Received: 17 October 2017 | Revised: 1 August 2018 | Accepted: 7 September 2018

DOI: 10.1002/nla.2218

RESEARCH ARTICLE

WILEY

Preconditioning for boundary control problems in incompressible fluid dynamics

Gennadij Heidel[1] | Andy Wathen[2]

[1]Fachbereich IV - Mathematik, Universität Trier, Trier, Germany

[2]Mathematical Institute, University of Oxford, Oxford, UK

Correspondence
Gennadij Heidel, Fachbereich IV - Mathematik, Universität Trier, 54286 Trier, Germany.
Email: heidel@uni-trier.de

Funding information
German Academic Scholarship Foundation; DFG Research Training Group 2126

Summary

PDE-constrained optimization problems arise in many physical applications, prominently in incompressible fluid dynamics. In recent research, efficient solvers for optimization problems governed by the Stokes and Navier–Stokes equations have been developed, which are mostly designed for distributed control. Our work closes a gap by showing the effectiveness of an appropriately modified preconditioner to the case of Stokes boundary control. We also discuss the applicability of an analogous preconditioner for Navier–Stokes boundary control and provide some numerical results.

KEYWORDS

Navier–Stokes control, Oseen system, PDE-constrained optimization, preconditioning, saddle-point problems, Stokes control

1 | INTRODUCTION

Optimal control problems were first introduced by Lions[1] and have recently attracted considerable research interest in applied mathematics, in terms of both theory[2] and computation.[3] An important area where problems of this kind naturally arise is that of fluid dynamics. An important problem within this field is that of systems governed by the Navier–Stokes equations and their limiting case for very viscous flow, the Stokes equations.

For both steady-state Stokes and Navier–Stokes forward problems, efficient solvers for finite element discretizations have been developed; see the works of Wathen et al.[4,5] and Axelsson et al.[6] for the Stokes case and see the works of Elman et al.[7–10] for the Navier–Stokes case. These methods are built on Krylov subspace iterations with specially designed preconditioners to achieve rapid convergence. An overview of state-of-the-art methods is given in the recent text by Elman et al.[11] Other recent surveys of preconditioning methods in the literature include the works of Axelsson,[12] Saad,[13] Vassilevski,[14] Olshanskii et al.,[15] and Málek et al.[16]

More recently, solvers for control problems governed by the Stokes and Navier–Stokes equations based on these forward solvers have been developed. Our work is largely based on a preconditioner developed by Rees et al.[17] for distributed control problems of the Stokes equations. A new development in PDE-constrained optimization is parameter-robust preconditioning, that is, methods whose quality does not depend on regularization parameters in the cost function. The notion was introduced by Schöberl et al.[18,19]; more recently, Pearson et al. have developed remarkable results for distributed Poisson control.[20] This framework has been extended to Poisson boundary control and heat control,[21] distributed Stokes control in the steady-state[22] and the time-dependent case,[23] and distributed steady-state Navier–Stokes control.[24] See also the work of Axelsson et al.[25] on Poisson and convection–diffusion problems, where a fully robust preconditioner with spectral bound 2 has been presented.

The contribution of this paper consists of the generalization of the Rees–Wathen preconditioner to the case of control only at the inflow boundary and the use of the Navier–Stokes equations. This confronts us with several challenges not addressed in existing literature. The first of them is related to the presence of control variables only defined within a small subregion of the domain, as this precludes the use of preconditioning techniques for distributed control, which reduce the sensitivity of the resulting preconditioner to the magnitude of the regularization parameter. Instead, we rely on the low-rank nature of the regularization term. The second main challenge is associated with the nonsymmetry of the linearized Navier–Stokes operator and restrictions on suitable approximations for subblocks of the system matrix. In the symmetric case, Rees et al. consider an Uzawa iteration that satisfies these restrictions. Here, we consider a Bramble et al.[26] version of the Uzawa iteration that can be suitable in the nonsymmetric case. However, the definition of the subblocks must be reorganized to avoid possible indefinite issues within subcomponents, which arise naturally from the fact that the control is defined on the inflow boundary.

This paper is organized as follows. In Section 2, we introduce the Stokes boundary control problem and discuss its discretization and optimality conditions. In Section 3, we present a Rees–Wathen-type preconditioner for the optimality system of this problem. In Section 4, we introduce the analogous Navier–Stokes boundary control problem, discuss the nonlinear iteration employed, and provide the discretization and optimality conditions for the linearized problems. In Section 5, we discuss spectral properties of the linearized and discretized Navier–Stokes problem and develop a Rees–Wathen-type preconditioner for this problem. In Section 6, we present numerical results to highlight the performance of our preconditioners, and in Section 7, we make some concluding remarks and discuss possible extensions of this work.

2 | THE STOKES BOUNDARY CONTROL PROBLEM

Let Ω be a channel domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, and let $\widehat{\vec{v}}$ and $\widehat{p}$ be functions on Ω that define a desired velocity and pressure profile. While it is reasonable to expect that $\widehat{\vec{v}}$ is always prescribed in relevant application cases, one can forgo $\widehat{p}$ if it seems reasonable (cf. the work of Pearson[22]); here, we will use a simple regularization $\widehat{p} = 0$ with a reasonably low α. We want to manipulate the inflow of the channel $\partial\Omega_{\text{in}}$ in such a way that the Stokes flow profile is as close as possible to $(\widehat{\vec{v}}, \widehat{p})$; in a mathematical sense, applying a force to the boundary of the channel is the same as imposing Neumann boundary conditions; this gives us a Neumann control problem in a natural way.

This may be formulated as minimizing a least-squares cost functional subject to the Stokes equations as the constraint, that is,

$$\min_{\vec{v},p,\vec{u}} \frac{1}{2}\left\|\vec{v}-\widehat{\vec{v}}\right\|^2_{L^2(\Omega)^2} + \frac{\alpha}{2}\|p-\hat{p}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|\vec{u}\|_{L^2(\partial\Omega_{\text{in}})^2}$$

$$\text{such that} \begin{cases} -\nabla^2\vec{v} + \nabla p &= \vec{0} \quad \text{in } \Omega, \\ \nabla\cdot\vec{v} &= 0 \quad \text{in } \Omega, \\ \vec{v} &= \vec{0} \quad \text{on } \partial\Omega_D, \\ \frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{u} \quad \text{on } \partial\Omega_{\text{in}}, \\ \frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{0} \quad \text{on } \partial\Omega_{\text{out}}. \end{cases} \tag{1}$$

Here, the Dirichlet boundary Ω_D represents the walls of the channel where the flow is equal to zero, and a zero-stress boundary condition is chosen on the outflow $\partial\Omega_{\text{out}}$; this ensures that the fluid leaves the channel domain freely without a force applied to it. The desired pressure is typically normalized to $\widehat{p} \equiv 0$. The positive regularization parameters α and β are chosen a priori; as long as they are not too large, the main focus of the cost function lies on the velocity term, which characterizes the desired flow profile. The outward normal unit vector is denoted by $\vec{n}$, as usual.

Problem formulations with a tracking-type objective function as in (1) are an important class of optimal control problems in the literature; see the work of Tröltzsch.[2] Other important classes of boundary control problems in fluid dynamics, which are beyond the scope of this paper, are minimum vorticity control and Dirichlet boundary control (e.g., the work of Heinkenschloss[27]). For formulations of Stokes control problems, see also the work of Axelsson et al.[6]

There are two different approaches to the discretization of this problem. We can either discretize first and then find optimality conditions for the discretized system, or we can find optimality conditions for the infinite-dimensional problem (1) using the formal Lagrange technique[1,2] and, then, discretize the obtained equations. Because the Stokes equations are self-adjoint, both approaches lead to the same discrete optimality conditions, as long as we use an adjoint-consistent discretization. Therefore, we only consider the discretize-then-optimize approach here.

Let $\{\vec{\varphi}_j\}_{j=1}^{n_v+n_\partial}$ and $\{\psi_k\}_{k=1}^{n_p}$ be finite element bases that form a stable mixed finite element discretization for the Stokes equations (see, e.g., chapter 3 of the work of Elman et al.[11] for further details). Note that, in general, we would also need basis functions $\vec{\varphi}_{n_v+1}, \ldots, \vec{\varphi}_{n_v+n_\partial}$, which interpolate the Dirichlet boundary data; this need not be considered here because the Dirichlet boundary data in (1) is identically zero. Because the control $\vec{u}$ is another unknown, we also choose a finite basis element basis $\{\vec{\chi}_l\}_{l=1}^{n_u}$, which lives on the inflow boundary of the channel. Because the boundary of a d-dimensional domain Ω is a manifold of dimension $d - 1$, the control space is canonically isomorphic to a finite element space on a domain of dimension $d - 1$; we will not distinguish between this and a boundary finite element space. Let $\vec{v}_h = \sum_{j=1}^{n_v} \boldsymbol{v}_j \vec{\varphi}_j$, $p_h = \sum_{k=1}^{n_p} \boldsymbol{p}_k \psi_k$, and $\vec{u}_h = \sum_{l=1}^{n_u} \boldsymbol{u}_l \vec{\chi}_l$ be finite-dimensional approximations to $\vec{v}, p$, and $\vec{u}$, respectively. Then, the discretized Stokes equations are given by

$$\begin{bmatrix} \boldsymbol{A} & B^T \\ B & O \end{bmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{Q}}\mathbf{u} \\ \mathbf{0} \end{pmatrix}, \tag{2}$$

where $\mathbf{v}$, $\mathbf{p}$, and $\mathbf{u}$ are the coefficient vectors in the expansions of $\vec{v}_h, p_h$, and $\vec{u}_h$, respectively, and the matrices are given by $\boldsymbol{A} = [\int_\Omega \nabla\vec{\varphi}_j : \nabla\vec{\varphi}_i]$, $B = [\int_\Omega \psi_k \nabla \cdot \vec{\varphi}_j]$ and $\widehat{\boldsymbol{Q}} = [\int_{\partial\Omega_{\text{in}}} \vec{\varphi}_i \cdot \vec{\chi}_l]$. Note that we use the convention to denote Gramian matrices obtained from vector-valued functions by bold letters.

The discretized cost functional of (1) is given by

$$\min_{\mathbf{v},\mathbf{p},\mathbf{u}} \frac{1}{2}\langle \boldsymbol{Q}_{\vec{v}}\mathbf{v}, \mathbf{v}\rangle - \langle \mathbf{b}, \mathbf{v}\rangle + \frac{\alpha}{2}\langle Q_p\mathbf{p}, \mathbf{p}\rangle - \alpha\langle \mathbf{d}, \mathbf{p}\rangle + \frac{\beta}{2}\langle \boldsymbol{Q}_{\vec{u}}\mathbf{u}, \mathbf{u}\rangle, \tag{3}$$

where the mass matrices are given by $\boldsymbol{Q}_{\vec{v}} = [\int_\Omega \vec{\varphi}_i \cdot \vec{\varphi}_j]$, $Q_p = [\int_\Omega \psi_i \psi_j]$ and $\boldsymbol{Q}_{\vec{u}} = [\int_{\partial\Omega_{\text{in}}} \vec{\chi}_i \cdot \vec{\chi}_j]$, and the vectors $\mathbf{b} = (\int_\Omega \vec{\varphi}_j \cdot \widehat{\vec{v}})$ and $\mathbf{d} = (\int_\Omega \psi_k \cdot \widehat{p})$ contain the linear terms.

In practice, it is convenient to choose the bases $\{\vec{\varphi}_j\}$ for $\vec{v}_h$ and $\{\vec{\chi}_l\}$ for $\vec{u}_h$ such that, for every l, there exists some $j(l)$ with $\vec{\chi}_l = \vec{\varphi}_{j(l)}|_{\partial\Omega_{\text{in}}}$. Then, the jth row of the mixed mass matrix $\widehat{\boldsymbol{Q}}$ will be equal to the lth row of $\boldsymbol{Q}_{\vec{u}}$ if $j = j(l)$ for some l, and zero otherwise. We will call this control discretization control consistent. Throughout this paper, we will use a control-consistent Taylor–Hood approximation,[28] that is, a $\boldsymbol{Q}_2$ approximation for the velocity (and hence for the control) and a $\boldsymbol{Q}_1$ approximation for the pressure.

As is usual, we approximate all velocity space components using a single scalar finite element space, which is given by a basis $\{\varphi_j\}$. Then, in two dimensions $\boldsymbol{Q}_{\vec{v}} = \text{blkdiag}(Q_v, Q_v)$, where $Q_v = [\int_\Omega \varphi_i \varphi_j]$, and analogously $\boldsymbol{Q}_{\vec{u}} = \text{blkdiag}(Q_u, Q_u)$ and $\boldsymbol{A} = \text{blkdiag}(A, A)$. The extension to three dimensions is obvious.

