Numerical Algorithms (2021) 87:365–380
https://doi.org/10.1007/s11075-020-00970-1

ORIGINAL PAPER

PMHSS iteration method and preconditioners for Stokes control PDE-constrained optimization problems

Shan-Mou Cao[1] **· Zeng-Qi Wang**[2]

Received: 11 July 2019 / Revised: 20 April 2020 / Accepted: 16 June 2020 / Published online: 18 July 2020

Abstract

The preconditioned modified Hermitian/skew-Hermitian splitting (PMHSS) iteration method and the corresponding preconditioning technique can achieve satisfactory results for solving optimal control problems governed by Poisson's equation. We explore the feasibility of such a method and preconditioner for solving optimization problems constrained by the more complicated Stokes system. Theoretical results demonstrate that the PMHSS iteration method is convergent because the spectral radius of the iterative matrix is less than $\frac{\sqrt{2}}{2}$. Additionally, the PMHSS preconditioner still clusters eigenvalues on a unitary segment. It guarantees that the convergence of the PMHSS iteration method and preconditioning is independent of not only discretizing mesh size, but also of the Tikhonov regularization parameter. A more effective preconditioner is proposed based on the PMHSS preconditioner. The proposed preconditioner avoids the inner iterations when solving saddle point systems appearing in the generalized residual equations. Furthermore, it is still convergent and maintains its independence of parameter and mesh size.

Keywords PDE-constrained optimization problems · Stokes control · Preconditioning · Preconditioned modified Hermitian/skew-Hermitian splitting

✉ Zeng-Qi Wang
wangzengqi@sjtu.edu.cn

Shan-Mou Cao
csmyq@163.com

[1] School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, China

[2] School of Mathematical Sciences and Ministry of Education Key Lab in Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai, China

1 Introduction

Flow control has been widely used in the petroleum, chemical, and aeronautical engineering fields, and has become a very active research area in scientific computing. It is clear that developing efficient numerical methods for flow control is one of the keys to its successful application. In this paper, we consider numerical solutions for Stokes control problems to prepare for more complicated fluid dynamic problems, such as Navier-Stokes control problems. This study focuses on the solution of the multiple saddle point problems generated by the discretize-then-optimize approach. Our goal is to construct iterative solvers that are independent of not only the mesh size of the finite element discretization, but also of the regularization parameters for optimization. Some successful solvers and preconditioners for Stokes control optimization problems have been designed from different perspectives. Most preconditioners have been designed according to the properties of saddle point matrices. In [25], a parameter-robust block-diagonal preconditioner was derived from the nonstandard norm argument. In [14], block-diagonal and block-triangular preconditioners were generated based on a commutator argument [9]. Additionally, some iteration methods and preconditioners have been designed for the following reduced structured block 2×2 linear system:

$$\mathbf{Ax} \equiv \begin{bmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \equiv \mathbf{g}. \tag{1.1}$$

In [2], the authors studied the properties and numerical behaviors of a preconditioned square block (PRESB) preconditioner for solving (1.1). There have also been some studies focusing on solutions for the equivalent formulation of (1.1) as a complex symmetric indefinite linear system:

$$(\mathbf{W} + \mathrm{i}\mathbf{T})(\mathbf{y} + \mathrm{i}\mathbf{z}) = \mathbf{p} + \mathrm{i}\mathbf{q}.$$

Several complex arithmetic algorithms were proposed in [21, 22]. Several real arithmetic algorithms were proposed in [13, 16] under the assumption that $-\mathbf{T} \prec \mathbf{W} \prec \mathbf{T}$. Additional iterative methods derived from different matrix splitting techniques were studied in [17, 23, 24].

