}{2}\beta\|u|^2_{L_2(\Omega)} \\ \text{such that } \mathcal{L}(y) &= u \end{aligned} \tag{1}$$

and satisfying appropriate boundary conditions. Here, $\mathcal{L}$ is a scalar or vector partial differential operator, y is the state function, u is the control function in the form of a distributed control, y_d is the target (desired) solution we want to achieve, $\beta > 0$ is the regularization parameter (also called the cost parameter) and in practice is chosen to be small. The PDE-constraint that models the underlying process to be controlled, is referred to as the *state equation*. This equation may itself contain a constraint, such as divergence-free criterion as for a Stokes problem.

We consider now the minimization of the cost functional with Stokes equations as the constraint. The independent variables for the Stokes equations are the velocity, denoted below by $\vec{y}$ and the pressure p. The control problem is stated as follows,

$$\begin{aligned} &\min_{\vec{y},\vec{u}} \mathcal{J}(y,u) \\ \text{s.t.} \quad -\Delta\vec{y} + \nabla p &= \vec{u} \quad \text{in } \Omega \\ \nabla \cdot \vec{y} &= 0 \quad \text{in } \Omega \\ \vec{y} &= \vec{g}_D \quad \text{on } \partial\Omega. \end{aligned} \tag{2}$$

We now solve the optimization problem concerning the attainment of desired state $\vec{y}_d$, by finding $\vec{u}$ such that the velocity $\vec{y}$ is close to $\vec{y}_d$. The constraint is implemented with Lagrange multipliers $\widetilde{\mathbf{l}}$, corresponding to the solution $\vec{y}$ and $\widetilde{\mathbf{m}}$, corresponding to the pressure p.

Since the Stokes system is *self-adjoint*, the two approaches for dealing with the problem, namely, *discretize-then-optimize* and *optimize-then-discretize*, yield the same optimality system, i.e.,

$$\mathcal{A}^F \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{u} \\ \widetilde{\mathbf{l}} \\ \widetilde{\mathbf{m}} \end{bmatrix} \equiv \begin{bmatrix} M_{\vec{y}} & 0 & 0 & \widetilde{F}_{\vec{y}} & \widetilde{B}^T \\ 0 & 0 & 0 & \widetilde{B} & 0 \\ 0 & 0 & \beta M_{\vec{y}} & -M^T_{\vec{y}} & 0 \\ \widetilde{F}_{\vec{y}} & \widetilde{B}^T & -M_{\vec{y}} & 0 & 0 \\ \widetilde{B} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{u} \\ \widetilde{\mathbf{l}} \\ \widetilde{\mathbf{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{0} \\ \tilde{\mathbf{f}} \\ \tilde{\mathbf{g}} \end{bmatrix}, \tag{3}$$

where

$$\widetilde{F}_{\vec{y}} = \int_\Omega \nabla\vec{\phi}_i : \nabla\vec{\phi}_j, \quad M_{\vec{y}} = \int_\Omega \vec{\phi}_i.\vec{\phi}_j, \quad \widetilde{B} = -\int_\Omega \psi_k \nabla.\vec{\phi}_j,$$

$$\mathbf{b} = \int_\Omega \widehat{\vec{y}}\vec{\phi}_i - \sum_{j=n_y+1}^{n_y+n_\partial} y_j \int_\Omega \nabla\vec{\phi}_i : \nabla\vec{\phi}_j,$$

$$\tilde{\mathbf{f}} = -\sum_{j=n_y+1}^{n_y+n_\partial} y_j \int_\Omega \nabla\vec{\phi}_i : \nabla\vec{\phi}_j, \quad \tilde{\mathbf{g}} = \sum_{j=n_y+1}^{n_y+n_\partial} y_j \int_\Omega \psi_i \nabla.\vec{\phi}_j.$$

The subscript $\vec{y}$ for M and $\widetilde{F}$ represent the standard notation for Gram matrices resulting from vector valued basis functions. Note that $\nabla\vec{\phi}_i : \nabla\vec{\phi}_j$ represents the component-wise scalar product. The coefficients $\{y_j\}_{j=n_v+1,\ldots,n_v+n_\partial}$ interpolate the boundary data $\vec{g}_D$. Recall that $\widetilde{\mathbf{l}}$ and $\widetilde{\mathbf{m}}$ are the adjoint variables associated with $\mathbf{u}$ and $\mathbf{p}$. We assume that the pair of the basis function set $\{\vec{\phi}_i\}$ and $\{\psi_j\}$ have been chosen such that the divergence constraint matrix B has full rank. Note that both $M_{\vec{y}}$ and $\widetilde{F}_{\vec{y}}$ are symmetric and positive definite (spd).

Next, using the relation $\mathbf{u} = \frac{1}{\beta}\widetilde{\mathbf{l}}$, we reduce the system as

$$\widetilde{\mathcal{A}}^R \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \widetilde{\mathbf{l}} \\ \widetilde{\mathbf{m}} \end{bmatrix} \equiv \begin{bmatrix} M_{\vec{y}} & 0 & \widetilde{F}_{\vec{y}} & \widetilde{B}^T \\ 0 & 0 & \widetilde{B} & 0 \\ \widetilde{F}_{\vec{y}} & \widetilde{B}^T & -\frac{1}{\beta}M_{\vec{y}} & 0 \\ \widetilde{B} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \widetilde{\mathbf{l}} \\ \widetilde{\mathbf{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \tilde{\mathbf{f}} \\ \tilde{\mathbf{g}} \end{bmatrix}. \tag{4}$$

Further, introducing the notations $\mathbf{l} = \frac{1}{\sqrt{\beta}}\widetilde{\mathbf{l}}$, $\mathbf{m} = \frac{1}{\sqrt{\beta}}\widetilde{\mathbf{m}}$, $f = \frac{1}{\beta}\tilde{f}$, $g = \frac{1}{\beta}\tilde{g}$ and $F = \sqrt{\beta}\widetilde{F}$, $B = \sqrt{\beta}\widetilde{B}$, we can transform the system to

$$\mathcal{A}^R \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{l} \\ \mathbf{m} \end{bmatrix} \equiv \begin{bmatrix} M & 0 & -F^T & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{l} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{5}$$

which has skew-symmetric off-diagonal blocks and is nonsingular. The latter can be seen, for instance by permuting rows 2 and 4 and columns 2 and 4. Then the diagonal blocks and the Schur complement become nonsingular.

We consider in this paper only preconditioners for the reduced system in the form $\widetilde{\mathcal{A}}^R$ and $\mathcal{A}^R$. To ease the notations, in the rest of the paper we drop the superscript 'R'.