If we introduce adjoint variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, then the Karush–Kuhn–Tucker (KKT) conditions for the discretized optimization problem are given by

$$\begin{bmatrix} \boldsymbol{Q}_{\vec{v}} & O & O & \boldsymbol{A} & B^T \\ O & \alpha Q_p & O & B & O \\ O & O & \beta\boldsymbol{Q}_{\vec{u}} & -\widehat{\boldsymbol{Q}}^T & O \\ \boldsymbol{A} & B^T & -\widehat{\boldsymbol{Q}} & O & O \\ B & O & O & O & O \end{bmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \alpha\mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{4}$$

Note that the discrete cost function (3) is strictly convex; thus, the solution of (4) is guaranteed to be the global minimizer.

A well-known analytic solution of the Stokes equations on a channel domain is the Poiseuille flow; see the work of Elman et al.[11(p122)] On a simple square domain $\Omega = (-1, 1)^2$ where the left-hand boundary $\partial\Omega_{\text{in}} = \{-1\} \times (-1, 1)$ is the inflow and the right-hand boundary $\partial\Omega_{\text{out}} = \{1\} \times (-1, 1)$ is the outflow, the velocity solution is given by $\vec{v} = (1 - y^2, 0)^T$. This describes a straight horizontal movement with a parabolic flow profile whose maximum is in the middle of the channel. In our model problem, we want to restrict this flow profile to the upper half of the channel with no fluid movement in the lower half, that is, the desired velocity is given by

$$\hat{v}_x = \begin{cases} 4y - 4y^2, & \text{if } 0 \le y < 1, \\ 0, & \text{if } -1 < y < 0, \end{cases} \qquad \hat{v}_y = 0. \tag{5}$$

3 | PRECONDITIONING FOR THE STOKES PROBLEM

The KKT matrix in (4) is clearly of block saddle-point structure in the form

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}^T \\ \mathcal{B} & \mathcal{O} \end{bmatrix}$$

with the blocks $\mathcal{A} = \text{blkdiag}(\boldsymbol{Q}_{\vec{v}}, Q_p, \boldsymbol{Q}_{\vec{u}})$ and

$$\mathcal{B} = \begin{bmatrix} \boldsymbol{A} & B^T & -\widehat{Q} \\ B & O & O \end{bmatrix}.$$

For a comprehensive survey of numerical methods for such matrices, see the work of Benzi et al.[29]

We would like to work with preconditioned Krylov subspace methods. A well-known property of saddle-point matrices is their indefiniteness; therefore, the method of choice is the minimal residual (MINRES) iteration of Paige et al.[30] The crucial part of an efficient method is the right choice of the preconditioner. A block diagonal preconditioner is given by

$$\begin{bmatrix} \mathcal{A} & \mathcal{O} \\ \mathcal{O} & \mathcal{S} \end{bmatrix}, \tag{6}$$

where $\mathcal{S} = \mathcal{B}\mathcal{A}^{-1}\mathcal{B}^T$ is the Schur complement. This preconditioner is ideal in the sense that the preconditioned system will have only three distinct eigenvalues, namely, 1 and $(1\pm\sqrt{5})/2$; see the work of Murphy et al.[31] However, the practical application of this preconditioner requires the solution of systems with $\mathcal{A}$ and $\mathcal{S}$, which can be expected to be prohibitively expensive; therefore, positive definite approximations $\widetilde{\mathcal{A}} \approx \mathcal{A}$ and $\widetilde{\mathcal{S}} \approx \mathcal{S}$ are used. This gives good clustering of the eigenvalues as long as the approximations are spectrally close to the exact operators; see Theorem 4.7 in the work of Elman et al.[11] for a rigorous statement.

Here, we describe approximations for $\mathcal{A}$ and $\mathcal{S}$ developed by Rees et al.[17] for distributed Stokes control and justify their usefulness for boundary control. As shown in the work of Axelsson et al.,[6] there exists another type of preconditioner that does not require any approximation of a Schur complement and gives a fully robust spectral condition number bound equal to 2. Hence, it does not give an indefinite spectrum as block diagonal matrices do.

3.1 | Approximation of the (1,1) block

Because the (1,1) block has a block-diagonal structure itself, it is sufficient to approximate all of the three blocks separately. A result for a general mass matrix Q due to the work of Wathen[32] says that, for $D = \text{diag}(Q)$, the eigenvalues of $D^{-1}Q$ are bounded below and above by some constants θ and Θ, independent of mesh size, and these constants can be calculated explicitly as minimal and maximal eigenvalues of diagonally scaled element mass matrices. For a control-consistent Taylor–Hood approximation, they are given in Table 1.

Therefore, diagonal scaling could be used as a good preconditioner for $\mathcal{A}$. An even better result can by achieved by using a fixed number of steps of a Chebyshev semi-iteration,[33] which accelerates the convergence of a simple splitting method by substituting an iterate by a linear combination of all previous iterates. The optimal linear combination can be found from the eigenvalue bounds in Table 1 and the minimax property of the Chebyshev polynomials. The usefulness of the Chebyshev semi-iteration for problems with the mass matrix was shown by Wathen et al.[34]

To quantify this, we calculate the eigenvalues of $M_{\mathrm{C}}^{-1}Q$, where M_{C} is the preconditioner given by 20 Chebyshev steps with initial guess equal to zero (see Table 2). It shows that 20 steps are enough to get very good approximation for all mass matrices. For more numerical results; see table I in the work of Rees et al.[35]

For another diagonal approximation of the mass matrices using best Frobenius approximation of its inverse; see the work of Axelsson[12] and the references therein.

TABLE 1 Eigenvalue bounds for diagonally scaled mass matrices for a $\boldsymbol{Q}_2$-$\boldsymbol{Q}_1$ approximation (valid for any domain)

	θ	Θ
$\boldsymbol{Q}_{\vec{v}}$	1/4	25/16
Q_p	1/4	9/4
$\boldsymbol{Q}_{\vec{u}}$	1/2	5/4

TABLE 2 Minimal and maximal eigenvalues and condition numbers of $M_{\mathrm{C}}^{-1}Q$ with 20 iterations for various mass matrices

	$\lambda_{\min}$	$\lambda_{\max}$	κ
$\boldsymbol{Q}_{\vec{v}}$	0.999999912603445	1.000000087211418	1.000000235482498
Q_p	0.999998092651363	1.000001906960303	1.000004093538232
$\boldsymbol{Q}_{\vec{u}}$	0.999999999999775	1.000000000000209	1.000000000000467

3.2 | Approximation of the Schur complement

Now, we want an approximation for the Schur complement

$$\mathcal{S} = \mathcal{B}\mathcal{A}^{-1}\mathcal{B}^T = \begin{bmatrix} \boldsymbol{A} & B^T & -\widehat{Q} \\ B & O & O \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_{\vec{v}}^{-1} & O & O \\ O & \frac{1}{\alpha}Q_p^{-1} & O \\ O & O & \frac{1}{\beta}\boldsymbol{Q}_{\vec{u}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & B^T \\ B & O \\ -\widehat{Q}^T & O \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \boldsymbol{A} & B^T \\ B & O \end{bmatrix}}_{=:\mathcal{K}} \underbrace{\begin{bmatrix} \boldsymbol{Q}_{\vec{v}}^{-1} & O \\ O & \frac{1}{\alpha}Q_p^{-1} \end{bmatrix}}_{=:\mathcal{Q}^{-1}} \begin{bmatrix} \boldsymbol{A} & B^T \\ B & O \end{bmatrix} + \frac{1}{\beta} \underbrace{\begin{bmatrix} \widehat{Q}\boldsymbol{Q}_{\vec{u}}^{-1}\widehat{Q}^T & O \\ O & O \end{bmatrix}}_{=:\mathcal{L}}.$$

Therefore, $\mathcal{S} = \mathcal{K}\mathcal{Q}^{-1}\mathcal{K} + \frac{1}{\beta}\mathcal{L}$. Note that the matrix $\mathcal{K}$ is the same Stokes operator as in (2). This additive structure of the Schur complement makes exact preconditioning difficult. For problems of this kind, corollary 3.3 in the work of Rees et al.[36] suggests dropping the second term—and hence using the approximation $\widetilde{\mathcal{S}} = \mathcal{K}\mathcal{Q}^{-1}\mathcal{K}$—and presents eigenvalue bounds (depending on β) for a control problem governed by the Poisson equation. This is also the strategy used in the Rees–Wathen preconditioner. The intuitive reasoning behind this is that the first summand clearly carries more information in some sense—it contains the discrete Stokes operator, whereas the second summand consists only of mass matrices, which can be thought of as identity or natural inclusion operators in some finite element spaces. Therefore, if β is sufficiently large, (hence $1/\beta$ sufficiently small) one can hope that this gives a reasonable approximation. Rees et al.[17] show that this strategy applied to distributed control gives good results with respect to the grid size for $\beta \geq 10^{-4}$.

Ideally, we would like to find a regularization-robust preconditioner for our optimal control problem, that is, the quality of preconditioning should be independent of the regularization parameter β. Pearson et al.[20] have proved the existence of such preconditioners for distributed Poisson control problems. In the work of Pearson,[22] a preconditioner for distributed control of the Stokes equations is presented, which shows regularization-robust behavior in numerics. Unfortunately, the framework of the Pearson–Wathen preconditioner and of the derived preconditioner for the Stokes equations heavily relies on the fact that, in the case of distributed control, the control $\mathbf{u}$ is just a scalar multiple of the adjoint $\boldsymbol{\lambda}$, and thus, the optimality system can be reduced to a 4×4 block structure. This is clearly not applicable to our situation, as $\mathbf{u}$ and $\boldsymbol{\lambda}$ do not even have the same dimension. For a more detailed discussion of the influence of the parameter β, see the work of Axelsson et al.[6]

By construction, the Rees–Wathen preconditioner for Stokes control cannot be regularization robust. However, in our case, it can be proved to be "almost regularization robust" in the sense that $\operatorname{rank}(\mathcal{S}) = n_v + n_p \gg n_u = \operatorname{rank}(\frac{1}{\beta}\mathcal{L})$. Thus, all we lose by our choice of approximation is a low-rank perturbation. The analysis of symmetric rank-1-perturbations is due to the work of Wilkinson[37(p87)] and can be written in the form of the following lemma (here and in the rest of the paper, we assume the usual ordering $\lambda_n \leq \cdots \leq \lambda_1$ for the eigenvalues of a symmetric $n \times n$ matrix).

Lemma 1. (See theorem 8.1.8 in the work of Golub et al.[38])
Suppose $B = A + \tau\mathbf{c}\mathbf{c}^T$ where $A \in \mathbb{R}^{n \times n}$ is symmetric and $\tau \geq 0$. Then,

$$\lambda_i(B) \in [\lambda_i(A), \lambda_{i-1}(A)], \quad \textit{for } i = 2, \ldots, n.$$

By an inductive argument, this can be generalized to additive perturbations of arbitrary rank.