In this paper, we discuss the preconditioned modified Hermitian/skew-Hermitian splitting (PMHSS) iteration method and a corresponding preconditioner for Stokes control distributed problems. The PMHSS iteration method and preconditioner were proposed in [4] for Poisson control optimization. The PMHSS preconditioner has nearly the same workload as the PRESB preconditioner. However, it can be used in the short-term recurrence iteration methods (e.g., minimal residual method (MINRES) and Chebyshev semi-iteration method) [5, 18, 19]. The convergence of the PMHSS iteration method is analyzed when $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n \times n}$ are symmetric positive semidefinite matrices. When $\mathbf{W}$ or $\mathbf{T}$ are indefinite, the convergence rate of the PMHSS method and the preconditioned eigenvalue distribution deteriorate. In this study, we analyze the convergence of the PMHSS iteration method and the eigenvalue distribution of the PMHSS preconditioned matrix when $\mathbf{W}$ and $\mathbf{T}$ are a positive semidefinite matrix and a saddle point matrix, respectively. Similar to the PRESB preconditioner, in every iteration the PMHSS preconditioner requires solutions of two saddle point linear systems. Saddle point problems can be solved utilizing inner iterations, such as the Flexible Generalized Minimal Residual (GMRES) or parameterized and preconditioned Uzawa iterations [6, 7, 10]. Alternatively, to save computing cost and avoid decisions regarding inner tolerance, we propose a modified version of the PMHSS preconditioner called the rotated block constraint (RBC) preconditioner. The eigenvalue distribution of the corresponding preconditioned matrix is also analyzed.

The remainder of this paper is organized as follows. In Section 2, we describe the PMHSS iteration method and preconditioner for solving the Stokes control optimization problem. The convergence results and eigenvalue distribution are also derived. In Section 3, we derive the RBC preconditioner and analyze the eigenvalue distribution of the preconditioned matrix. In Section 4, we present existing feasible preconditioners for solving the Stokes control problems and describe their implementations in detail. In Section 5, the numerical performances of the introduced preconditioners are compared via testing on model problems. Our final conclusions are summarized in Section 6.

2 PMHSS iteration and preconditioner for optimality systems

We consider the following Stokes distributed control problem:

$$\min_{u,f} \frac{1}{2}\|y - y_d\|_2^2 + \frac{1}{2}\beta\|u\|_2^2,$$

$$\begin{aligned} \text{subject to} -\nabla^2 y + \nabla p &= u \quad \text{in} \quad \Omega, \\ \nabla \cdot y &= 0 \quad \text{in} \quad \Omega, \\ y &= g_D \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{2.1}$$

where Ω is a domain in $\mathbb{R}^2$ or $\mathbb{R}^3$ and $\partial\Omega$ is the boundary of Ω. The desired state function y_d and boundary value function g_D are given. Because the Stokes equation is self-adjoint, the discretize-then-optimize and optimize-then-discretize processes are mathematically equivalent and lead to the same solution. In this section, we discuss the algebraic systems that are obtained from the discretize-then-optimize approach. The rectangular Taylor-Hood finite element method is then adopted because it is inf-sup stable. Specifically, the velocity y and control u are approximated by linear combinations of the Q_2-basis functions $\{\phi_j\}$, $j = 1, \cdots, n_v$, while the pressure p is approximated by linear combinations of the Q_1-basis functions $\{\psi_j\}$, $k = 1, \cdots, n_p$. The first-order necessary optimality condition for the discretized optimization problem then yields the following linear system:

$$\mathcal{A}^R \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{l} \\ \mathbf{m} \end{bmatrix} \equiv \begin{bmatrix} M & 0 & -F^T & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{l} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{2.2}$$

where

$$F = \sqrt{\beta}\int_{\Omega} \nabla\phi_i : \nabla\phi_j, \quad M = \int_{\Omega} \phi_i\phi_j, \quad B = -\sqrt{\beta}\int_{\Omega} \psi_k \cdot \nabla\phi_j,$$

$$b = \int_{\Omega} y_d\phi_i - \sum_{j=n_y+1}^{n_y+n_\partial} \int_{\Omega} \nabla\phi_i : \nabla\phi_j,$$

$$f_i = -\sqrt{\beta}\sum_{j=n_y+1}^{n_y+n_\partial} y_j \int_{\Omega} \nabla\phi_i : \nabla\phi_j, \quad g_i = \sqrt{\beta}\sum_{j=n_y+1}^{n_y+n_\partial} y_j \int_{\Omega} \psi_i \nabla \cdot \phi_j,$$

$$f = [f_i], \quad g = [g_i],$$

and $\mathbf{l}$ and $\mathbf{m}$ are scaled Lagrange multipliers corresponding to $\mathbf{y}$ and $\mathbf{p}$, respectively. The matrices $M \in \mathbb{R}^{n_v \times n_v}$ and $F \in \mathbb{R}^{n_v \times n_v}$, which are referred to as the mass matrix and scaled stiffness matrix, respectively, are symmetric positive definite. The matrix $B \in \mathbb{R}^{n_p \times n_v}$ is of full row rank. We refer to [1, 11, 14] for the details of finite element discretization. The 4×4 block matrix $\mathcal{A}^R$ in (2.2) can be partitioned into the following 2×2 block form:

$$\mathcal{A}^R := \begin{bmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{bmatrix}, \tag{2.3}$$

where

$$\mathbf{W} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \tag{2.4}$$

are a symmetric positive semidefinite matrix and a non-singular saddle point matrix, respectively. In [4], Bai et al. defined the PMHSS iteration method when $\mathbf{W}$ and $\mathbf{T}$ were symmetric positive semidefinite matrices based on the following matrix splitting:

$$\mathcal{A}^R = \mathbf{F} - \mathbf{G},$$

where

$$\mathbf{F} := \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{W}+\mathbf{T} & 0 \\ 0 & \mathbf{W}+\mathbf{T} \end{bmatrix}, \quad \text{and} \quad \mathbf{G} := \begin{bmatrix} \mathbf{T} & -\mathbf{W} \\ \mathbf{W} & \mathbf{T} \end{bmatrix}. \tag{2.5}$$

The PMHSS iteration scheme for solving a 2×2 block system can be written as:

$$\mathbf{F}\mathbf{x}^{(k+1)} = \mathbf{G}\mathbf{x}^{(k)} + \mathbf{g}.$$

The matrix $\mathbf{F}$ is referred to as the PMHSS preconditioner. The PMHSS iteration method and preconditioner provide excellent results based on the clustered eigenvalue distributions and the fact that when $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive definite the eigenvectors form a normal matrix. In this study, we explored the performance of the PMHSS iteration method and preconditioner for the linear system in (2.2).

Lemma 2.1 *Let $\boldsymbol{W}$ and $\boldsymbol{T}$ be the matrices in* (2.4)*, where M and F are symmetric positive definite and B is of full row rank. Then, the eigenvalues of $\boldsymbol{T}^{-1}\boldsymbol{W}$ are all nonnegative.*

Proof Let τ and $x = [x_1; x_2]$ be the eigenvalues and corresponding eigenvectors of the following generalized eigenvalue problem:

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \tau \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which can be rewritten as:

$$\begin{cases} \tau(Fx_1 + B^T x_2) = Mx_1, \\ \tau Bx_1 \qquad\qquad\; = 0. \end{cases} \tag{2.6}$$

It is clear that $\tau = 0$ is the eigenvalue with a multiplicity n_p because $[0; x_2]$ are the corresponding eigenvectors for an arbitrary $x_2 \in \mathbb{R}^{n_p}$. In the case of $\tau \neq 0$, we have $Bx_1 = 0$. After multiplying by x_1^* on both sides of the first equality in (2.6), we have:

$$\tau(x_1^* F x_1 + x_1^* B^T x_2) = x_1^* M x_1.$$

Therefore, it holds that:

$$\tau = \frac{x_1^* M x_1}{x_1^* F x_1}. \tag{2.7}$$

Note that $x_1 \neq 0$ when $\tau \neq 0$. Otherwise, we have $x_2 = 0$ and x is not an eigenvector. Therefore, we have $\tau > 0$ because F and M are symmetric positive definite. It follows that:

$$x_2 = \frac{1}{\tau}(BB^T)^{-1}B(M - \tau F)x_1$$

for any $x_1 \in \mathrm{null}(B)/\{0\}$ and τ in (2.7). $\square$

Define $E = (\mathbf{W} + \mathbf{T})^{-1}(\mathbf{W} - \mathbf{T})$. It is trivial to verify that $\mu = \dfrac{\tau - 1}{\tau + 1}$ are the eigenvalues of E and $-1 \leq \mu < 1$. Furthermore, $\mu = -1$ are eigenvalues with a multiplicity at least n_p.