3 A preconditioner for $\mathcal{A}$, based on its algebraic structure

3.1 The genesis of the idea

The matrix $\mathcal{A}$ can be seen as a two-by-two block matrix of the form

$$\mathcal{A} = \begin{bmatrix} \mathcal{M} & -\mathcal{F} \\ \mathcal{F} & \mathcal{M} \end{bmatrix},$$

where $\mathcal{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$ and $\mathcal{F} = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix}$. According to the theory, developed in [25, 1, 10], a very efficient preconditioner for matrices with such a structure is of the form

$$\mathcal{P}_F = \begin{bmatrix} \mathcal{M} & -\mathcal{F} \\ \mathcal{F} & \mathcal{M} + (\mathcal{F} + \mathcal{F}^T) \end{bmatrix}$$

and under certain conditions the condition number of $\mathcal{P}_F^{-1}\mathcal{A}$ is bounded by 2. The result holds for instance when $\mathcal{M}$ is positive semidefinite and $\mathcal{F}$ satisfies the requirement that $\mathcal{F} + \mathcal{F}^T$ is positive definite.

Further, it is shown that systems with $\mathcal{P}_F$ can be solved in a computationally very efficient manner, requiring just two solutions with the systems $\mathcal{M} + \mathcal{F}$ and $\mathcal{M} + \mathcal{F}^T$ and some vector updates. The algorithm utilizes the fact that we have an explicit form of the exact inverse of $\mathcal{P}_F$.

To make the paper self-contained, we include the algorithm to solve systems of the form

$$\begin{bmatrix} \mathcal{M} & -b\mathcal{F} \\ a\mathcal{F} & \mathcal{M} + \sqrt{ab}(\mathcal{F} + \mathcal{F}^T) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}.$$

Denote $\mathcal{H}_1 = \mathcal{M} + \sqrt{ab}\mathcal{F}$ and $\mathcal{H}_2 = \mathcal{M} + \sqrt{ab}\mathcal{F}^T$. Then, $\mathbf{x}$ and $\mathbf{y}$ are computed as follows (in our case $a = b = 1$):

Algorithm 1

1: Solve $\mathcal{H}_1\mathbf{g} = \mathbf{f}_1 + \sqrt{\frac{b}{a}}\mathbf{f}_2$.
2: Compute $\mathcal{M}\mathbf{g}$ and $\mathbf{f}_1 - \mathcal{M}\mathbf{g}$.
3: Solve $\mathcal{H}_2\mathbf{h} = \mathbf{f}_1 - \mathcal{M}\mathbf{g}$.
4: Compute $\mathbf{x} = \mathbf{g} + \mathbf{h}$ and $\mathbf{y} = -\sqrt{\frac{a}{b}}\mathbf{h}$.

The cost of performing Algorithm 1 includes one solution with each of the matrices $\mathcal{H}_1$ and $\mathcal{H}_2$, one matrix multiplication with $\mathcal{M}$ and five vector updates.

3.2 Analysis of the preconditioner for the optimal control problem with Stokes control

Based on the result, outlined in Section 3.1, we precondition the matrix $\mathcal{A}$ in (5) by a preconditioner formed by adding the matrix

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} + \begin{bmatrix} F^T & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} F+F^T & 2B^T \\ 2B & 0 \end{bmatrix}$$

to the lower $(2,2)$ block of the transformed matrix $\mathcal{A}$. Thus, we precondition $\mathcal{A}$ by the matrix

$$\mathcal{P}_F = \begin{bmatrix} M & 0 & -F & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M+F+F^T & 2B^T \\ B & 0 & 2B & 0 \end{bmatrix}.$$

To find the eigenvalues (λ) of the corresponding preconditioned matrix $\mathcal{P}_F^{-1}\mathcal{A}$ we consider the generalized eigenvalue problem

$$\lambda \begin{bmatrix} M & 0 & -F^T & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M_1 & \\ B & 0 & & \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} M & 0 & -F^T & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix},$$

where $M_1 = \begin{bmatrix} M+F+F^T & 2B^T \\ 2B & 0 \end{bmatrix}$. The preconditioned matrix is nonsingular, so $\lambda \neq 0$. There holds

$$\mu \begin{bmatrix} M & 0 & -F^T & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & F+F^T & 2B^T \\ 0 & 0 & 2B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{6}$$

where $\mu = \frac{1}{\lambda} - 1$. Here $\mu = 0$, i.e. $\lambda = 1$ if $\begin{bmatrix} F+F^T & 2B^T \\ 2B & 0 \end{bmatrix} \mathbf{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $|\mathbf{x}|+|\mathbf{y}| \neq 0$. However, since $F+F^T$ is spd and B has full rank, this system has only the trivial solution, $\mathbf{y} = 0$, so $\lambda = 1$ for $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, 0)$, $\mathbf{x} \neq 0$. Consider now $\mu \neq 0$ ($\lambda \neq 1$). Let $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$, where $\mathbf{y} \neq 0$.

It holds that $B\mathbf{y}_1 = 0$ and $\mu B\mathbf{x}_1 = 2B\mathbf{y}_1$ so also $B\mathbf{x}_1 = 0$. Note that for complex vectors this holds both for the real and imaginary parts of the vectors. From the remaining equations in (6) it follows

$$\begin{cases} \mu(M\mathbf{x}_1 - F^T\mathbf{y}_1 - B^T\mathbf{y}_2) = 0 \\ \mu(F\mathbf{x}_1 + B^T\mathbf{x}_2 + M\mathbf{y}_1) = (F+F^T)\mathbf{y}_1 + 2B^T\mathbf{y}_2 = M\mathbf{x}_1 + F\mathbf{y}_1 + B^T\mathbf{y}_2. \end{cases} \tag{7}$$

Let now $\widehat{\mathbf{x}}_i = M^{1/2}\mathbf{x}_i$, $\widehat{\mathbf{y}}_i = M^{1/2}\mathbf{y}_i$, $i = 1, 2$ and multiply the equations in (7) by $M^{-1/2}$. Then

$$\begin{cases} \widehat{\mathbf{x}}_1 = \widehat{F}^T\widehat{\mathbf{y}}_1 + \widehat{B}^T\widehat{\mathbf{y}}_2 \\ \mu(\widehat{F}\widehat{\mathbf{x}}_1 + \widehat{B}^T\widehat{\mathbf{x}}_2 + \widehat{\mathbf{y}}_1) = \widehat{\mathbf{x}}_1 + \widehat{F}\widehat{\mathbf{y}}_1 + \widehat{B}^T\widehat{\mathbf{y}}_2 \end{cases} \tag{8}$$

where $\widehat{F} = M^{-1/2}FM^{-1/2}$, $\widehat{B} = M^{-1/2}BM^{-1/2}$.