Theorem 1. *Suppose $B = A + L$ where $A, L \in \mathbb{R}^{n \times n}$ are symmetric, L is positive semidefinite, and $\operatorname{rank}(L) \leq m < n$. Then,*

$$\lambda_i(B) \in [\lambda_i(A), \lambda_{i-m}(A)], \quad \textit{for } i = m+1, \ldots, n.$$

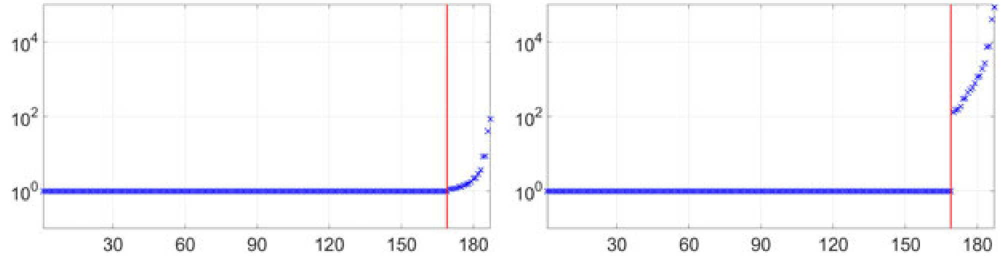


FIGURE 1 Eigenvalues of $\widetilde{\mathcal{S}}^{-1}\mathcal{S}$ for (left) $\beta = 10^{-3}$ and (right) $\beta = 10^{-6}$

Proof. For $m = 0$, there is nothing to prove. Let the statement be true for some $m \in \mathbb{N}_0$ and L be a symmetric positive semidefinite matrix with $\operatorname{rank}(L) \leq m + 1$. Then, $L = K + \tau_{m+1}\mathbf{c}_{m+1}\mathbf{c}_{m+1}^T$ can be written as

$$L = \underbrace{\sum_{j=1}^{m}\tau_j\mathbf{c}_j\mathbf{c}_j^T}_{=:K} + \tau_{m+1}\mathbf{c}_{m+1}\mathbf{c}_{m+1}^T$$

with $\tau_{m+1} \geq 0$. Then, $\operatorname{rank}(K) \leq m$, and by the induction hypothesis,

$$\lambda_i(A+K) \in [\lambda_i(A), \lambda_{i-m}(A)], \quad \text{for } i = m+1, \dots, n.$$

In addition, by Lemma 1,

$$\lambda_i(B) \in [\lambda_i(A+K), \lambda_{i-1}(A+K)], \quad \text{for } i = 2, \dots, n.$$

Combining these, we get

$$\lambda_i(B) \in [\lambda_i(A), \lambda_{i-m-1}(A)], \quad \text{for } i = m+2, \dots, n,$$

as desired. □

Therefore, if we drop the low-rank perturbation, we get a parameter-robust approximation for all but n_u eigenvalues. This is illustrated in Figure 1, where $n_v = 162$, $n_p = 25$, and $n_u = 18$. The red line indicates $\operatorname{rank}(\mathcal{K}) - \operatorname{rank}\mathcal{L} = n_v + n_p - n_u$; we observe a good clustering of all but the largest n_u eigenvalues around 1 even for a fairly small β; the nonclustered eigenvalues show a behavior dependent on β. Table 3 shows that it gives us both reasonable preconditioning for an exact application of the preconditioner, as well as for using iterative methods.

Now, we can hope that we will get good results if we can find an approximation for the action of $\widetilde{\mathcal{S}}^{-1} = \mathcal{K}^{-1}\mathcal{Q}\mathcal{K}^{-1}$. This needs two inverses of the discrete Stokes operator $\mathcal{K}$ and one multiply by $\mathcal{Q}$; the latter is obviously trivial. The approximation of the discrete Stokes operator requires some extra care here because we cannot simply use a well-known Stokes preconditioner such as that in the work of Silvester et al.[5] The issue that has been pointed out by Braess et al.[39] is the following: If we have a good preconditioner $\tilde{K}$ for a matrix K, then $\tilde{K}\tilde{K}^T$ is not necessarily a good preconditioner for KK^T. Indeed, we find that commonly used preconditioners for the Stokes operator fail in our situation. In their paper, Braess et al. also show sufficient conditions for a squared preconditioner to work.

Theorem 2. (Braess–Peisker conditions)
We consider the linear equation $K\mathbf{x} = \mathbf{b}$. Let $\tilde{K}_j$ be a sequence of invertible matrices, such that $\mathbf{x}^{(j)} := \tilde{K}_j^{-1}\mathbf{b}$ converges to the exact solution $\mathbf{x}$ in the sense that

$$\|\mathbf{x}^{(j)} - \mathbf{x}\|_2 \leq \eta_j\|\mathbf{x}\|_2$$

TABLE 3 Iteration counts for solving a Schur complement equation preconditioned by $\widetilde{\mathcal{S}}$, using direct solves (exact) and iterative methods (inexact) for applying the preconditioner

β	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Exact	17	19	27	43	79	141
Inexact	20	24	36	67	158	283

with $\eta_j \to 0$. Then, for j large enough, all $\mathbf{y} \neq \mathbf{0}$ *satisfy*

$$(1-\eta_j)^2 \leq \frac{\langle KK^T\mathbf{y},\mathbf{y}\rangle}{\langle \tilde{K}_j\tilde{K}_j^T\mathbf{y},\mathbf{y}\rangle} \leq (1+\eta_j)^2.$$

Note that the transposition of $\mathcal{K}$ is not strictly needed here because the discrete Stokes operator is symmetric, but in the Navier–Stokes case, we will have to deal with a nonsymmetric $\mathcal{K}$ in the next chapter; therefore, we include this case here.

Remember that we actually need to precondition $\mathcal{K}Q^{-1}\mathcal{K}^T$. Rees et al.[17] show that the matrix Q^{-1} simply introduces a scaling to the Braess–Peisker result in the form

$$c_*(1-\eta_j)^2 \leq \frac{\langle KQ^{-1}K^T\mathbf{x},\mathbf{x}\rangle}{\langle \tilde{K}_jQ^{-1}\tilde{K}_j^T\mathbf{x},\mathbf{x}\rangle} \leq (1+\eta_j)^2C_*,$$

with some constants $0 < c_* \leq 1$ and $1 \leq C_*$.

By Theorem 2, we have to approximate the Stokes equations by a contracting linear iteration. In fact, it is sufficient to use a convergent fixed-point iteration; standard preconditioners for Krylov solvers of the Navier–Stokes equations, such as the pressure convection diffusion method or the least squares commutator method, may not satisfy this property. The Rees–Wathen preconditioner uses the inexact Uzawa[40] iteration given by Algorithm 1 for a generic saddle-point problem

$$\begin{bmatrix} \boldsymbol{A} & B^T \\ B & O \end{bmatrix}\begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix},$$

with preconditioners $\widetilde{\boldsymbol{A}}$ for $\boldsymbol{A}$ and $\widetilde{S}$ for the Schur complement $S = B\boldsymbol{A}^{-1}B^T$. The convergence properties of the inexact Uzawa method have been studied in other works.[41–43] It can be shown to provide preconditioning for the Stokes operator independently of the grid size as long as the approximations $\widetilde{\boldsymbol{A}}$ and $\widetilde{S}$ are good enough; see subsection 2.4 of the work of Rees et al.[17] For the Laplacian operator $\boldsymbol{A}$, a fixed number of multigrid cycles[44,45] may be used; this is known to give a spectrally equivalent preconditioner, that is, if the action of a fixed number of multigrid cycles is given by a matrix $\widetilde{\boldsymbol{A}}$, then bounds δ and Δ independent of the grid size h exist such that $0 < \delta \leq \lambda(\widetilde{\boldsymbol{A}}^{-1}\boldsymbol{A}) \leq \Delta$; see section 2.5 of the work of Elman et al.[11] The Schur complement of the discrete Stokes operator is spectrally equivalent to the pressure mass matrix Q_p; see theorem 3.29 in the work of Elman et al.[11]; hence, a fixed number of Chebyshev steps can be used here, as discussed in the previous subsection.

```
Algorithm 1. (Inexact Uzawa)
Choose $\sigma, \tau > 0$
Choose $\mathbf{v}^{(0)}$, $\mathbf{p}^{(0)}$
for $k = 0$ until convergence do
  Solve $\widetilde{\boldsymbol{A}}\delta\mathbf{v}^{(k)} = \mathbf{f} - \boldsymbol{A}\mathbf{v}^{(k)} - B^T\mathbf{p}^{(k)}$
  $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \sigma\boldsymbol{\delta}\mathbf{v}^{(k)}$
  Solve $\frac{1}{\tau}\widetilde{S}\delta\mathbf{p}^{(k)} = B\mathbf{v}^{(k+1)} - \mathbf{g}$
  $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \tau\boldsymbol{\delta}\mathbf{p}^{(k)}$
end for
```

4 | THE NAVIER–STOKES BOUNDARY CONTROL PROBLEM

We would like to extend our treatment to the more general case of Navier–Stokes control. The analogue to the Stokes problem is given by

$$\min_{\vec{v},p,\vec{u}} \frac{1}{2}\left\|\vec{v}-\hat{\vec{v}}\right\|^2_{L^2(\Omega)^2} + \frac{\alpha}{2}\|p-\hat{p}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|\vec{u}\|^2_{L^2(\partial\Omega_{\text{in}})^2}$$

$$\text{such that}\begin{cases} -\nu\nabla^2\vec{v}+\vec{v}\cdot\nabla\vec{v}+\nabla p &= \vec{0} \quad \text{in } \Omega, \\ \nabla\cdot\vec{v} &= 0 \quad \text{in } \Omega, \\ \vec{v} &= \vec{0} \quad \text{on } \partial\Omega_D, \\ \nu\frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{u} \quad \text{on } \partial\Omega_{\text{in}}, \\ \nu\frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{0} \quad \text{on } \partial\Omega_{\text{out}}, \end{cases} \tag{7}$$

where the only new terms are the viscosity parameter ν and the nonlinear convection term $\vec{v} \cdot \nabla\vec{v}$. A Navier–Stokes flow is usually characterized by the Reynolds number $\mathcal{R} \sim 1/\nu$; in fact, for our channel domain, $\mathcal{R} = 1/\nu$. To solve (7), we need to linearize the constraint, that is, the convection term. This involves computing solutions $(\vec{v}_k, p_k)$ to a sequence of linearized problems starting from some initial guess*$(\vec{v}_0, p_0)$. The simplest way to do this is by a fixed-point iteration where we replace the nonlinear convection term $\vec{v} \cdot \nabla\vec{v}$ by its linearized version $\vec{v}_h \cdot \nabla\vec{v}$ with $\vec{v}_h$ being the velocity solution from the previous iterate. This is referred to as the Picard linearization of the Navier–Stokes equations, and the resulting linear PDE is called the Oseen equation. Hence, to find a solution of (7), we solve a sequence of Oseen control problems

$$\min_{\vec{v},p,\vec{u}} \frac{1}{2}\left\|\vec{v}-\hat{\vec{v}}\right\|^2_{L^2(\Omega)^2} + \frac{\alpha}{2}\left\|p-\hat{p}\right\|^2_{L^2(\Omega)} + \frac{\beta}{2}\left\|\vec{u}\right\|^2_{L^2(\partial\Omega_{\text{in}})^2}$$

$$\text{such that}\begin{cases} \text{s. t.} -\nu\nabla^2\vec{v} + \vec{v}_h\cdot\nabla\vec{v} + \nabla p &= \vec{0} \quad \text{in } \Omega,\\ \nabla\cdot\vec{v} &= 0 \quad \text{in } \Omega,\\ \vec{v} &= \vec{0} \quad \text{on } \partial\Omega_D,\\ \nu\frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{u} \quad \text{on } \partial\Omega_{\text{in}},\\ \nu\frac{\partial\vec{v}}{\partial n} - p\vec{n} &= \vec{0} \quad \text{on } \partial\Omega_{\text{out}}. \end{cases} \tag{8}$$

Pošta et al.[46] suggest augmenting the cost function of (8) by the term $-\int_\Omega(\vec{v}\cdot\nabla\vec{v}_h)\cdot\vec{\lambda}_h$, where $\vec{\lambda}_h$ denotes the continuous version of the Lagrange multiplier $\boldsymbol{\lambda}$ from the previous iteration†; this approach is used in the work of Pearson.[24] Then, convergence of a distributed control problem can be proved; see the work of Pošta et al.[46] for details. The practical advantage lies in a reduced number of iterations for high Reynolds numbers because it introduces a fixed-point iteration not only in the constraint but also in the objective function. Our treatment is restricted to relatively low Reynolds numbers; we will find that the Picard iteration with the system (8) gives satisfactory results in this case; thus, we will not consider this augmentation.

The linear convection operator $\vec{v}_h \cdot \nabla\vec{v}$ is discretized in the two-dimensional finite element space by $\boldsymbol{N} = \text{blkdiag}(N, N)$ with $N = [\int_\Omega(\vec{v}_h\cdot\nabla\varphi_j)\varphi_i]$. If we define the vector convection–diffusion operator $\boldsymbol{F} = \nu\boldsymbol{A} + \boldsymbol{N}$, then the discretized Oseen equations are given by

$$\begin{bmatrix} \boldsymbol{F} & B^T \\ B & O \end{bmatrix}\begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{Q}}\mathbf{u} \\ \mathbf{0} \end{pmatrix}. \tag{9}$$

The cost functional remains unchanged from (1) to (8); hence, the discrete cost function is also given by (3). Applying the discretize-then-optimize approach‡ gives the Oseen KKT system

$$\begin{bmatrix} \boldsymbol{Q}_{\vec{v}} & O & O & \boldsymbol{F}^T & B^T \\ O & \alpha Q_p & O & B & O \\ O & O & \beta\boldsymbol{Q}_{\vec{u}} & -\hat{\boldsymbol{Q}}^T & O \\ \boldsymbol{F} & B^T & -\hat{\boldsymbol{Q}} & O & O \\ B & O & O & O & O \end{bmatrix}\begin{pmatrix} \mathbf{v} \\ \mathbf{p} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \alpha\mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{10}$$

Note that, here, $\boldsymbol{F} \neq \boldsymbol{F}^T$ because the convection matrix $\boldsymbol{N}$ is nonsymmetric.