Theorem 2.1 *Let μ_j and $j = 1, \cdots, n_v + n_p$ be the eigenvalues of E. Then, the eigenvalues of $\boldsymbol{F}^{-1}\mathcal{A}^R$ are:*

$$\lambda_j^{\pm} = \frac{1}{2}(1 \pm \mathrm{i}\mu_j).$$

Proof The PMHSS preconditioned matrix is written as:

$$\begin{aligned}
\mathbf{F}^{-1}\mathcal{A}^R &= \frac{1}{2}\begin{bmatrix} \mathbf{W}+\mathbf{T} & 0 \\ 0 & \mathbf{W}+\mathbf{T} \end{bmatrix}^{-1}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \mathbf{W}+\mathbf{T} & 0 \\ 0 & \mathbf{W}+\mathbf{T} \end{bmatrix}^{-1}\begin{bmatrix} \mathbf{W}+\mathbf{T} & \mathbf{W}-\mathbf{T} \\ \mathbf{T}-\mathbf{W} & \mathbf{W}+\mathbf{T} \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \mathbf{I} & (\mathbf{W}+\mathbf{T})^{-1}(\mathbf{W}-\mathbf{T}) \\ (\mathbf{W}+\mathbf{T})^{-1}(\mathbf{T}-\mathbf{W}) & \mathbf{I} \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \mathbf{I} & E \\ -E & \mathbf{I} \end{bmatrix}.
\end{aligned}$$

Let λ_j and $w_j = (u_j^T,\ v_j^T)^T$ be the eigenvalues and eigenvectors of $\mathbf{F}^{-1}\mathcal{A}^R$, respectively. It holds that:

$$\frac{1}{2}\begin{bmatrix} \mathbf{I} & E \\ -E & \mathbf{I} \end{bmatrix}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \lambda_j \begin{bmatrix} u_j \\ v_j \end{bmatrix}.$$

Consequently, we have:

$$E^2 v_j = -(2\lambda - 1)^2 v_j.$$

This indicates that $-(2\lambda_j - 1)^2$ and v_j are the eigenvalues and corresponding eigenvectors of E^2. Then, $\lambda_j^{\pm} = \frac{1}{2}(1 \pm \mathrm{i}\mu_j)$. □

According to Theorem 2.1, the eigenvalues of preconditioned coefficient matrix $\mathbf{F}^{-1}\mathcal{A}^R$ are located on the unitary segment between $\frac{1}{2}(1+\mathrm{i})$ and $\frac{1}{2}(1-\mathrm{i})$. The eigenvalues of the PMHSS iterative matrix:

$$\mathbf{L} = \mathbf{F}^{-1}\mathbf{G} = I - \mathbf{F}^{-1}\mathcal{A}^R$$

are $1 - \lambda_j^{\pm} = \frac{1}{2}(1 \mp \mathrm{i}\mu_j)$. The PMHSS iteration method is convergent because the spectral radius of the iterative matrix is not greater than $\frac{1}{\sqrt{2}}$.

When PMHSS preconditioning is used in Krylov subspace methods, the generalized residual equation $\mathbf{Fr} = \mathbf{z}$ must be solved in every iteration. The main workload lies in solving two linear saddle point problems with:

$$\mathbf{W} + \mathbf{T} = \begin{bmatrix} F + M & B^T \\ B & 0 \end{bmatrix} \tag{2.8}$$

in every iteration. These saddle point equations can be solved utilizing inner iteration methods, as discussed in [1].

3 RBC preconditioner for optimality systems

In this section, we propose a new preconditioner to avoid solving saddle point equations. We introduce the matrices:

$$\widetilde{\mathbf{W}} = \begin{bmatrix} M & 0 \\ 0 & S \end{bmatrix} \quad \text{and} \quad \widetilde{\mathbf{T}} = \begin{bmatrix} F & B^T \\ B & -\hat{S} \end{bmatrix}, \tag{3.1}$$

where $S = B(M+F)^{-1}B^T$ is the (negative) Schur complement of (2.8) and $\hat{S}$ serves as a symmetric positive definite approximation of S. Therefore, the matrices $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{T}}$ are symmetric positive definite and symmetric indefinite, respectively. The RBC preconditioner is defined as:

$$\widetilde{\mathbf{F}} := \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \widetilde{\mathbf{W}} + \widetilde{\mathbf{T}} & 0 \\ 0 & \widetilde{\mathbf{W}} + \widetilde{\mathbf{T}} \end{bmatrix}. \tag{3.2}$$