Since $B\mathbf{x}_1 = 0$, $B\mathbf{y}_1 = 0$, it follows that $\widehat{B}\widehat{\mathbf{x}}_1 = 0$, $\widehat{B}\widehat{\mathbf{y}}_1 = 0$. Hence, a multiplication of the equations in (8) with $\widehat{B}$ leads to $\widehat{B}\widehat{F}^T\widehat{\mathbf{y}}_1 + \widehat{B}\widehat{B}^T\widehat{\mathbf{y}}_2 = 0$, i.e. $\widehat{\mathbf{y}}_2 = -(\widehat{B}\widehat{B}^T)^{-1}\widehat{B}\widehat{F}^T\widehat{\mathbf{y}}_1$ and

$$\mu(\widehat{B}\widehat{F}\widehat{\mathbf{x}}_1 + \widehat{B}\widehat{B}^T\widehat{\mathbf{x}}_2) = \widehat{B}\widehat{F}\widehat{\mathbf{y}}_1 + \widehat{B}\widehat{B}^T\widehat{\mathbf{y}}_2 = \widehat{B}(\widehat{F} - \widehat{F}^T)\widehat{\mathbf{y}}_1.$$

Thus, $\widehat{\mathbf{x}}_1 = (I-P)\widehat{F}^T\widehat{\mathbf{y}}_1$, where $P = \widehat{B}^T(\widehat{B}\widehat{B}^T)^{-1}\widehat{B}$, i.e. P is a projection matrix. It follows that $\widehat{\mathbf{y}}_1^*\widehat{\mathbf{x}}_1 = \widehat{\mathbf{y}}_1^*\widehat{F}^T\widehat{\mathbf{y}}_1$ and $\widehat{\mathbf{x}}_1^*\widehat{\mathbf{x}}_1 = \widehat{\mathbf{x}}_1^*\widehat{F}^T\widehat{\mathbf{y}}_1$, where $\mathbf{x}^*$ stands for the complex conjugate vector. Hence, $\widehat{\mathbf{x}}_1^*\widehat{\mathbf{y}}_1 = \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1$ and $\widehat{\mathbf{x}}_1^*\widehat{\mathbf{x}}_1 = \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{x}}_1$.

Multiplications of the second equation in (8) with $\widehat{\mathbf{x}}_1^*$ and $\widehat{\mathbf{y}}_1^*$ leads to

$$\mu\,\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1 + \mu\,\widehat{\mathbf{x}}_1^*\widehat{\mathbf{y}}_1 = |\widehat{\mathbf{x}}_1|^2 + \widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{y}}_1 = 2|\widehat{\mathbf{x}}_1|^2 \tag{9}$$

$$\mu\,\widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{x}}_1 + \mu\,|\widehat{\mathbf{y}}_1|^2 = \widehat{\mathbf{y}}_1^*\widehat{\mathbf{x}}_1 + \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1. \tag{10}$$

It follows then from (9) that

$$\mu(\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1 + \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1) = 2|\widehat{\mathbf{x}}_1|^2 \tag{11}$$

and from (10) that

$$\mu(|\widehat{\mathbf{x}}_1|^2 + |\widehat{\mathbf{y}}_1|^2) = \widehat{\mathbf{y}}_1^*(\widehat{F} + \widehat{F}^T)\widehat{\mathbf{y}}_1 = 2\widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1. \tag{12}$$

Since by assumptions made, $\widehat{F} + \widehat{F}^T$ is spd, it follows from (12) that μ is real and positive. Further $\mu \le 2\|\widehat{F}\|$. Also from (11) and (12),

$$\mu^2(|\widehat{\mathbf{x}}_1|^2 + |\widehat{\mathbf{y}}_1|^2) = 2\mu\widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1 = 2(2|\widehat{\mathbf{x}}_1|^2 - \mu\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1),$$

so, since $|\widehat{\mathbf{y}}_1| \neq 0$ then $\mu^2 < 4$. Hence, $\mu \le 2\min\{1, \|\widehat{F}\|\}$. For $\mu = 2$ we get $\lambda = 1/3$. Therefore, since $\mu > 0$, it follows that $1/3 \le \lambda \le 1$.

If F is spd, as for the Stokes problem, then (11) shows that

$$\mu(\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1 + \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1) = 2\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{y}}_1 \le 2(\widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1)^{1/2}(\widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1)^{1/2} \le \widehat{\mathbf{x}}_1^*\widehat{F}\widehat{\mathbf{x}}_1 + \widehat{\mathbf{y}}_1^*\widehat{F}\widehat{\mathbf{y}}_1,$$

so $\mu \le 1$ and $1/2 \le \lambda \le 1$. These are the same eigenvalue bounds that hold for optimal control of the Poisson equation, see [10].

4 Other preconditioners for Stokes constrained optimal control problems

In order to provide some basic understanding on the preconditioning techniques that are currently available for the Stokes control problem, we briefly describe the work done in [11] along with the best performing preconditioner proposed in [26].

4.1 Block-diagonal preconditioner for the reduced system, based on nonstandard norms

A preconditioner for the system (4) that is parameter-independent has been proposed in [26] using non-standard norm arguments. It is of the form

$$\mathcal{P}_{nsn} = \begin{bmatrix} M_{\vec{y}} + \sqrt{\beta}F_{\vec{y}} & 0 & 0 & 0 \\ 0 & \frac{1}{\beta}(M_{\vec{y}} + \sqrt{\beta}F_{\vec{y}}) & 0 & 0 \\ 0 & 0 & (F_p^{-1} + \sqrt{\beta}M_p^{-1})^{-1} & 0 \\ 0 & 0 & 0 & \beta(F_p^{-1} + \sqrt{\beta}M_p^{-1})^{-1} \end{bmatrix}. \tag{13}$$

The idea is to find a norm for which (4) satisfies the well-posedness inf-sup condition, cf. [2]. This norm then is utilized to construct a preconditioner to (4). The condition number of the corresponding preconditioned system is shown to be 4.25, see [27].

On our test problems, the performance of this preconditioner shows both h- and β-independence, see Tables 2 and 7.

4.2 Block-diagonal preconditioner for the reduced system, utilising the underlying system of PDEs

Another preconditioner for the system in (4) has been recently derived in [11], using the so-called commutator argument. The preconditioner is of the form

$$\mathcal{P}_{cta} = \begin{bmatrix} M_{\vec{y}} & 0 & 0 & 0 \\ 0 & (F_{\vec{y}} + \frac{1}{\sqrt{\beta}}M_{\vec{y}})M_{\vec{y}}^{-1}(F_{\vec{y}} + \frac{1}{\sqrt{\beta}}M_{\vec{y}})^T & 0 & 0 \\ 0 & 0 & F_p & 0 \\ 0 & 0 & & (M_p^{-1}F_pM_p^{-1} + \frac{1}{\beta}F_p^{-1})^{-1} \end{bmatrix}. \tag{14}$$

Utilizing the knowledge of the fundamental saddle point theory and viewing (4) as a two-by-two block structure, one can see that the pivot block (1,1) is structurally equivalent to a Poisson control problem; hence this block is approximated by an equivalent preconditioner for the Poisson control problem as described in [21]. For the Schur complement, however, two approximations are required – one for $BM_p^{-1}B^T$ and another for $B\Theta^{-1}B^T$, where $\Theta = FM^{-1}F + \frac{1}{\beta}M$. Here M_p is the pressure, i.e. scalar, mass matrix and F_p is the scalar analogue of $F_{\vec{y}}$. The first approximation is based on the well known result that $BM_p^{-1}B^T \approx F_p$, cf. [20]. The second approximation is obtained using the assumption that a convection-diffusion operator can also be defined on the pressure space and the *commutator* of the convection-diffusion operators with the gradient operator ∇ is small in some sense, cf. [Chapter 8, [20]]. Using the strategy, it is shown that

$B\Theta^{-1}B^T \approx F_p\Theta^{-1}M_p \approx (M_p^{-1}F_pM_p^{-1} + \frac{1}{\beta}F_p^{-1})$. We note that the quality of the so-arising preconditioner $\mathcal{P}_{cta}$ has not been rigorously quantified.