5 | PRECONDITIONING FOR THE NAVIER–STOKES PROBLEM

Due to the similarity of the KKT system for the Oseen problem to the one for the Stokes problem (4), our preconditioner will be based on the Rees–Wathen method discussed in the previous section. Indeed, the (1,1) blocks of both systems are identical, namely, the block-diagonal matrix blkdiag$(\boldsymbol{Q}_{\vec{v}}, Q_p, \boldsymbol{Q}_{\vec{u}})$, so there is no additional work needed here.

*We will choose $(\vec{v}_0, p_0)$ to be the solution of the corresponding Stokes problem (1).

†This is motivated by viewing the Picard iteration as a sequential quadratic programming-type iteration for (7).

‡The alternative optimize-then-discretize method does not necessarily result in a symmetric matrix; see, for example, the work of Pearson et al.[47] and section 6.3 of the work of Rees[48] for the convection diffusion equation. We will not consider it here.

The main difference is the operator $\boldsymbol{F}$ instead of $\boldsymbol{A}$, which is increasingly nonsymmetric for higher Reynolds numbers. If we use the Schur complement approximation dropping the low-rank perturbation $\frac{1}{\beta}\mathcal{L}$, namely,

$$\mathcal{S} \approx \widetilde{\mathcal{S}} = \underbrace{\begin{bmatrix} \boldsymbol{F} & B^T \\ B & O \end{bmatrix}}_{=:\mathcal{K}} \begin{bmatrix} \boldsymbol{Q}_{\vec{v}}^{-1} & O \\ O & \frac{1}{\alpha} Q_p^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}^T & B^T \\ B & O \end{bmatrix},$$

we need linear approximations for $\mathcal{K}$ and $\mathcal{K}^T$ that satisfy the Braess–Peisker conditions in Theorem 2. Note that we cannot use Krylov subspace methods such as the generalized MINRES (GMRES[49]) method because they are nonlinear and hence unsuitable as preconditioners for MINRES; that is, a fixed number of iterations of a Krylov method does not define a linear mapping. A candidate for such is the Uzawa-type iteration for nonsymmetric systems introduced by Bramble et al.[26] It may be written in the same way as Algorithm 1; the subtle difference here is however that $\widetilde{\boldsymbol{A}}$ is not a preconditioner for $\boldsymbol{F}$ but for the symmetric part $\boldsymbol{F}_S = \frac{1}{2}(\boldsymbol{F} + \boldsymbol{F}^T)$ with eigenvalue bounds $1 \leq \lambda(\widetilde{\boldsymbol{A}}^{-1}\boldsymbol{F}_S) \leq \Delta$. Similarly, $\widetilde{S}$ is a preconditioner for the symmetric part of the Schur complement $S_S = B\boldsymbol{F}_S^{-1}B^T$ with eigenvalue bounds $\gamma \leq \lambda(\widetilde{S}^{-1}S_S) \leq 1$. Note that, for the given approximations, the factor 1 in both inequalities can be achieved by a scaling. Then, Algorithm 1 will converge if the step-size parameters δ and τ are small enough; for a rigorous statement, see theorem 3.1 in the work of Bramble et al.[26]

There exist efficient solvers for the Navier–Stokes equations such as the pressure convection–diffusion preconditioner[8,9] and the least-squares commutator preconditioner.[7,10] However, these preconditioners do not satisfy the Braess–Peisker conditions and we do not know how they can be symmetrized efficiently to embed them in an Uzawa-type iteration; thus, we will not use them here.

The properties of $\mathcal{K}$ depend on the discrete convection operator $\boldsymbol{N}$, and the lower the viscosity parameter ν, the more so. Therefore, we will study some spectral properties of $\boldsymbol{N}$ in the next subsection and, then, construct a preconditioner taking them into account.

5.1 | Matrix properties

Due to the block-diagonal structure of the vector-convection operator $\boldsymbol{N}$, we can restrict our discussion to the scalar case without loss of generality.

We consider a continuous bilinear form

$$c(u, v) := \int_\Omega (\vec{w} \cdot \nabla u)v, \tag{11}$$

where $\vec{w}$ is a given wind with $\nabla \cdot \vec{w} = 0$; this is no restriction because, in the Oseen system, the wind is the solution of a (Navier–)Stokes boundary value problem. The bilinear form $c(\cdot, \cdot)$ is associated with the convection operator, which is usually thought of as being skew self-adjoint. However, in general there, will be some self-adjoint perturbation. We want to find an explicit expression for it. Application of the divergence theorem to (11) gives

$$\begin{aligned} c(u, v) = -\int_\Omega (v\vec{w}) \cdot \nabla u = -\int_\Omega \nabla \cdot (v\vec{w})u + \int_{\partial\Omega_N} uv \ \vec{w} \cdot \vec{n} \\ = -\int_\Omega \big((v\nabla \cdot \vec{w})u + (\vec{w} \cdot \nabla v)u\big) + \int_{\partial\Omega} uv \ \vec{w} \cdot \vec{n} \\ = -\int_\Omega (\vec{w} \cdot \nabla v)u + \int_{\partial\Omega_N} uv \ \vec{w} \cdot \vec{n}, \end{aligned}$$

where we used the product rule and the fact that $\vec{w}$ is divergence free. The last equality uses the assumption that $\vec{w} \cdot \vec{n} = 0$ on $\partial\Omega_D$, that is, there is no inflow or outflow on the Dirichlet boundary. This is consistent with our channel domain where the Dirichlet boundary represents the channel walls. Thus, the self-adjoint part of $c(\cdot, \cdot)$ is given by

$$h(u, v) := \frac{1}{2}\left(c(u, v) + c(v, u)\right) = \frac{1}{2}\int_{\partial\Omega_N} uv \ \vec{w} \cdot \vec{n}. \tag{12}$$

It is important to understand the consequences of (12) in our finite element framework. Consider the basis $\{\varphi_j\}$ for the finite element space. Then, we can express the symmetric part of the scalar convection matrix $N_S = \frac{1}{2}(N + N^T)$ in the form

$$N_S = \left[n_{ij}^S\right], \qquad n_{ij}^S = \int_{\partial\Omega_N} \varphi_i\varphi_j \vec{w} \cdot \vec{n},$$

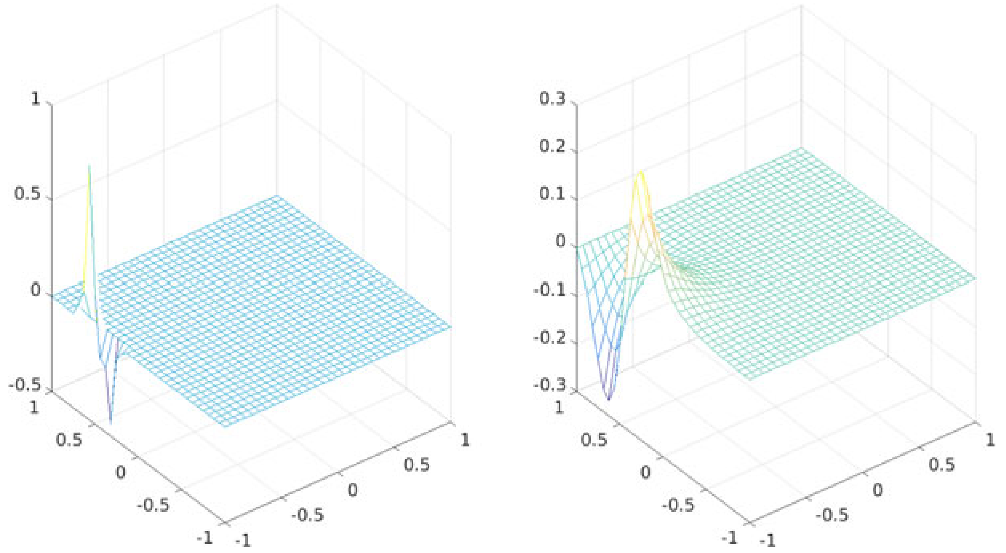


FIGURE 2 Eigenvectors of (left) N_S corresponding to $\lambda = -0.0314$ and of (right) $F_S = N_S + \frac{\nu}{2}A$ corresponding to $\lambda = -0.0020$ with $\nu = 1/10$

and the self-adjoint part of $c(\cdot,\cdot)$, applied to a function $v_h \in \operatorname{span}\{\varphi_j\}$, becomes

$$h(v_h, v_h) = \frac{1}{2}\int_{\partial\Omega_N} v_h^2 \vec{w}\cdot\vec{n} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{v}_i\mathbf{v}_j \int_{\partial\Omega_N} \varphi_i\varphi_j\,\vec{w}\cdot\vec{n}. = \frac{1}{2}\mathbf{v}^T N_S \mathbf{v}. \tag{13}$$

In the discrete bilinear form $\mathbf{v}^T N\mathbf{v}$, the contribution of $\mathbf{v}^T N_S\mathbf{v}$ will clearly dominate if the coordinate vector $\mathbf{v}$ (and hence the associated function v_h) is relatively large (in modulus) on the Neumann boundary and small everywhere else. From (13), it is clear that the sign of the contribution of $\mathbf{v}^T N_S\mathbf{v}$ is determined by the sign of $\vec{w}\cdot\vec{n}$. Recall that $\vec{n}$ is the outward normal vector on $\partial\Omega$; this motivates the definition of

$$\begin{aligned}
\text{the outflow boundary} \quad \partial\Omega_+ &:= \left\{x \in \partial\Omega_N \,\middle|\, \vec{w}\cdot\vec{n} > 0\right\},\\
\text{the characteristic boundary} \quad \partial\Omega_0 &:= \left\{x \in \partial\Omega_N \,\middle|\, \vec{w}\cdot\vec{n} = 0\right\},\\
\text{and the inflow boundary} \quad \partial\Omega_- &:= \left\{x \in \partial\Omega_N \,\middle|\, \vec{w}\cdot\vec{n} < 0\right\}.
\end{aligned}$$

Note that these definitions do not necessarily coincide with the inflow $\partial\Omega_{\text{in}}$ and the outflow $\partial\Omega_{\text{in}}$ of the channel; the latter are concepts based only on the geometry of the channel, whereas the former depends on the actual flow profile. The quadratic form for $\mathbf{v}$ in (13) is a scaled Rayleigh quotient (especially, it has the same sign as the Rayleigh quotient), and thus, it is related to the eigenvalues of N_S. Clearly, the contribution of $\partial\Omega_+$ is related to positive eigenvalues of N_S, and the contribution of $\partial\Omega_-$ is related to negative eigenvalues.

We will concentrate on $\partial\Omega_-$ and the associated negative eigenvalues of N_S. In our control problem, the wind is the solution in the previous iterate $\vec{v}_h$. Because we are controlling the inflow of the channel, we may assume that $\vec{v}_h$ is close enough to the desired flow profile $\hat{\vec{v}}$ such that $\vec{v}_h\cdot\vec{n} < 0$ if and only if $\hat{\vec{v}}\cdot\vec{n} < 0$. Thus, it is sufficient to consider the inflow boundary with respect to $\vec{v}$, which is known from (5). Hence, the inflow boundary is $\partial\Omega_- = \{-1\} \times (0, 1)$.

This issue is illustrated in Figure 2. The eigenvector of N_S corresponding to a negative eigenvalue lives mostly on the inflow boundary; the eigenvector of the scalar convection–diffusion operator $F_S := N_S + \frac{\nu}{2}A$ tends to have greater components in the interior of Ω but shows similar asymptotic behavior. In general, for decreasing ν, we may expect that the eigenvalues and eigenvectors of F_S converge to those of N_S.

5.2 | Permutational preconditioner

As discussed in the introduction to this section, the Bramble–Pasciak–Vassilev Uzawa-type iteration is guaranteed to converge only if the symmetric part of the (1,1) block of the saddle-point system is positive definite. As we have seen in the previous subsection, this is in general not the case for the discretized Oseen operator in our boundary control problem.