The preconditioner $\widetilde{\mathbf{F}}$ does not need to be generated explicitly because:

$$\widetilde{\mathbf{W}} + \widetilde{\mathbf{T}} = \begin{bmatrix} M + F & B^T \\ B & S - \hat{S} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ B(M+F)^{-1} & \mathbf{I} \end{bmatrix}\begin{bmatrix} M + F & B^T \\ 0 & -\hat{S} \end{bmatrix},$$

is the product of a block unit lower triangular matrix and block upper triangular matrix. $\widetilde{\mathbf{W}}+\widetilde{\mathbf{T}}$ is considered as a constraint preconditioner for the saddle point matrix $\mathbf{W}+\mathbf{T}$. The computing cost for solving the generalized residual equations:

$$\widetilde{\mathbf{F}}\mathbf{z} = \mathbf{r},$$

includes two solutions with $M+F$ and one solution with $\hat{S}$. A reasonable $\hat{S}$ will yield an effective and efficient preconditioner $\widetilde{\mathbf{F}}$. We discuss how the preconditioner $\widetilde{\mathbf{F}}$ approximates $\mathcal{A}^R$ in the following theorem.

Theorem 3.1 *Let $M \in \mathbb{R}^{n_v \times n_v}$ and $F \in \mathbb{R}^{n_v \times n_v}$ be symmetric positive definite matrices and let $B \in \mathbb{R}^{n_p \times n_v}$ be a full column rank. $\widetilde{\boldsymbol{F}} \in \mathbb{R}^{2n \times 2n}$, which is defined in (3.2), is the RBC preconditioner for $\mathcal{A}^R \in \mathbb{R}^{2n \times 2n}$ in (2.2) with $n = n_v + n_p$. $\hat{S}$ serves as a symmetric positive definite approximation of $S = B(M+F)^{-1}B^T$. We define $\Delta S = S - \hat{S}$. The Jordan decomposition of $\boldsymbol{F}^{-1}\mathcal{A}^R$ is:*

$$\boldsymbol{F}^{-1}\mathcal{A}^R = \boldsymbol{Q}\boldsymbol{J}\boldsymbol{Q}^{-1}, \quad \boldsymbol{J} = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{bmatrix}, \tag{3.3}$$

where

$$J_i = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_i & 1 \\ & & & & \lambda_i \end{bmatrix} \quad or \quad J_i = \begin{bmatrix} \lambda_i & & & & \\ & \lambda_i & & & \\ & & \ddots & & \\ & & & \lambda_i & \\ & & & & \lambda_i \end{bmatrix}, \quad i = 1, \cdots, k$$

are the Jordan blocks in $\boldsymbol{J}$ and λ_i are the eigenvalues of $\boldsymbol{F}^{-1}\mathcal{A}^R$. If m is the order of the largest Jordan block in $\boldsymbol{J}$ and

$$\|\hat{S}^{-1}\Delta S\| \le \frac{\sqrt{2}}{1+\sqrt{2}} \frac{1}{m+1} \frac{1}{\kappa(\boldsymbol{Q})} \frac{1}{\sqrt{1+\|(M+F)^{-1}B^T\|^2}},$$

then for any eigenvalue σ of $\widetilde{\boldsymbol{F}}^{-1}\mathcal{A}^R$, there is an eigenvalue λ of $\boldsymbol{F}^{-1}\mathcal{A}^R$ such that

$$|\sigma - \lambda| \le \frac{m}{m+1}(m+1)^{\frac{1}{m}} \left(1+\frac{\sqrt{2}}{2}\right)^{\frac{1}{m}} \kappa(\boldsymbol{Q})^{\frac{1}{m}} \left(\sqrt{1+\|(M+F)^{-1}B^T\|^2}\right)^{\frac{1}{m}} \|\hat{S}^{-1}\Delta S\|^{\frac{1}{m}},$$

where $\kappa(Q) = \|\boldsymbol{Q}^{-1}\|\|\boldsymbol{Q}\|$. Furthermore, if m_0 is the order of the largest Jordan block to which λ belongs, then:

$$|\sigma - \lambda| \le \frac{m_0}{m_0+1}(m_0+1)^{\frac{1}{m_0}} \left(1+\frac{\sqrt{2}}{2}\right)^{\frac{1}{m_0}} \kappa(\boldsymbol{Q})^{\frac{1}{m_0}} \left(\sqrt{1+\|(M+F)^{-1}B^T\|^2}\right)^{\frac{1}{m_0}} \|\hat{S}^{-1}\Delta S\|^{\frac{1}{m_0}}.$$