The performance of $\mathcal{P}_{cta}$ is illustrated in Tables 3 and 8. Similarly to the results in [11], we observe that for the tested range of the parameters h and β, it shows h-independence, but not fully β-independence.

5 Computational complexity of the preconditioners $\mathcal{P}_{nsn}$, $\mathcal{P}_{cta}$ and $\mathcal{P}_F$

We note first, that since in the Stokes control problem F is the discrete vector Laplacian (thus $F = F^T$), we have that

$$\mathcal{H}_1 = \mathcal{H}_2 \equiv \mathcal{H} = \begin{bmatrix} M+F & B^T \\ B & 0 \end{bmatrix}. \tag{15}$$

In order to efficiently solve systems with $\mathcal{H}$ we introduce

$$\mathcal{P}_{\mathcal{H}} = \begin{bmatrix} M+F & 0 \\ B & -S_p \end{bmatrix} \tag{16}$$

where $S_p^{-1} = \sqrt{\beta}M_p^{-1} + F_p^{-1}$ is a well-known approximation of the corresponding exact Schur complement obtained in [3]. Moreover, the preconditioner $P_{\mathcal{H}}$ in (16) has been shown to possess a mesh-independent convergence, cf. [24, 13].

A successful application of the preconditioners $\mathcal{P}_{nsn}$, $\mathcal{P}_{cta}$ and $\mathcal{P}_F$ requires that each block can be approximated efficiently in a numerical sense. In the numerical tests in Section 6, the following approximations are used.

(i) Based on the analysis done in [12] and used in [11], a system with a mass matrix block, whether in the velocity space or the pressure space, i.e., $M_{\vec{y}}$ and M_p, is solved by the Chebyshev semi-iteration method, terminated either after at most 20 iterations or when the norm of the relative residual becomes less than 10^{-4}.

(ii) All blocks of the form F_p, $F_{\vec{v}} + \frac{1}{\sqrt{\beta}}M_{\vec{v}}$, $(F_{\vec{v}} + \frac{1}{\sqrt{\beta}}M_{\vec{v}})^T$ and $M_{\vec{y}} + \sqrt{\beta}F_{\vec{y}}$ are replaced by one V-cycle of an Algebraic Multigrid (AMG) solver.

Table 1 summarizes the computational costs for each preconditioner per iteration.

Remark (Solving the 2×2 $\mathcal{H}$ block)**.** We can numerically solve the $\mathcal{H}$ block in various ways. One such option is by preconditioning it with $\mathcal{P}_H$ and using an iterative solver such as the flexible GMRES (FGMRES) method ([14]). Another way is to use the *inexact Uzawa* method [4, 19], in which case $\mathcal{P}_H$ then acts as a splitting matrix for the iteration,

$$\boldsymbol{\rho_{n+1}} = \boldsymbol{\rho_n} + \mathcal{P}_H^{-1}\mathcal{H}\mathbf{r_n} \tag{17}$$

Preconditioner	Operations
$\mathcal{P}_{nsn}$	4 block solves with AMG, and 2 block solves with Chebyshev semi-iterations (Tables 2 and 7)
$\mathcal{P}_{cta}$	4 block solves with AMG, and 3 block solves with Chebyshev semi-iterations (Tables 3 and 8)
$\mathcal{P}_F$	2 block solves with AMG, and 1 block solve with Chebyshev semi-iterations per each block $\mathcal{H}$. Since two systems with $\mathcal{H}$ are solved, in total we have 4 block solves with AMG, and 2 block solves with Chebyshev semi-iterations. In turn, each $\mathcal{H}$ is solved using the preconditioner $P_{\mathcal{H}}$ in (16) (Tables 4, 5, 6, 9, 10 and 11)

Table 1: Summary of the computational complexity of the preconditioners

where $\boldsymbol{\rho_n} = (\mathbf{x}, \mathbf{y})^{\mathbf{T}}$ and $\mathbf{r_n}$ are the solution and residual vectors at the n-th iteration. However, a parameter τ is introduced with S_p in $\mathcal{P}_H$ to improve performance. This approach has been previously used in [22] in a similar context with $\tau = 3/5$.

6 Numerical Results

In this section we demonstrate the numerical and computational performance of the three preconditioners, i.e., $\mathcal{P}_{cta}$, $\mathcal{P}_{nsn}$, and $\mathcal{P}_F$. To this end, we use the following two test problems.

Problem 1 (Velocity tracking problem with distributed control, cf. [26])**.** *Find the state $y \in H_0^1(\Omega)$ and the control $u \in L^2(\Omega)$ that minimizes the cost functional (2) with $\Omega = [0,1]^2$ and a desired state $\vec{y}_d(x_1, x_2) = (y_{d,1}(x_1, x_2), y_{d,2}(x_1, x_2))$ given by*

$$y_{d,1}(x_1, x_2) = 10\frac{\partial}{\partial x_2}(\varphi(x_1)\varphi(x_2)) \quad \textit{and} \quad y_{d,2}(x_1, x_2) = -10\frac{\partial}{\partial x_1}(\varphi(x_1)\varphi(x_2)),$$

where $\varphi(z) = (1 - cos(0.8\pi z))(1 - z)^2$.

Problem 2 (Lid-driven cavity)**.** $\widehat{\vec{y}} = x_2\mathbf{i} - x_1\mathbf{j}$ *with*

$$\vec{y} = \begin{cases} -\mathbf{j} & \textit{if } x_1 = 1, 0 \leq x_2 \leq 1, \\ 0 & \textit{otherwise.} \end{cases}$$

Here $\mathbf{i}$ *and* $\mathbf{j}$ *are the unit vectors in x and y directions, correspondingly.*

We discretize the problems with *inf-sup* stable Taylor-Hood finite element basis functions, (also known as the Q2-Q1 stable pair), see for details [5]. Thus, the state y, the control u and the adjoint $\mathbf{l}$ are discretized using piece-wise quadratic (Q2) basis functions, while the pressure p and its corresponding adjoint $\mathbf{m}$ are discretized using piece-wise linear (Q1) basis functions. The results obtained by solving the problem using the three different preconditioners discussed earlier, i.e., $\mathcal{P}_{nsn}$, $\mathcal{P}_{cta}$, and $\mathcal{P}_F$ are presented in Tables 2-5 for