We have also seen both a theoretical explanation and numerical evidence that the negative eigenvalues of the symmetric part of the discrete convection–diffusion operator F_S are associated with modes that live mostly on the inflow boundary and are close to zero everywhere else. Let v_h be such an eigenmode of F_S. It has a representation in the finite element basis in the form

$$v_h = \sum_{j=1}^{n} \boldsymbol{v}_j \varphi_j.$$

Because we are employing Lagrange elements, each of the basis functions φ_j is equal to 1 at one node and equal to 0 at the others. This means that the coefficients $\boldsymbol{v}_j$ corresponding to basis functions that are zero on the inflow boundary will be relatively small in modulus; thus, in the matrix–vector product $F_S\mathbf{v}$, those columns of F_S will dominate, which correspond to the inflow nodes. On the other hand, the matrix F is a weighted sum of the discrete Laplacian A and the convection operator N. The former is well known to be positive semidefinite; the latter is skew self-adjoint with the exception of the boundary contribution (12) discussed in the previous subsection. Thus, if we remove the columns (and the corresponding rows, to retain symmetry) from F_S corresponding to the inflow components, we will eliminate most of the contribution of the negative modes and we might expect the resulting matrix to be positive definite.

We explore this issue for the viscosity parameter $\nu = 1/20$ and grid size $h = 10^{-5}$ in Table 4. The matrices F_S', F_S'', and $F_S^{(4)}$ represent F_S with every inflow node, every second inflow node, and every fourth inflow node eliminated, respectively, in the way described above. Clearly, in this case, it is sufficient to eliminate every fourth node to get a positive definite matrix; however, the condition number is increased in this case. Presumably, the elimination of a higher number of inflow nodes eliminates not only negative eigenvalues but also some positive ones that are close to zero, giving a better condition number.

When eliminating the negative eigenvalues, it is desirable to preserve as much of the structure of F_S as possible, that is, we want the positive eigenvalues to remain largely unchanged. Figure 3 shows evidence that this is, indeed, the case. The eigenvalues of F_S and F_S' are mostly indiscernible; we only observe a small perturbation for the smallest ones. This is consistent with the theory, as the smallest eigenvalues of F_S are presumably the ones most influenced by the boundary behavior of the convection operator. We may expect even better behavior if the number of deleted nodes is chosen to be smaller.

Now, we cannot simply delete rows and columns from the discrete Oseen operator as this would give us a different problem with a different solution. Instead, we need to build a preconditioner that works around this issue. The idea of removing certain eigenspaces of a matrix has been studied in the literature; a common technique is known as deflation. It is based on projecting the eigenspaces of the system matrix into a space where the "undesired" eigenvalues are equal to zero; see, for example, the work of Frank et al.[50] for details. However, all these methods produce a singular system matrix, which is not a problem for Krylov subspace methods such as CG—they will still converge as long as the problem is consistent—but renders usual stationary iterations used in preconditioning infeasible. Therefore, deflation ideas cannot be used in our preconditioner in the usual way.

TABLE 4 Eigenvalue structures and condition numbers of different node elimination schemes

	F_S	F_S'	F_S''	$F_S^{(4)}$
# (negative eigenvalues)	2	0	0	0
κ	$2.5261 \cdot 10^3$	$1.8061 \cdot 10^3$	$1.8178 \cdot 10^3$	$4.9132 \cdot 10^3$

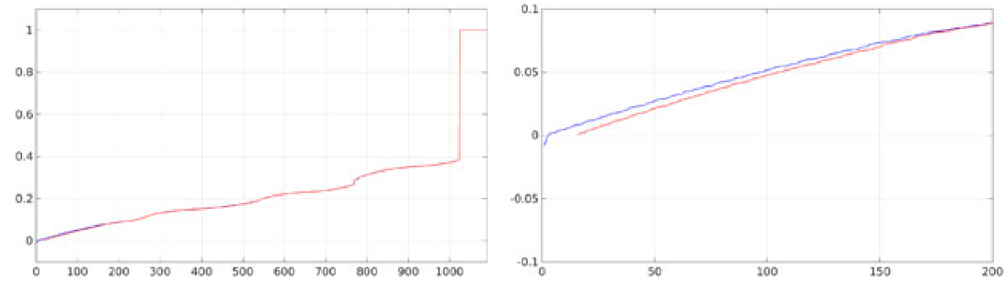


FIGURE 3 Eigenvalues of (blue) F_S and (red) F_S' (plotted as continuous spectra) with a zoom in the lower left corner

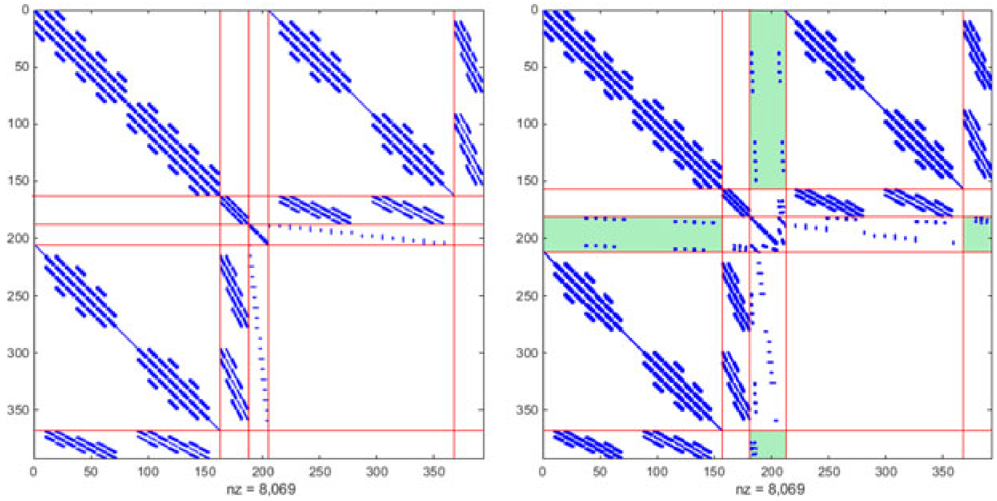


FIGURE 4 MATLAB `spy` plots of the Oseen Karush–Kuhn–Tucker matrix (left) without and (right) with permutation. The blocks with extra fill introduced by the permutation are marked

Instead, we will use an idea based on permutation. The negative eigenvalues of F_S pose a problem inside the Uzawa-type iteration, but if we can move them to another part of our preconditioner, we might use the Uzawa-type iteration for the remaining positive definite problem and take care of the negative eigenvalues in some other way. We take the following approach: We change the order of rows and columns in the system matrix of the discrete Oseen optimality conditions. If $\vec{\varphi}_j$ is a basis function that is equal to one at an inflow node, we would like to remove it so $\boldsymbol{a}_{jj}$ is moved to the (3,3) block of the KKT matrix. If the original system (10) is given by $\mathcal{A}\mathbf{x} = \mathbf{c}$, this can be thought of as solving the system

$$\mathcal{P}\mathcal{A}\mathcal{P}^T\mathbf{y} = \mathcal{P}\mathbf{c} \tag{14}$$

with a permutation matrix $\mathcal{P}$. The system (14) is clearly symmetric and equivalent to the original problem with $\mathbf{x} = \mathcal{P}^T\mathbf{y}$. This permutation of $\mathcal{A}$ is illustrated in Figure 4.

Aside from augmenting the (3,3) block of the Oseen system, the permutation also changes the sparsity structure: It introduces some fill in blocks that are equal to zero in the original system—note that there is no fill in the system matrix as a whole; it is introduced only in some blocks by filling them with nonzeros from other blocks. We do not know how to deal with these efficiently; therefore, in the computational preconditioner, they are set to zero. This can be justified heuristically from two points of view. First, this is a variation of the idea that incomplete factorizations are based on: The preconditioner is enforced to have the same sparsity structure as the system matrix; the use of incomplete factorizations is justified by both theoretical and numerical results. Second, the fill created by the permutation can be seen as a low-rank perturbation of a permuted system without fill; Theorem 1 suggests that we do not lose much information about almost all eigenvalues by dropping it. Therefore, the spectrum of the system matrix remains largely unchanged, and we can expect a small number of additional iterations in the outer MINRES method.

Now, the Schur complement of the permuted system (14) can be approximated in the same way as discussed in the previous section: We approximate the inverses of the Oseen operator and its transpose by inexact Uzawa-type iterations and multiply by a block-diagonal mass matrix, as discussed in Section 3.2. Because the permutation is designed to remove the nodes associated with negative eigenvalues of $\boldsymbol{F}_S$, as discussed above, we are left with a positive definite system and are guaranteed to get convergence of the inner Uzawa-type iteration.

The approximation of the block-diagonal upper-left block requires more care here because the diagonal blocks are all not simple mass matrices any more. While the (2,2) block, which is the pressure mass matrix, remains unchanged, by construction, the (3,3) block is now much more complicated than just a simple mass matrix; recall that we use it to collect the boundary information corresponding to the inflow, which we removed from the other blocks. We do not know enough about the structure of this matrix to find a good iterative preconditioner. However, especially for fine grids, this will be a relatively small matrix, compared to the total number of degrees of freedom. On a two-dimensional domain, the number of grid points (and thus, the dimension of the Oseen KKT matrix) is of order $\mathcal{O}(h^{-2})$. On the other hand, the (3,3) block only contains entries related to boundary nodes; their number is of order $\mathcal{O}(h^{-1})$. On a three-dimensional domain, we would get analogous behavior with $\mathcal{O}(h^{-3})$ and $\mathcal{O}(h^{-2})$, respectively. This is illustrated in Table 5.

TABLE 5 Dimensions N of the discrete Oseen Karush–Kuhn–Tucker system and n of the $(3, 3)$ block of the permutational preconditioner with every second inflow node removed

l	2	3	4	5	6	7
N	128	392	1,352	5,000	19,208	75,272
n	14	26	50	98	194	386

Algorithm 2. (Permutational preconditioner)

```
    Require: Right-hand side $(\mathbf{v}^T, \mathbf{p}^T, \mathbf{u}^T, \boldsymbol{\lambda}^T, \boldsymbol{\mu}^T)^T$, block matrix $\mathcal{W} = \mathcal{P}\mathcal{A}\mathcal{P}^T$ as given by the right-hand side picture of Figure 4, with blocks $W_{ij}$.
    Output: Preconditioned iteration $(\hat{\mathbf{v}}^T, \hat{\mathbf{p}}^T, \hat{\mathbf{u}}^T, \hat{\boldsymbol{\lambda}}^T, \hat{\boldsymbol{\mu}}^T)^T = \mathcal{M}^{-1}(\mathbf{v}^T, \mathbf{p}^T, \mathbf{u}^T, \boldsymbol{\lambda}^T, \boldsymbol{\mu}^T)^T$
    $\hat{\mathbf{v}} \leftarrow$ cheb_semi_it$(\boldsymbol{Q}_{\bar{v}}, \mathbf{v})$
    $\hat{\mathbf{p}} \leftarrow \frac{1}{\alpha}$cheb_semi_it$(Q_p, \mathbf{u})$
    $\hat{\mathbf{u}} \leftarrow S_{33}^{-1}\mathbf{u}$, where $S_{33}$ is the Schur complement of $W_{33}$
    $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}) \leftarrow$ inex_uzawa$(W_{14}, W_{24}, \boldsymbol{\lambda}, \boldsymbol{\mu})$
    $\hat{\boldsymbol{\lambda}} \leftarrow \boldsymbol{Q}_{\bar{v}}\hat{\boldsymbol{\lambda}}$
    $\hat{\boldsymbol{\mu}} \leftarrow \alpha Q_p \hat{\boldsymbol{\mu}}$
    $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}) \leftarrow$ inex_uzawa$(W_{14}, W_{24}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}})$
```

Note that the (3,3) block inherits the saddle-point structure of the KKT matrix. Therefore, it is indefinite and cannot be used as a preconditioner for MINRES. Instead, we use the block-diagonal preconditioner (6) with the exact Schur complement, which is also cheap due to the small dimension.

Now, we consider the (1,1) block of the KKT system, namely, the velocity mass matrix $\boldsymbol{Q}_{\bar{v}}$. Due to the block-diagonal structure, it is sufficient to consider a scalar mass matrix Q_v. Recall that the permutation is constructed such that the contribution of finite element basis functions on the inflow boundary is removed. We would like to precondition this modified mass matrix as usual by a fixed number of Chebyshev semi-iterative steps. For this, we need explicit eigenvalue bounds of $D^{-1}Q$, where Q is the modified mass matrix and $D = \text{diag}(Q)$. This Q is just a Galerkin mass matrix with the action of certain basis functions removed; thus, the eigenvalue results from the work of Wathen[32] discussed earlier still apply here.