Proof We denote:

$$\mathbf{D} = \begin{bmatrix} \mathbf{W}+\mathbf{T} & 0 \\ 0 & \mathbf{W}+\mathbf{T} \end{bmatrix}, \quad \widetilde{\mathbf{D}} = \begin{bmatrix} \widetilde{\mathbf{W}}+\widetilde{\mathbf{T}} & 0 \\ 0 & \widetilde{\mathbf{W}}+\widetilde{\mathbf{T}} \end{bmatrix}, \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix}.$$

Then, the preconditioned matrix can be expressed as:

$$\widetilde{\mathbf{F}}^{-1}\mathcal{A}^R = \widetilde{\mathbf{D}}^{-1}\mathbf{P}^{-1}\mathcal{A}^R.$$

Note that

$$\widetilde{\mathbf{D}}^{-1} = \mathbf{D}^{-1} - \widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{D}} - \mathbf{D})\mathbf{D}^{-1}$$

and we denote

$$\delta_{\mathbf{D}} = \widetilde{\mathbf{D}} - \mathbf{D} = \begin{bmatrix} 0 & 0 & & \\ 0 & S - \hat{S} & & \\ & & 0 & 0 \\ & & 0 & S - \hat{S} \end{bmatrix}.$$

Therefore, we have:

$$\widetilde{\mathbf{F}}^{-1}\mathcal{A}^R = \mathbf{F}^{-1}\mathcal{A}^R - \widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{F}^{-1}\mathcal{A}^R.$$

Let $\Delta S = S - \hat{S}$. Then,

$$\left\|\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\right\| = \|(\widetilde{\mathbf{W}} + \widetilde{\mathbf{T}})^{-1}(\widetilde{\mathbf{W}} + \widetilde{\mathbf{T}} - (\mathbf{W} + \mathbf{T}))\| = \left\|\begin{bmatrix} 0 & (M+F)^{-1}B^T\hat{S}^{-1}\Delta S \\ 0 & -\hat{S}^{-1}\Delta S \end{bmatrix}\right\|.$$

It is clear that $\left\|\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\right\|$ is the square root of the maximum eigenvalue of

$$\begin{bmatrix} 0 & 0 \\ 0 & ((M+F)^{-1}B^T\hat{S}^{-1}\Delta S)^T((M+F)^{-1}B^T\hat{S}^{-1}\Delta S) + (\hat{S}^{-1}\Delta S)^T(\hat{S}^{-1}\Delta S) \end{bmatrix}.$$

Note that

$$\begin{aligned}
&\|((M+F)^{-1}B^T\hat{S}^{-1}\Delta S)^T((M+F)^{-1}B^T\hat{S}^{-1}\Delta S) + (\hat{S}^{-1}\Delta S)^T(\hat{S}^{-1}\Delta S)\| \\
\le\ &\|((M+F)^{-1}B^T\hat{S}^{-1}\Delta S)^T((M+F)^{-1}B^T\hat{S}^{-1}\Delta S)\| + \|(\hat{S}^{-1}\Delta S)^T(\hat{S}^{-1}\Delta S)\| \\
\le\ &\|(M+F)^{-1}B^T\hat{S}^{-1}\Delta S)\|^2 + \|\hat{S}^{-1}\Delta S\|^2 \\
\le\ &(1 + \|(M+F)^{-1}B^T\|^2)\|\hat{S}^{-1}\Delta S\|^2,
\end{aligned}$$

we have

$$\left\|\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\right\| \le \sqrt{1 + \left\|(M+F)^{-1}B^T\right\|^2}\left\|\hat{S}^{-1}\Delta S\right\|.$$

Consider the eigenvalue distribution of $\mathbf{F}^{-1}\mathcal{A}^R$, we have $\|\mathbf{J}\| \le 1 + \frac{\sqrt{2}}{2}$. We derive the estimates of σ as follows. When

$$\|\hat{S}^{-1}\Delta S\| \le \frac{1}{\sqrt{1 + \|(M+F)^{-1}B^T\|^2}}\frac{1}{\|\mathbf{Q}^{-1}\|\|\mathbf{Q}\|}\frac{\sqrt{2}}{1+\sqrt{2}}\frac{1}{m+1},$$

it is clear that

$$\begin{aligned}
\|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{F}^{-1}\mathcal{A}^R\mathbf{Q}\| &= \|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{F}^{-1}\mathcal{A}^R\mathbf{Q}\| \\
&= \|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{F}^{-1}\mathcal{A}^R\mathbf{Q}\| \\
&= \|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{Q}\mathbf{J}\| \\
&\le \|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{Q}\|\|\mathbf{J}\| \\
&\le \frac{1}{m+1}.
\end{aligned}$$