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**74**(4+23)	**74**(4+23)	**71**(4+24)	**61**(4+24)	**53**(4+23)	**41**(4+23)	**36**(4+22)	**31**(4+23)	**25**(4+22)
	0.16	0.16	0.156	0.134	0.116	0.09	0.079	0.068	0.056
19078	**82**(4+22)	**82**(4+22)	**80**(4+23)	**71**(4+23)	**59**(4+23)	**45**(4+22)	**39**(4+22)	**33**(4+21)	**27**(4+21)
	0.789	0.79	0.772	0.688	0.57	0.436	0.38	0.321	0.266
75014	**90**(4+21)	**89**(4+21)	**87**(4+22)	**77**(4+22)	**65**(4+22)	**49**(4+21)	**41**(4+21)	**35**(4+21)	**29**(4+19)
	3.885	3.848	3.764	3.338	2.822	2.131	1.788	1.526	1.276
297478	**96**(4+21)	**97**(4+21)	**95**(4+21)	**85**(4+21)	**72**(4+20)	**53**(4+19)	**43**(4+19)	**37**(4+20)	**29**(4+19)
	18.654	18.872	18.495	16.574	14.043	10.329	8.391	7.252	5.713

Table 2: Problem P1: Performance of the preconditioner $\mathcal{P}_{nsn}$

Problem 1 and in Tables 7-10 for Problem 2. In accordance with the experiments performed in [26, 11], when applying $\mathcal{P}_{cta}$ and $\mathcal{P}_{nsn}$, we use MINRES ([15]) as an *outer* solver. The outer solver when using the preconditioner $\mathcal{P}_F$ is the FGMRES. Systems with $\mathcal{P}_{\mathcal{H}}$ are also solved with an (inner) FGMRES solver or with an inexact Uzawa type method. The relative convergence tolerance for both MINRES and outer FGMRES is set to 10^{-6}.

All preconditioners are implemented in C++ and executed using the open source finite element library deal.ii ([7]) (Version 8.2.1) that provides the meshes, the finite element discretization and the basic iterative methods. Further, deal.ii supplies interface to the Trilinos library [23], giving access to the Trilinos Algebraic Multigrid (AMG) solver. All experiments are performed on Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz-2.80GHz with installed memory RAM of 4GB. The main settings for the AMG to mention are : aggregation_threshold = 0.8, and smoother_type=symmetric Gauss-Seidel, with the rest taking on package default values. The results are presented in the Tables 2-10 and obey the following convention.

- Regarding $\mathcal{P}_{nsn}$ and $\mathcal{P}_{cta}$: For each value of β and h, in the first row we show the number of outer (MINRES) iterations in bold, followed in brackets by the number of V-cycle AMG iterations (always 4) and the average number of Chebyshev semi-iterations per outer iteration. The next row shows the total solution time (in seconds).

- Regarding $\mathcal{P}_F$: the first row shows the outer FGMRES iterations in bold, followed by the the average number of inner FGMRES iterations (or inexact Uzawa iterations) required to solve the blocks $\mathcal{H}_1$ and $\mathcal{H}_2$ (separated by a plus sign) at each outer iteration; In the following row we show the total solution time (in seconds).

To better illustrate the performance of $\mathcal{P}_F$, we present three experiments per test problem. In the first experiment the stopping tolerance is 10^{-4} (Tables 4 and 9). In the second experiment the inner FGMRES iterations are fixed to 4 (Tables 5 and 10). At last, in Tables 6 and 11, we show the result obtained using inexact Uzawa method for the two problems. When reporting the performance of $\mathcal{P}_F$, we do not include information regarding AMG and the Chebyshev iteration method as these do not bring any new insight compared to the performance observed for the other two preconditioners.

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**101**(4+29)	**102**(4+29)	**93**(4+29)	**84**(4+30)	**73**(4+31)	**57**(4+33)	**42**(4+32)	**30**(4+32)	**24**(4+31)
	0.31	0.308	0.282	0.257	0.232	0.187	0.14	0.105	0.08
19078	**122**(4+28)	**119**(4+28)	**108**(4+28)	**96**(4+28)	**84**(4+29)	**63**(3+30)	**50**(4+31)	**39**(4+31)	**28**(4+30)
	1.569	1.509	1.364	1.217	1.095	0.823	0.68	0.538	0.387
75014	**143**(4+27)	**134**(4+27)	**122**(4+26)	**107**(4+26)	**94**(4+27)	**75**(4+27)	**60**(4+28)	**47**(4+28)	**36**(4+28)
	8.184	7.612	6.8	5.958	5.239	4.205	3.406	2.681	2.123
297478	**156**(4+27)	**145**(4+26)	**140**(4+25)	**119**(4+25)	**104**(4+25)	**91**(4+26)	**66**(4+25)	**54**(4+27)	**41**(4+26)
	39.44	35.767	34.146	29.042	25.136	22.215	16.188	13.281	10.099

Table 3: Problem 1: Performance of the preconditioner $\mathcal{P}_{cta}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**6**(10+10)	**8**(9+9)	**8**(8+8)	**7**(7+7)	**7**(6+5)	**6**(6+5)	**5**(6+5)	**4**(6+6)	**3**(6+6)
	0.205	0.215	0.219	0.16	0.142	0.136	0.1	0.088	0.066
19078	**6**(10+11)	**8**(9+10)	**8**(8+8)	**7**(8+7)	**7**(7+5)	**6**(6+5)	**5**(6+5)	**4**(6+6)	**3**(6+6)
	0.732	0.919	0.799	0.629	0.592	0.446	0.378	0.317	0.242
75014	**6**(12+12)	**7**(10+10)	**8**(9+8)	**7**(8+7)	**7**(7+5)	**6**(6+5)	**5**(6+5)	**5**(6+5)	**4**(6+5)
	3.378	3.459	3.481	2.74	2.436	1.793	1.559	1.515	1.253
297478	**6**(12+12)	**8**(11+11)	**8**(10+10)	**7**(9+8)	**7**(7+6)	**6**(6+5)	**5**(6+5)	**5**(6+5)	**4**(6+5)
	15.345	18.307	16.589	12.706	10.565	7.744	6.715	6.642	5.336

Table 4: Problem 1: Performance of the preconditioner $\mathcal{P}_F$. Stopping tolerance for the inner FGMRES 10^{-4}

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**8**(4+4)	**9**(4+4)	**9**(4+4)	**8**(4+4)	**7**(4+4)	**6**(4+4)	**5**(4+4)	**4**(4+4)	**3**(4+4)
	0.121	0.138	0.138	0.142	0.122	0.098	0.083	0.085	0.055
19078	**8**(4+4)	**9**(4+4)	**9**(4+4)	**8**(4+4)	**7**(4+4)	**6**(4+4)	**6**(4+4)	**5**(4+4)	**3**(4+4)
	0.478	0.536	0.538	0.483	0.432	0.373	0.367	0.314	0.203
75014	**8**(4+4)	**9**(4+4)	**9**(4+4)	**8**(4+4)	**7**(4+4)	**6**(4+4)	**6**(4+4)	**5**(4+4)	**4**(4+4)
	1.947	2.171	2.183	1.975	1.751	1.512	1.501	1.263	1.042
297478	**8**(4+4)	**9**(4+4)	**9**(4+4)	**8**(4+4)	**7**(4+4)	**6**(4+4)	**6**(4+4)	**5**(4+4)	**4**(4+4)
	8.496	9.45	9.473	8.547	7.601	6.576	6.489	5.475	4.516

Table 5: Problem 1: Performance of the preconditioner $\mathcal{P}_F$. Number of iterations for the inner FGMRES fixed to 4