Especially, to get lower and upper bounds for the eigenvalues of $D^{-1}Q$, it is sufficient to compute the minimal and maximal eigenvalues of $D_e^{-1}Q_e$ for all element mass matrices Q_e and $D_e = \text{diag}(Q_e)$. In each of the elements, we remove the action of basis functions that are equal to one at one edge node and equal to zero at all the others. This is equivalent to removing the row and column associated with this node from a standard element matrix Q_e. We use a $\boldsymbol{Q}_2$ approximation for the velocity space, represented in Figure 5. A standard element mass matrix is a 9 × 9 matrix because we have nine nodes in each element. We remove some nodes located on one boundary, say, the left-hand one. Then, we need to consider index sets of the form $J \in 2^{\{1,4,7\}}$. We get modified element mass matrices Q_e^J by deleting the rows and columns of Q_e with the indices in J. By the symmetry of the nodes at 1 and 7, we can restrict ourselves to the cases shown in Table 6. Clearly, the eigenvalues bounds from Table 1 remain unchanged and we can apply the usual Chebyshev preconditioner.

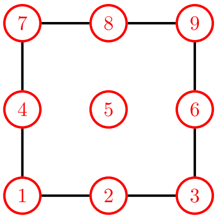


FIGURE 5 Representation of (left) $\boldsymbol{Q}_1$ and (right) $\boldsymbol{Q}_2$ elements

TABLE 6 Minimal and maximal eigenvalues of $(D_e^J)^{-1}Q_e^J$ for element mass matrices without nodes on the left-hand edge

J	$\varnothing$	{1}	{4}	{1,4}	{1,7}	{1,5,7}
$\lambda_{\min}$	0.2500	0.3125	0.3125	0.3506	0.3506	0.3750
$\lambda_{\max}$	1.5625	1.5625	1.5625	1.5625	1.5625	1.5625

6 | NUMERICAL RESULTS

We present some numerical results for the problems (1) and (7) using MINRES with the Rees–Wathen-type preconditioners discussed in the previous sections. All results were obtained with MATLAB and the IFISS package.[51,52] The convergence condition for both MINRES and the Picard outer iteration is a residual reduction by a factor of 10^{-6}.

6.1 | Stokes control

We apply the Rees–Wathen-type preconditioner from Section 3 with five Uzawa iterations as a Braess–Peisker approximation for the Stokes operator. The Laplacian is approximated by five algebraic multigrid (AMG) V-Cycles of the `HSL_MI20` solver,[53] which uses the Ruge–Stüben heuristics.[54] The mass matrices are approximated with 20 steps of Chebyshev semi-iteration based on a damped Jacobi method.

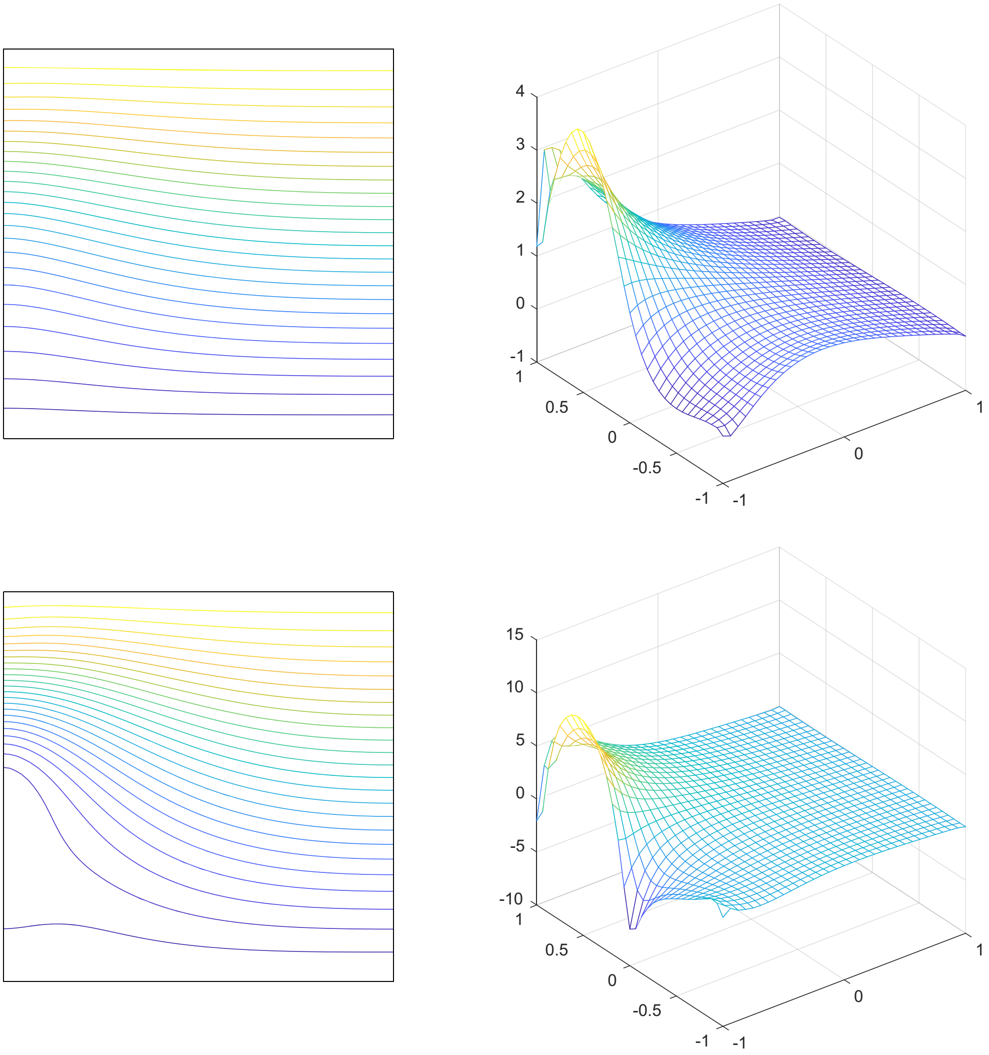
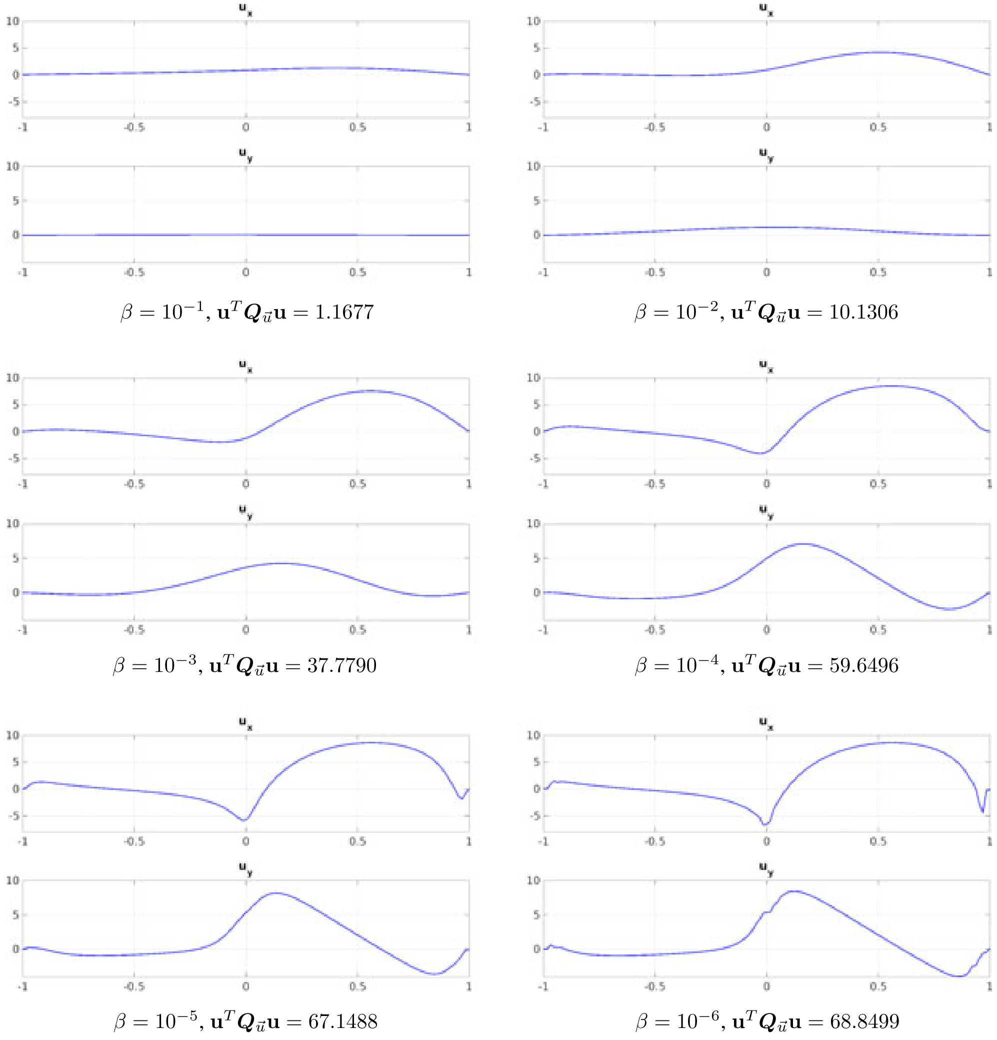


FIGURE 6 Solution of the test problem with $\alpha = 10^{-3}$ and (top) $\beta = 10^{-2}$ and (bottom) $\beta = 10^{-5}$

Typical solutions of the test problem are shown in Figure 6. It is clearly seen how the regularization parameter β influences the solution: For a small β, the optimal solution is closer to the desired velocity $\hat{\vec{v}}$. For a small β, the action of the control is penalized less, and so we can influence the solution more by applying the optimal control. However, as we can see in Figure 7, this effect appears to be bounded: $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u}$, which is the discrete approximation of the control penalty function $\|\vec{u}\|^2_{L^2(\partial\Omega)}$, grows as β decreases. However, the relatively small increase in the penalty function from $\beta = 10^{-5}$ to $\beta = 10^{-6}$ indicates that it may be bounded above by some constant. In the setting of the control problem, this would mean that there is a solution of the Stokes equations on this domain, which minimizes the cost function with $\beta = 0$, possibly with a noncontinuous control as the inflow boundary condition.

The presumed upper bound for $\|\vec{u}\|^2_{L^2(\partial\Omega)}$ corresponds to a lower bound for $\|\vec{v}\|^2_{L^2(\Omega)}$. As can be seen in Figure 6, even for relatively small β, we can only control the part of the channel close to the inflow; as it approaches the outflow, the



$\beta = 10^{-1}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 1.1677$

$\beta = 10^{-2}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 10.1306$

$\beta = 10^{-3}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 37.7790$

$\beta = 10^{-4}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 59.6496$

$\beta = 10^{-5}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 67.1488$

$\beta = 10^{-6}$, $\mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 68.8499$

FIGURE 7 Computed control with $\alpha = 10^{-3}$ and different β

TABLE 7 MINRES and MATLAB `backslash` performance on a 2^{-l} grid with $\alpha = \beta = 10^{-3}$

l	2	3	4	5	6	7	8	9
DoF	128	392	1,352	5,000	19,208	75,272	297,992	1,185,800
MINRES iterations	43	48	47	54	54	57	61	63
MINRES time	1.28	1.80	2.43	4.33	9.58	32.28	157.88	618.06
`backslash` time	0.01	0.01	0.04	0.2	1.46	11.4	—	—

Note. The dash “—” indicates failure of the direct method. MINRES = minimal residual.