According to Theorem 2 in [12], for any eigenvalue σ of $\widetilde{\mathbf{F}}^{-1}\mathcal{A}^R$, there is an eigenvalue λ of $\mathbf{F}^{-1}\mathcal{A}^R$ such that:

$$\begin{aligned}
|\sigma-\lambda| &\leq \frac{m}{m+1}(m+1)^{\frac{1}{m}}\|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{Q}\mathbf{J}\|^{\frac{1}{m}} \\
&\leq \frac{m}{m+1}(m+1)^{\frac{1}{m}}\|\mathbf{Q}^{-1}\|^{\frac{1}{m}}\|\mathbf{Q}\|^{\frac{1}{m}}\left\|\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\right\|^{\frac{1}{m}}\|\mathbf{J}\|^{\frac{1}{m}} \\
&\leq \frac{m}{m+1}(m+1)^{\frac{1}{m}}\|\mathbf{Q}^{-1}\|^{\frac{1}{m}}\|\mathbf{Q}\|^{\frac{1}{m}}\left(\sqrt{1+\|(M+F)^{-1}B^T\|^2}\right)^{\frac{1}{m}}\|\hat{S}^{-1}\Delta S\|^{\frac{1}{m}}\|\mathbf{J}\|^{\frac{1}{m}} \\
&\leq \frac{m}{m+1}(m+1)^{\frac{1}{m}}\left(1+\frac{\sqrt{2}}{2}\right)^{\frac{1}{m}}\|\mathbf{Q}^{-1}\|^{\frac{1}{m}}\|\mathbf{Q}\|^{\frac{1}{m}}\left(\sqrt{1+\|(M+F)^{-1}B^T\|^2}\right)^{\frac{1}{m}}\|\hat{S}^{-1}\Delta S\|^{\frac{1}{m}}.
\end{aligned}$$

Furthermore, if m_0 is the order of the largest Jordan block to which λ belongs, then:

$$\begin{aligned}
|\sigma-\lambda| &\leq \frac{m_0}{m_0+1}(m_0+1)^{\frac{1}{m_0}}\|\mathbf{Q}^{-1}\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\mathbf{Q}\mathbf{J}\|^{\frac{1}{m_0}} \\
&\leq \frac{m_0}{m_0+1}(m_0+1)^{\frac{1}{m_0}}\|\mathbf{Q}^{-1}\|^{\frac{1}{m_0}}\|\mathbf{Q}\|^{\frac{1}{m_0}}\left\|\widetilde{\mathbf{D}}^{-1}\delta_{\mathbf{D}}\right\|^{\frac{1}{m_0}}\|\mathbf{J}\|^{\frac{1}{m_0}} \\
&\leq \frac{m_0}{m_0+1}(m_0+1)^{\frac{1}{m_0}}\left(1+\frac{\sqrt{2}}{2}\right)^{\frac{1}{m_0}}\|\mathbf{Q}^{-1}\|^{\frac{1}{m_0}}\|\mathbf{Q}\|^{\frac{1}{m_0}}\left(\sqrt{1+\|(M+F)^{-1}B^T\|^2}\right)^{\frac{1}{m_0}}\|\hat{S}^{-1}\Delta S\|^{\frac{1}{m_0}}.
\end{aligned}$$

□

It is clear that σ tends towards λ when ΔS tends towards zero. Furthermore, it is always true that $|\sigma-\lambda| \leq \frac{m}{m+1}$ when the premise in Theorem 3.1 holds. A good approximation of the Schur complement leads to a value of σ_i close to λ_i. For the Stokes problem in a convex domain, $S_p = \beta\left(\sqrt{\beta}M_p^{-1}+F_p^{-1}\right)^{-1}$ is a well-known approximation of S that was proposed in [9], where M_p and F_p are the mass matrix and stiffness matrix of the pressure space, respectively. This approximation was also adopted in [1, 14, 25].