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**8**(4+4)	**10**(4+4)	**12**(4+4)	**12**(4+4)	**11**(4+4)	**11**(4+4)	**9**(4+4)	**7**(4+4)	**7**(4+4)
	0.085	0.101	0.117	0.117	0.11	0.11	0.092	0.074	0.075
19078	**8**(4+4)	**10**(4+4)	**12**(4+4)	**12**(4+4)	**12**(4+4)	**11**(4+4)	**10**(4+4)	**8**(4+4)	**6**(4+4)
	0.345	0.417	0.488	0.488	0.497	0.455	0.418	0.344	0.266
75014	**8**(4+4)	**10**(4+4)	**12**(4+4)	**12**(4+4)	**12**(4+4)	**11**(4+4)	**10**(4+4)	**8**(4+4)	**6**(4+4)
	1.467	1.775	2.088	2.095	2.091	1.942	1.788	1.465	1.14
297478	**8**(4+4)	**10**(4+4)	**12**(4+4)	**12**(4+4)	**12**(4+4)	**11**(4+4)	**10**(4+4)	**8**(4+4)	**6**(4+4)
	6.562	7.943	9.327	9.334	9.344	8.645	7.947	6.534	5.077

Table 6: Problem 1: Performance of the preconditioner $\mathcal{P}_F$. Number of iterations for the inner inexact Uzawa iteration is fixed to 4

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**80**(4+25)	**64**(4+25)	**60**(4+25)	**52**(4+26)	**46**(4+27)	**41**(4+27)	**38**(4+27)	**34**(4+27)	**30**(4+27)
	0.212	0.194	0.199	0.139	0.125	0.112	0.104	0.094	0.092
19078	**80**(4+24)	**68**(4+24)	**64**(4+25)	**58**(4+25)	**50**(4+26)	**46**(4+27)	**40**(4+27)	**36**(4+27)	**32**(4+27)
	0.89	0.757	0.714	0.643	0.561	0.522	0.452	0.408	0.366
75014	**78**(4+24)	**70**(4+24)	**66**(4+24)	**62**(4+24)	**56**(4+25)	**50**(4+26)	**44**(4+27)	**37**(4+27)	**33**(4+27)
	3.485	3.125	2.954	2.775	2.521	2.261	1.993	1.683	1.51
297478	**74**(4+23)	**74**(4+23)	**68**(4+23)	**64**(4+24)	**58**(4+24)	**53**(4+25)	**48**(4+26)	**42**(4+26)	**35**(4+27)
	14.457	14.482	13.328	12.541	11.396	10.455	9.524	8.344	7.002

Table 7: Problem 2: Performance of the preconditioner $\mathcal{P}_{nsn}$

Lid-driven cavity problem

For completeness, in Table 12 we illustrate the behavior of the cost functional $\mathcal{J}$ for different values of β for the velocity tracking problem. As is known, how close the state y approaches the desired state $\widehat{y}$ is determined by the regularization parameter β. Hence, we observe that $\|y-\widehat{y}\|$ continues to decrease with decreasing β while $\|u\|$ stops increasing further around $\beta \leq 10^{-4}$. This implies that the optimal value of β for the problem is fairly small, i.e., around 10^{-8} to 10^{-10}. Table 12 is produced using the preconditioner $\mathcal{P}_F$. For mesh size 2^{-6}, we show the number of *outer* FGMRES iterations represented as "iter". $\|u\|$ represents the discrete $L^2(\Omega)$ norm of the control u, $\|y-\widehat{y}\|$ measures how closely the state y matches the desired state $\widehat{y}$, $\|y-\widehat{y}\|/\|\widehat{y}\|$ measures the relative error. The column $\mathcal{J}$ shows the calculated cost functional, $\|b-\mathcal{A}x\|/\|b\|$ represents the residual norm of the related Kuhn-Karush-Tucker (KKT) system of equations to show that the system has converged in the discrete $L^2(\Omega)$ norm. The last column shows the time (sec) to solve the system. We note that the differences between the cost functionals using all other preconditioners are insignificant.

Finally, we reproduce the plots from [26] for state y, control u and pressure p for $\beta = 10^{-6}$ and mesh size $h = 2^{-6}$ using [8]. We use the preconditioner $\mathcal{P}_{nsn}$ to generate these plots. Clearly, the region with the highest magnitude in Figure 1(a) contains the vectors

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**97**(4+30)	**93**(4+30)	**74**(4+31)	**67**(4+32)	**62**(4+33)	**54**(4+34)	**43**(4+35)	**34**(4+35)	**31**(4+35)
	0.346	0.331	0.266	0.256	0.227	0.205	0.166	0.131	0.121
19078	**106**(4+29)	**101**(4+29)	**80**(4+29)	**74**(4+30)	**68**(4+31)	**62**(4+33)	**53**(4+34)	**41**(4+35)	**33**(4+35)
	1.523	1.438	1.141	1.059	0.988	0.909	0.793	0.625	0.5
75014	**113**(4+28)	**107**(4+28)	**84**(4+28)	**80**(4+28)	**73**(4+29)	**68**(4+30)	**59**(4+32)	**50**(4+33)	**39**(4+34)
	6.594	6.204	4.832	4.61	4.229	3.984	3.501	3.028	2.428
297478	**119**(4+27)	**110**(4+26)	**88**(4+26)	**83**(4+27)	**77**(4+28)	**72**(4+29)	**66**(4+30)	**56**(4+31)	**47**(4+32)
	29.817	27.129	21.626	20.429	19.084	17.887	16.439	14.182	12.106

Table 8: Problem 2: Performance of the preconditioner $\mathcal{P}_{cta}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**6**(7+7)	**8**(8+7)	**9**(8+8)	**9**(8+7)	**9**(7+7)	**8**(7+7)	**8**(7+7)	**7**(7+6)	**6**(8+7)
	0.159	0.203	0.228	0.216	0.206	0.196	0.203	0.158	0.144
19078	**6**(7+8)	**7**(8+8)	**8**(9+9)	**9**(8+8)	**9**(8+7)	**9**(7+7)	**8**(7+7)	**8**(7+7)	**7**(7+6)
	0.563	0.692	0.844	0.89	0.865	0.813	0.736	0.7	0.612
75014	**5**(7+8)	**7**(9+8)	**7**(10+9)	**8**(9+8)	**8**(9+8)	**9**(8+7)	**8**(7+7)	**8**(7+7)	**8**(7+7)
	1.904	2.989	3.298	3.443	3.352	3.566	2.986	3.004	2.901
297478	**5**(8+8)	**6**(9+9)	**7**(11+10)	**7**(10+9)	**8**(9+8)	**8**(9+8)	**9**(8+7)	**9**(8+7)	**8**(7+7)
	8.864	11.701	15.516	14.579	14.752	14.432	15.361	14.863	13.097

Table 9: Problem 2: Performance of the preconditioner $\mathcal{P}_F$. Stopping tolerance for the inner FGMRES 10^{-4}