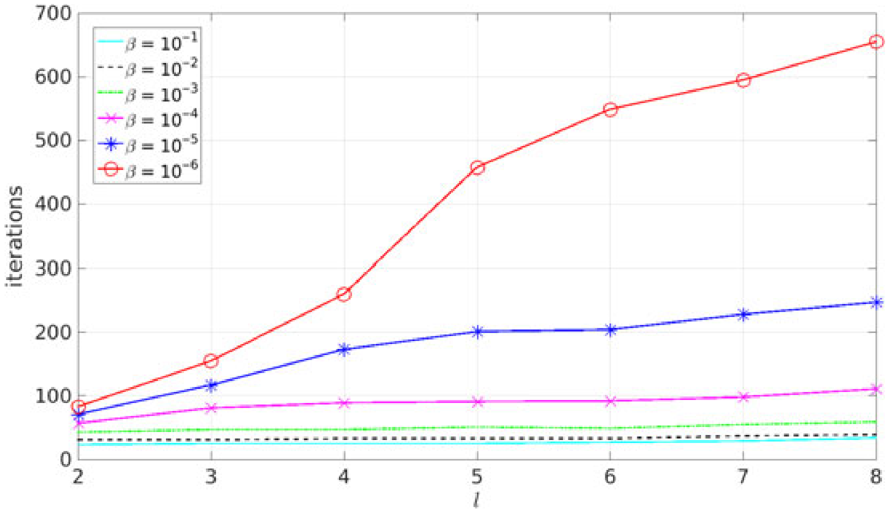


FIGURE 8 The minimal residual method's iteration counts with $\alpha = 10^{-3}$ and different β

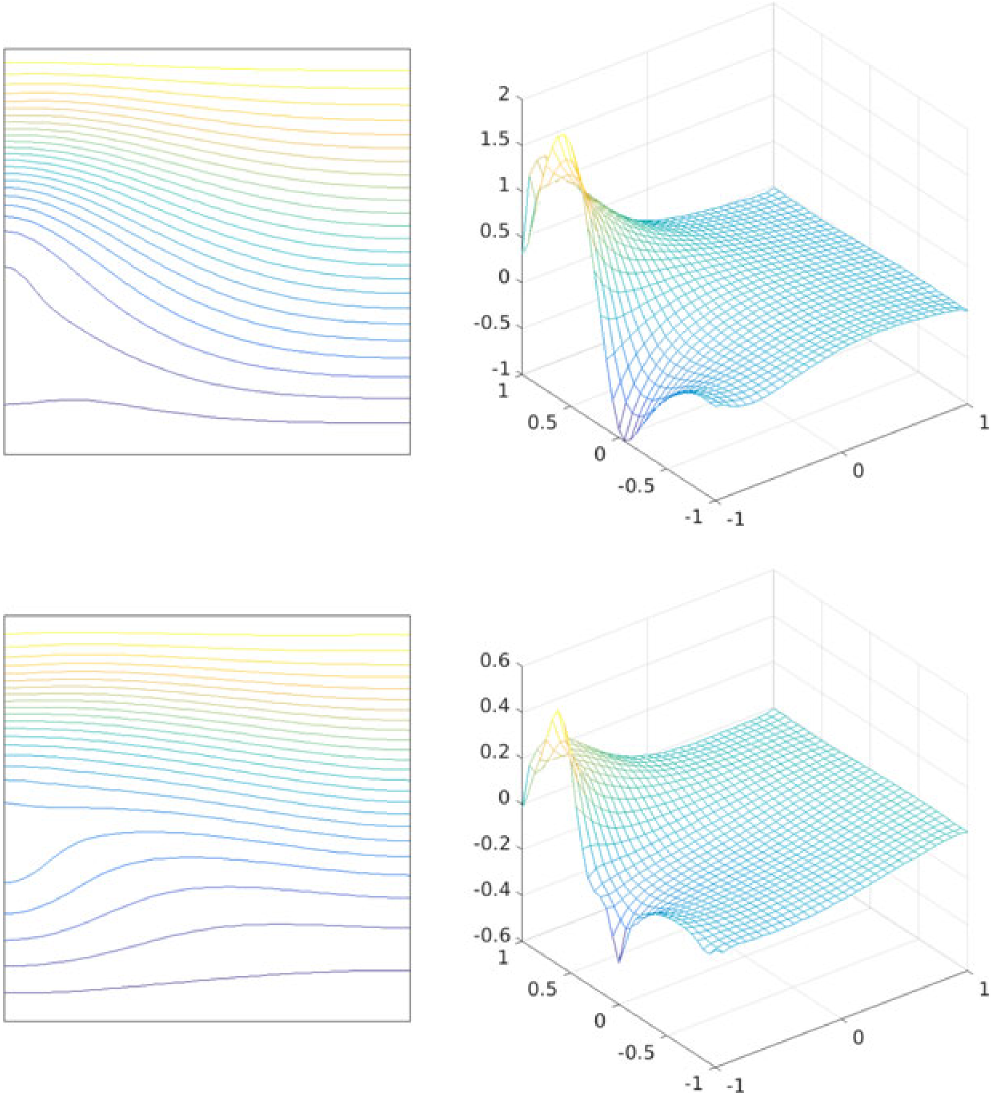


FIGURE 9 Solution of the test problem for (top) $\nu = 1/5$ and (bottom) $\nu = 1/30$

flow profile resembles more and more the Poiseuille flow we have seen in Section 2. This coincides with our intuition: If the control can only be applied on some part of the boundary, its impact gets smaller the farther we move away from this boundary in the domain.

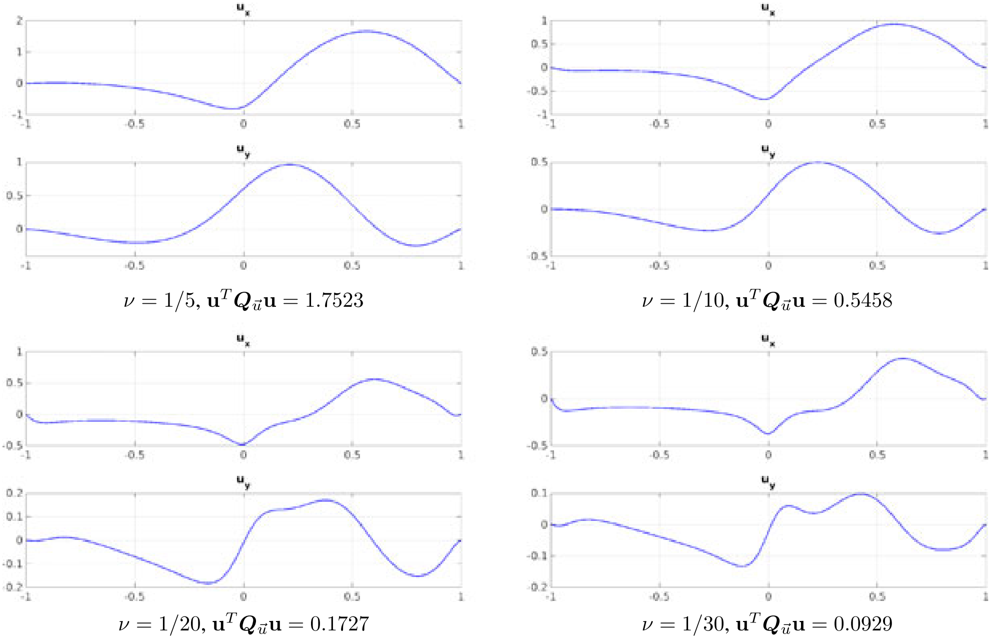
The preconditioning qualities are shown in Table 7. The preconditioned MINRES iteration needs $\mathcal{O}(1)$ iterations and has a complexity of $\mathcal{O}(N)$ per iteration because only sparse matrix–vector products are required. Thus, the time required for the computation is linear in N. For relatively small matrices, it is nevertheless outperformed by the MATLAB direct solver because of the constants involved. However, for very large matrices, it is clearly better than the direct solver that has much higher memory requirements, which is why it achieves no solution for very fine grids. The MINRES method only needs to store vectors. The growth rates of the computational time also suggest that MINRES should be faster for $h \leq 2^{-8}$ even with sufficient storage possibilities for $\Omega \subset \mathbb{R}^2$.

The dependence of preconditioning qualities on the control regularization parameter β is shown in Figure 8. We can observe the parameter independence of the preconditioner for relatively large values of β. For the smallest values, especially $\beta = 10^{-6}$, this behavior deteriorates, but still the results compare favorably with the case of distributed control presented by Rees et al.[17]

6.2 | Navier–Stokes control

We apply the permutational Rees–Wathen-type preconditioner from Section 5 with 30 Uzawa iterations as a Braess–Peisker approximation for the Oseen operator. Numerical experiments by Bramble et al. (section 6 of the work of Bramble et al.[26]) suggest using a δ in Algorithm 1 depending on the viscosity parameter ν; our choice is $\delta = \nu$. The parameter β is chosen to be equal to one. As a preconditioner for the (1,1) block in the discrete Oseen operator, we chose a modified incomplete Cholesky decomposition[55] implemented in MATLAB because the AMG solver used for the Stokes control problem in the previous chapter fails to construct meaningful algebraic course grid spaces here; it is based on a heuristic for elliptic problems, and for high Reynolds numbers, this is clearly not a good approximation here. We observe no sufficient smoothing. Note that incomplete decompositions in general do not give a grid-independent preconditioner; typical behavior of the condition number of the preconditioned system is $\kappa = \mathcal{O}(h^{-1})$. The Schur complement approximation in the Uzawa-type iteration is achieved, as usual, by a Chebyshev semi-iteration based on the pressure mass matrix.

A typical solution of the test problem is shown in Figure 9. As we would expect, for a viscous heavily diffusion-dominated flow with $\nu = 1/5$, the optimal solution is very similar to the Stokes case in Figure 6. For a less viscous flow with $\nu = 1/30$, the optimal solution is perceivably different and much closer to the desired flow profile. In addition, the pressure difference needed to maintain the flow is smaller, which is consistent with the general theory for the Navier–Stokes equations.



$\nu = 1/5,\ \mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 1.7523$

$\nu = 1/10,\ \mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 0.5458$

$\nu = 1/20,\ \mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 0.1727$

$\nu = 1/30,\ \mathbf{u}^T \boldsymbol{Q}_{\vec{u}} \mathbf{u} = 0.0929$

FIGURE 10 Computed control with different viscosity parameters ν

Figure 10 shows the computed control for different values of ν. The $\boldsymbol{Q}_{\bar{u}}$-norm of the control, and hence the energy required to obtain the optimal solution, decreases with the viscosity parameter ν. This means that, for relatively high Reynolds numbers, not only is the optimal solution closer to the desired state but it is also (in a physical sense) cheaper to achieve. In addition, we observe that the qualitative behavior of the control, that is, the number and location of maxima and minima, is similar to the Stokes case in Figure 7 for low Reynolds numbers. For higher Reynolds numbers, that is, $\mathcal{R} \geq 20$, we observe a qualitatively different behavior.

The preconditioning qualities are shown in Table 8. In contrast to the Stokes case, we do not observe grid-independent behavior. This is related to two issues. First, we lose some of the structure of our problem by permuting the rows and columns of the system matrix and dropping a low-rank perturbation. Second, in contrast to AMG, the incomplete Cholesky preconditioner applied here cannot guarantee grid-independent behavior. Nevertheless, we do achieve useful preconditioning qualities, especially in the case $\nu = 1/10$. We can conclude that preconditioners of this kind can be useful for Navier–Stokes problems with relatively low Reynolds numbers. For coarse grids, the removal of every inflow node is a good choice; for finer grids (which is probably more interesting in realistic applications), it appears advantageous to remove only some of the inflow nodes. However, if not enough inflow nodes are removed, we risk getting an indefinite system, especially on relatively coarse grids.

TABLE 8 Average iteration counts and number of nonlinear iterations (in parentheses) for the minimal residual method with the permutational preconditioner removing every inflow node and every second, fourth, sixth, and eighth inflow node (from top to bottom)

			l		
ν	**2**	**3**	**4**	**5**	**6**
1/5	48(8)	70(8)	108(8)	234(8)	705(8)
1/10	76(10)	117(10)	164(9)	273(9)	718(9)
1/20	—	196(14)	310(14)	495(14)	1,449(15)
1/30	—	273(18)	429(16)	775(18)	2,517(18)
			l		
ν	**2**	**3**	**4**	**5**	**6**
1/5	48(8)	70(8)	99(7)	202(8)	706(8)
1/10	76(10)	124(10)	159(10)	219(10)	626(10)
1/20	—	—	331(14)	411(14)	1,347(15)
1/30	—	—	563(16)	667(17)	2,285(18)
			l		
ν	**2**	**3**	**4**	**5**	**6**
1/5	—	62(8)	83(7)	138(8)	425(8)
1/10	—	123(10)	136(10)	188(10)	436(10)
1/20	—	—	—	457(15)	1,042(15)
1/30	—	—	819(16)	—	1,727(18)
			l		
ν	**2**	**3**	**4**	**5**	**6**
1/5	—	66(8)	73(7)	133(7)	346(8)
1/10	—	—	—	200(10)	406(10)
1/20	—	367(14)	1,294(14)	—	928(15)
1/30	—	—	—	—	3,341(19)
			l		
ν	**2**	**3**	**4**	**5**	**6**
1/5	—	—	72(8)	120(8)	318(8)
1/10	—	—	—	192(10)	381(10)
1/20	—	—	—	—	970(15)
1/30	—	—	—	—	—

Note. In the third table, one fourth of the inflow nodes are removed. The dash "—" indicates Uzawa divergence due to indefiniteness.