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**8**(4+4)	**10**(4+4)	**11**(4+4)	**11**(4+4)	**12**(4+4)	**11**(4+4)	**11**(4+4)	**12**(4+4)	**14**(4+4)
	0.126	0.245	0.189	0.198	0.182	0.170	0.170	0.193	0.221
19078	**7**(4+4)	**9**(4+4)	**10**(4+4)	**11**(4+4)	**11**(4+4)	**11**(4+4)	**11**(4+4)	**11**(4+4)	**11**(4+4)
	0.424	0.532	0.589	0.644	0.65	0.652	0.654	0.654	0.651
75014	**7**(4+4)	**9**(4+4)	**10**(4+4)	**10**(4+4)	**11**(4+4)	**11**(4+4)	**11**(4+4)	**10**(4+4)	**10**(4+4)
	1.724	2.183	2.408	2.413	2.641	2.651	2.663	2.447	2.499
297478	**7**(4+4)	**8**(4+4)	**9**(4+4)	**9**(4+4)	**10**(4+4)	**10**(4+4)	**11**(4+4)	**11**(4+4)	**10**(4+4)
	7.836	8.561	9.509	9.550	10.526	10.562	11.544	11.588	10.654

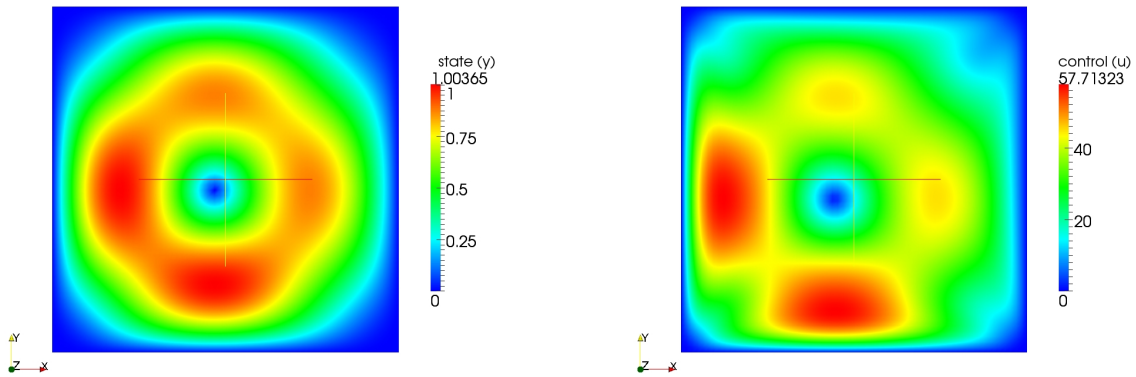
Table 10: Problem 2: Performance of the preconditioner $\mathcal{P}_F$. Number of iterations for the inner FGMRES fixed to 4

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
4934	**6**(6+6)	**9**(6+6)	**11**(6+6)	**12**(6+6)	**13**(6+6)	**13**(6+6)	**12**(6+6)	**11**(6+6)	**11**(6+6)
	0.096	0.128	0.153	0.169	0.181	0.182	0.169	0.156	0.162
19078	**6**(6+6)	**8**(6+6)	**10**(6+6)	**12**(6+6)	**13**(6+6)	**14**(6+6)	**13**(6+6)	**13**(6+6)	**12**(6+6)
	0.384	0.486	0.591	0.701	0.759	0.817	0.8	0.765	0.711
75014	**6**(6+6)	**8**(6+6)	**10**(6+6)	**11**(6+6)	**13**(6+6)	**14**(6+6)	**14**(6+6)	**13**(6+6)	**13**(6+6)
	1.661	2.121	2.586	2.818	3.298	3.542	3.56	3.334	3.328
297478	**6**(6+6)	**7**(6+6)	**9**(6+6)	**11**(6+6)	**12**(6+6)	**13**(6+6)	**14**(6+6)	**14**(6+6)	**13**(6+6)
	7.445	8.443	10.503	12.593	13.664	14.757	15.88	15.94	14.91

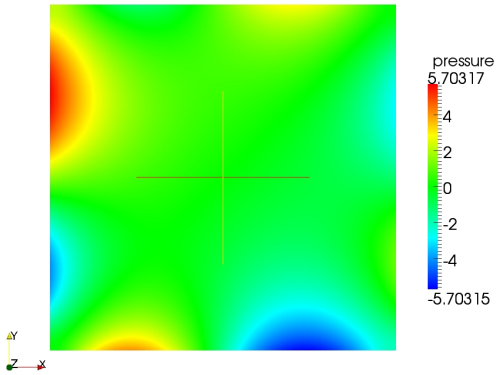
Table 11: Problem 2: Performance of the preconditioner $\mathcal{P}_F$. Number of iterations for the inner inexact Uzawa iteration is fixed to 6

β	iter	$\|u\|_2$	$\|y-\widehat{y}\|_2$	$\|y-\widehat{y}\|_2/\|\widehat{y}\|_2$	J	$\|b-\mathcal{A}x\|_2/\|b\|_2$	time
1e-02	8	1.39e+02	4.08e-01	9.66e-01	9.70e+01	2.80e-09	1.947
1e-03	9	1.06e+03	3.12e-01	7.39e-01	5.61e+02	2.60e-09	2.171
1e-04	9	3.16e+03	9.84e-02	2.33e-01	5.00e+02	2.38e-09	2.183
1e-05	8	4.06e+03	1.61e-02	3.80e-02	8.25e+01	1.72e-09	1.975
1e-06	7	4.27e+03	2.90e-03	6.86e-03	9.10e+00	2.68e-09	1.751
1e-07	6	4.33e+03	6.29e-04	1.49e-03	9.39e-01	3.13e-09	1.512
1e-08	6	4.37e+03	1.46e-04	3.46e-04	9.53e-02	1.49e-09	1.501
1e-09	5	4.38e+03	3.34e-05	7.92e-05	9.61e-03	1.51e-09	1.263
1e-10	4	4.39e+03	7.26e-06	1.72e-05	9.64e-04	1.35e-09	1.042

Table 12: Problem 1: Study of the cost functional $\mathcal{J}$ of the distributed optimal control problem constrained by the Stokes system



(a) State **y** (b) Control **u**



(c) Pressure **p**

Figure 1: State y, control u and pressure p distribution for $h = 2^{-6}$ and $\beta = 10^{-6}$.

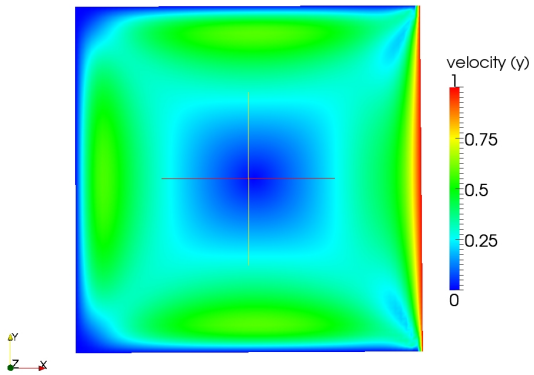
of the highest magnitude of one for the state **y**. The region of the highest magnitude in Figure 1(b) corresponds to the value 57.7 for the control **u**. At last, in Figure 1(c) we can see that the pressure **p** lies in the range of -5.7 to 5.7.