TABLE 9 Average iteration counts of the generalized minimal residual method with an incomplete Crout LU

	ν			
l	1/5	1/10	1/20	1/30
4	107	151	298	401
5	229	262	481	613

To assess the competitiveness of our novel permutation-based preconditioner, we compute some examples with GMRES using a generic incomplete LU preconditioner. As can be seen in Table 9, it outperforms our method slightly in terms of inner iteration counts. However, because our method does not require an incomplete LU of the whole matrix but only of smaller matrices, it is competitive and can be expected to outperform the generic method. Another advantage of our method is that we are able to use MINRES instead of GMRES, leading to further speedup.

7 | CONCLUSIONS AND POSSIBLE EXTENSIONS

In this article, we extended the application of a preconditioner for distributed Stokes control problems presented in the literature to the case of boundary control. It speeds up the convergence of MINRES considerably; as long as the regularization parameter β is not too small, the convergence is independent of the grid size. However, even for small values of β, we get useful preconditioning properties. We provide a theoretical explanation for this in the form that, for low β, preconditioning only deteriorates in terms of a low-rank perturbation. We believe that low-rank structures can be exploited in preconditioning for a range of optimal control problems; another example in recent literature is the work of Ye.[56] An objective for future research might be the development of parameter-independent preconditioners for Stokes boundary control, in the sense introduced in the work of Pearson et al.[20]

In a next step, we have discussed the applicability of preconditioners of this type to Navier–Stokes boundary control problems. We have seen that applicability is limited due to the intrinsic structures of Navier–Stokes Neumann boundary value problems, namely, the indefiniteness of the symmetric perturbation of the convection operator. We have presented a way to deal with these issues for problems governed by Navier–Stokes equations with low Reynolds numbers. Our heuristic strategy can be justified in terms of a low-rank perturbation. We have presented numerical results that support the theoretical reasoning. This preconditioner does not show grid-independent behavior because we do not have a grid-independent approximation for the nonstandard finite element operator involved. We believe that the performance of this preconditioner can be greatly improved by the application of an appropriate multigrid solver. The development of such a solver would require a closer analysis of this operator and could be the subject of future work.

A natural extension of the framework presented here is time-dependent boundary control; this can be combined with the work done by Stoll et al.[23] for time-dependent distributed control. In our test problems, the energy of the control is typically low; in fact, there is numerical evidence that it might be bounded above independently of the regularization parameter β. However, for other applications, it is conceivable that control constraints might be useful; this would require an active-set strategy, as implemented in the work of Stoll et al.[57] State constraints might also be useful in some settings; we refer to the recent works by Herzog et al.[58] and Pearson et al.[59] Another issue is the potential nonsmoothness and even noncontinuity of the control; if this is undesired from a practical point of view, it could possibly be dealt with by introducing regularization in a Sobolev seminorm (see the work of John[60]). Practical preconditioners for these problem formulations are yet to be developed. Another line of research may be a combination of low-rank perturbation ideas with the block matrix preconditioners introduced in the works of Axelsson et al.[6,25]

From a numerical point of view, an interesting extension would be a study the preconditioners presented here in the three-dimensional case. While all theoretical results carry over in a straightforward way, instead of a line, the inflow boundary becomes a two-dimensional interface, and the Schur complement approximation retains a low-rank error structure, as does the permutation based preconditioner. Because direct solvers are mostly infeasible in three dimensions, efficient preconditioning would be even more important here.

Another relevant question concerns the comparison of our novel method with standard preconditioners.

ACKNOWLEDGEMENTS

The first author has been supported by the German Academic Scholarship Foundation during his stay in Oxford; this work has been partially supported by the DFG Research Training Group 2126 "Algorithmic Optimization." Helpful remarks by two anonymous referees have helped considerably improve a first version of this paper.

ORCID

Gennadij Heidel https://orcid.org/0000-0002-4564-7356
Andy Wathen https://orcid.org/0000-0001-9992-5588

REFERENCES

1. Lions JL. Optimal control of systems governed by partial differential equations. Berlin, Germany: Springer Berlin Heidelberg; 1971.
2. Tröltzsch F. Optimal control of partial differential equations: Theory, methods and applications. Providence, RI: American Mathematical Society; 2010.
3. Borzì A, Schulz V. Computational optimization of systems governed by partial differential equations. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2012.
4. Wathen A, Silvester D. Fast iterative solution of stabilised Stokes systems. Part I: using simple diagonal preconditioners. SIAM J Numer Anal. 1993;30(3):630–649.
5. Silvester D, Wathen A. Fast iterative solution of stabilised Stokes systems. Part II: using general block preconditioners. SIAM J Numer Anal. 1994;31(5):1352–1367.
6. Axelsson O, Farouq S, Neytcheva M. Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems: Stokes control. Numer Algorithms. 2017;74(1):19–37.
7. Elman HC. Preconditioning for the steady-state Navier–Stokes equations with low viscosity. SIAM J Sci Comput. 1999;20(4):1299–1316.
8. Silvester D, Elman H, Kay D, Wathen A. Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow. J Computat Appl Math. 2001;128:261–279.
9. Kay D, Loghin D, Wathen A. A preconditioner for the steady-state Navier–Stokes equations. SIAM J Sci Comput. 2002;24(1):237–256.
10. Elman H, Howle VE, Shadid J, Shuttleworth R, Tuminaro R. Block preconditioners based on approximate commutators. SIAM J Sci Comput. 2006;27(5):1651–1668.
11. Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers: With applications in incompressible fluid dynamics. 2nd ed. Oxford, UK: Oxford University Press; 2014.
12. Axelsson O. Iterative solution methods. Cambridge, UK: Cambridge University Press; 1994.
13. Saad Y. Iterative methods for sparse linear systems. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2003.
14. Vassilevski PS. Multilevel block factorization preconditioners. New York, NY: Springer-Verlag New York; 2008.
15. Olshanskii MA, Tyrtyshnikov EE. Iterative methods for linear systems: Theory and applications. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2014.
16. Málek J, Strakoš Z. Preconditioning and the conjugate gradient method in the context of solving PDEs. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2014.
17. Rees T, Wathen AJ. Preconditioning iterative methods for the optimal control of the Stokes equations. SIAM J Sci Comput. 2011;33(5):2903–2926.
18. Schöberl J, Zulehner W. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. SIAM J Matrix Anal Appl. 2007;29(3):752–773.
19. Schöberl J, Simon R, Zulehner W. A robust multigrid method for elliptic optimal control problems. SIAM J Numer Anal. 2011;49(4):1482–1503.
20. Pearson JW, Wathen AJ. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer Linear Algebra Appl. 2012;19(5):816–829.
21. Pearson JW, Stoll M, Wathen AJ. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. SIAM J Matrix Anal Appl. 2012;33(4):1126–1152.
22. Pearson JW. On the development of parameter-robust preconditioners and commutator arguments for solving Stokes control problems. Electron Trans Numer Anal. 2015;44:53–72.
23. Stoll M, Wathen A. All-at-once solution of time-dependent Stokes control. J Comput Phys. 2013;232:498–515.
24. Pearson JW. Preconditioned iterative methods for Navier–Stokes control problems. J Comput Phys. 2015;292:194–207.
25. Axelsson O, Farouq S, Neytcheva M. Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems: Poisson and convection-diffusion control. Numer Algorithms. 2016;73(3):631–663.
26. Bramble JH, Pasciak JE, Vassilev AT. Uzawa type algorithms for nonsymmetric saddle point problems. Math Comput. 1999;69:667–689.
27. Heinkenschloss M. Formulation and analysis of a sequential quadratic programming method for the optimal Dirichlet boundary control of Navier–Stokes flow. In: Optimal control: Theory, algorithms, and applications. Boston, MA: Springer, 1998; p. 178–203.

28. Taylor C, Hood P. A numerical solution of the Navier–Stokes equations using the finite element technique. Comput Fluids. 1973;1(1):73–100.
29. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. Acta Numer. 2005;14:1–137.
30. Paige CC, Saunders MA. Solution of sparse indefinite systems of linear equations. SIAM J Numer Anal. 1975;12(4):617–629.
31. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. SIAM J Sci Comput. 2005;21(6):1969–1972.
32. Wathen AJ. Realistic eigenvalue bounds for the Galerkin mass matrix. IMA J Numer Anal. 1987;7(4):449–457.
33. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. Numer Math. 1961;3(1):147–156.
34. Wathen A, Rees T. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. Electron Trans Numer Anal. 2009;34:125–135.
35. Rees T, Stoll M. Block-triangular preconditioners for PDE-constrained optimization. Numer Linear Algebra Appl. 2010;17(6):977–996.
36. Rees T, Dollar HS, Wathen AJ. Optimal solvers for PDE-constrained optimization. SIAM J Sci Comput. 2010;32(1):271–298.
37. Wilkinson JH. The algebraic eigenvalue problem. Oxford, UK: Oxford University Press; 1965.
38. Golub GH, Van Loan CF. Matrix computations. Baltimore, MD: John Hopkins University Press; 2013.
39. Braess D, Peisker P. On the numerical solution of the biharmonic equation and the role of squaring matrices for preconditioning. IMA J Numer Anal. 1986;6(4):393–404.
40. Uzawa H. Iterative methods for concave programming. In: Studies in linear and nonlinear programming. Palo Alto, CA: Stanford University Press, 1958; p. 154–165.
41. Elman HC, Golub GH. Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J Numer Anal. 1994;31(6):1645–1661.
42. Bramble JH, Pasciak JE, Vassilev AT. Analysis of the inexact Uzawa algorithm for saddle point problems. SIAM J Numer Anal. 1997;34(3):1072–1092.
43. Zulehner W. Analysis of iterative methods for saddle point problems: a unified approach. Math Comput. 2002;71:479–505.
44. Hackbusch W. Multi-grid methods and applications. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 1985.
45. Trottenberg U, Oosterlee CW, Schüller A. Multigrid. San Diego, CA: Academic Press; 2001.
46. Pošta M, Roubíček T. Optimal control of Navier–Stokes equations by Oseen approximation. Comput Math Appl. 2007;53(3–4):569–581.
47. Pearson JW, Wathen AJ. Fast iterative solvers for convection-diffusion control problems. Electron Trans Numer Anal. 2013;40:294–310.
48. Rees T. Preconditioning iterative methods for PDE constrained optimization [PhD thesis]. Oxford, UK: University of Oxford; 2010.
49. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput. 1986;7(3):856–869.
50. Frank J, Vuik C. On the construction of deflation-based preconditioners. SIAM J Sci Comput. 2001;23(2):442–462.
51. Elman HC, Ramage A, Silvester DJ. Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. ACM Trans Math Softw. 2007;33(2):14.
52. Elman HC, Ramage A, Silvester DJ. IFISS: a computational laboratory for investigating incompressible flow problems. SIAM Rev. 2014;56(2):261–273.
53. Boyle J, Mihajlović M, Scott J. HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D. Int J Numer Meth Eng. 2010;82(1):64–98.
54. Ruge JW, Stüben K. Algebraic multigrid. In: McCormick SF, editor. Multigrid methods. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1987; p. 73–130.
55. Gustafsson I. A class of first order factorization methods. BIT Numer Math. 1978;18(2):142–156.
56. Ye X. Some preconditioners in optimization with partial differential equations [PhD thesis]. Trier, Germany: University of Trier; 2013.
57. Stoll M, Wathen A. Preconditioning for partial differential equation constrained optimization with control constraints. Numer Linear Algebra Appl. 2012;19(1):53–71.
58. Herzog R, Sachs E. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J Matrix Anal Appl. 2010;31(5):2291–2317.
59. Pearson JW, Stoll M, Wathen AJ. Preconditioners for state-constrained optimal control problems with Moreau–Yosida penalty function. Numer Linear Algebra Appl. 2014;21(1):81–97.
60. John LJ. Optimal boundary control in energy spaces: preconditioning and applications [PhD thesis]. Graz, Austria: Verlag der Technischen Universität; 2014.

How to cite this article: Heidel G, Wathen A. Preconditioning for boundary control problems in incompressible fluid dynamics. *Numer Linear Algebra Appl.* 2019;26:e2218. https://doi.org/10.1002/nla.2218