For the lid driven cavity problem using $\mathcal{P}_F$, the plots for state y, control u and pressure p for $\beta = 10^{-6}$ and mesh size $h = 2^{-6}$ are given in Figures 2(a)-2(c).

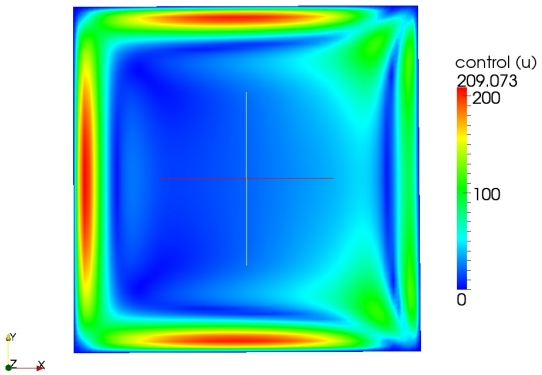
Discussion

We have tested the three preconditioners, $\mathcal{P}_{nsn}$, $\mathcal{P}_{cta}$ and $\mathcal{P}_F$ for two different problems, namely, the velocity tracking problem and the lid-driven cavity problem. All preconditioners are robust with respect to the mesh size h. The iterations for the preconditioner $\mathcal{P}_{cta}$ show some more pronounced dependence on β.

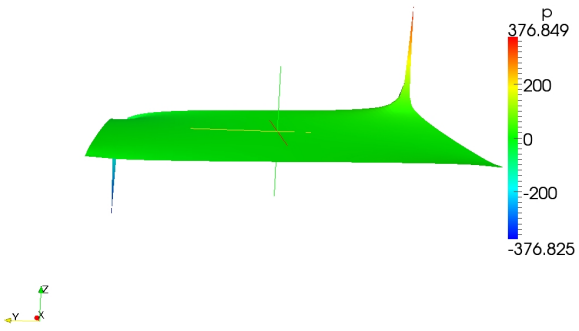
As predicted by the theory, $\mathcal{P}_F$ is fully parameter-independent and converges in very few iterations for both the tested problems. For the velocity tracking problem it is the best performing both in terms of number of iterations and execution time. For the lid driven cavity problem, however, it is still the best performing in terms of iterations and time apart from the cases when beta becomes of order 10^{-7} and smaller, then the preconditioner $\mathcal{P}_{nsn}$ exhibits slightly faster performance in time. A detailed look at the numerical results indicates that the performance of the AMG method we use here worsens for smaller β and h which opens the possibility to look for better solvers for the blocks $\mathcal{H}_1$ and $\mathcal{H}_2$.



(a) State **y**



(b) Control **u**



(c) Pressure **p**

Figure 2: State y, control u and pressure p distribution for $h = 2^{-6}$ and $\beta = 10^{-6}$.

7 Conclusions

To the best of our knowledge there exist only two preconditioners for the Stokes control problem that are independent with respect to both the mesh size h and regularization parameter β, cf. [11, 26]. In this paper, by transforming the saddle point system to acquire the form of a Stokes control problem and then exploiting the rich block structure, we have been able to construct a competitive preconditioner with a favourable condition number. The numerical experiments indicate a very good performance of preconditioner $(\mathcal{P}_F)$ in terms of number of iterations as well as small execution time when compared to other preconditioners for the same target problem.

References

[1] O. Axelsson, M. Neytcheva, B. Ahmad. A comparison of iterative methods to solve complex valued linear algebraic systems. *Numer. Algorithms*, 66:811–841, 2014.

[2] I. Babuska, A.K. Aziz. *Survey lectures on the mathematical foundations of the finite element method*, pages 1–359. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*. Academic Press, New York, 1972.

[3] J. Cahouet, J. P. Chabard. Some fast 3d finite element solvers for the generalized Stokes problem. *Int. J. Numer. Meth. Fl.*, 8:869–895, 1988.

[4] H. C. Elman and G. H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31:1645–1661, 1994.

[5] F. Brezzi, M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.

[6] R. Hiptmair. Operator preconditioning. *Comput. Math. Appl.*, 52:699–706, 2006.

[7] W. Bangerth, R. Hartmann, G. Kanschat. deal.ii-a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software*, 33, 2007. `http://doi.acm.org/10.1145/1268776.1268779`.

[8] J. Ahrens, B. Geveci, C. Law. *ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook*. Elsevier, 2005.

[9] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.

[10] O. Axelsson, S. Farouq, M. Neytcheva. Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems. Poisson and

convection-diffusion control. Technical Report TR 2015-024, Department of Information Technology, Uppsala University, August 2015.

[11] J.W. Pearson. On the development of parameter-robust preconditioners and commutator arguments for solving Stokes control problems. *ETNA*, 44:53–72, 2015.

[12] A. J. Wathen, T. Rees. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *ETNA*, 34:125–135, 09 2008.

[13] M. A. Olshanskii, J. Peters, A. Reusken. Uniform preconditioners for a parameter dependent saddle point problem with application to generalized Stokes interface equations. *Numer. Math.*, 105:159–191, 2006.

[14] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14:461–469, 1993.

[15] C. Paige, M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[16] A. Borzí, V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*, volume 8 of *Computational Science and Engineering*. SIAM, Philadelphia, PA, 2012.

[17] A. Drăgănescu, A. M. Soane. Multigrid solution of a distributed optimal control problem constrained by the Stokes equations. *Appl. Math. Comput.*, 219:5622–5634, January 2013.

[18] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich. *Optimization with PDE Constraints*. Springer-Verlag, 2009.

[19] J.H. Bramble, J. E. Pasciak, A. T. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM J. Numer. Anal.*, 34:1072–1092, 1997.

[20] H. C. Elman, D. J. Silvester, 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.

[21] J.W. Pearson, A.J. Wathen. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numer. Linear Alg. Appl.*, 19:816–829, 2012.

[22] T. Rees, A.J. Wathen. Preconditioning iterative methods for the optimal control of the Stokes equations. *SIAM J. Sci. Comput.*, 33:2903–2926, 2011.

[23] M. A. Heroux, J. M. Willenbring. Trilinos users guide. Technical Report SAND2003-2952, Sandia National Laboratories, 2003.

[24] K.-A. Mardal, R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numer. Math.*, 98:305–327, 2004.

[25] O. Axelsson, P. Boyanova, M. Kronbichler, M. Neytcheva, X.Wu. Numerical and computational efficiency of solvers for two-phase problems. *Comput. Math. Appl.*, 65:301–314, 2013.

[26] W. Zulehner. Nonstandard norms and robust estimates for saddle-point problems. *SIAM J. Matrix Anal. Appl.*, 32:536–560, 2011.

[27] W. Zulehner. Efficient solvers for saddle point problems with applications to PDE-constrained optimization. In *Lecture Notes in Applied and Computational Mechanics*, volume 66, chapter Advanced Finite Element Methods and Applications, pages 197–216. Springer, 2013.