4 Feasible preconditioners and computational implementations

We now discuss some effective preconditioners for the Stokes control problem and their corresponding main workloads. In [25, 26], a preconditioner of the form:

$$\mathcal{P}_{nsn} = \begin{bmatrix} M+F & 0 & 0 & 0 \\ 0 & \frac{1}{\beta}(M+F) & 0 & 0 \\ 0 & 0 & \left(F_p^{-1}+\sqrt{\beta}M_p^{-1}\right)^{-1} & 0 \\ 0 & 0 & 0 & \beta\left(F_p^{-1}+\sqrt{\beta}M_p^{-1}\right)^{-1} \end{bmatrix} \tag{4.1}$$

was proposed for the linear system in (2.2), where M_p and F_p are the pressure mass matrix and pressure stiffness matrix, respectively.

In [14], a preconditioner of the form:

$$\mathcal{P}_{cta} = \begin{bmatrix} M & 0 & 0 & 0 \\ 0 & \frac{1}{\beta}(F+M)M^{-1}(F+M)^T & 0 & 0 \\ 0 & 0 & F_p & 0 \\ 0 & 0 & 0 & (M_p^{-1}F_pM_p^{-1} + \frac{1}{\beta}F_p^{-1})^{-1} \end{bmatrix} \tag{4.2}$$

was designed to approximate the reordered coefficient matrix:

$$\mathcal{A} = \begin{bmatrix} M & F & B^T & 0 \\ F & -\frac{1}{\beta}M & 0 & B^T \\ B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \end{bmatrix} \tag{4.3}$$

Utilizing an operator commutator argument. The preconditioners $\mathcal{P}_{nsn}$ and $\mathcal{P}_{cta}$ can be utilized in the MINRES method because they are symmetric positive definite. Numerical experiments verified that the convergence of preconditioned MINRES methods is independent of the mesh size h of discretization. However, it does rely on the regularization parameter β. The preconditioner:

$$\mathcal{P}_{presb} = \begin{bmatrix} M & 0 & -F & -B^T \\ 0 & 0 & -B & 0 \\ F & B^T & M+2F & 2B^T \\ B & 0 & 2B & 0 \end{bmatrix} \tag{4.4}$$

was proposed in [1] and later referred to as the PRESB preconditioner in [3]. It was proven that the condition number for the preconditioned system is no greater than 2 if the preconditioning is performed precisely. The performance of this preconditioner is not only parameter independent, but also mesh size independent. In the implementation of PRESB, there are two subsystems with a saddle point matrix (2.8) that must be solved in every iteration. Similarly, PMHSS must also solve two subsystems with a matrix (2.8) in every iteration. However, these two subsystems can be solved independently. For practical consideration, the linear saddle point systems should be solved approximately utilizing inner iterations, such as Uzawa-type methods [6, 10] or the flexible GMRES method with a block triangular preconditioner.

$$\mathcal{P}_{tri} = \begin{bmatrix} M+F & \\ B & -\hat{S} \end{bmatrix}, \tag{4.5}$$

where $\hat{S}$ serves as an approximation of the Schur complement of matrix (2.8). In our numerical experiments, we set $\hat{S} = S_p$. Table 1 lists the linear systems involved in every iteration for different preconditions. The common workload of the preconditioning processes is the solving of linear systems with F_p, M_p, M, and $M+F$. As implemented in previous works [1, 14], the linear systems with F_p were solved utilizing algebraic multigrid (AMG) methods. According to the eigenvalue distribution in [15, 20], linear systems with M_p can be solved in 20 steps using Chebyshev semi-iteration with relaxed Jacobi iteration with a damping parameter of $\omega = 4/5$.

Table 1 Computational complexity of different preconditioners

Preconditioner	Systems to be solved
$\mathcal{P}_{nsn}$	$2 \times (M+F), 2 \times F_p, 2 \times M_p$
$\mathcal{P}_{cta}$	$2 \times (M+F), 2 \times F_p, 2 \times M_p, 1 \times M$
$\mathcal{P}_{presb}$	$2 \times \begin{bmatrix} M+F & B^T; B & 0 \end{bmatrix}$
	Inner iterations: $\{1 \times (M+F), 1 \times F_p, 1 \times M_p\}$
$\mathcal{P}_{pmhss}$	$2 \times \begin{bmatrix} M+F & B^T; B & 0 \end{bmatrix}$
	Inner iterations: $\{1 \times (M+F), 1 \times F_p, 1 \times M_p\}$
$\mathcal{P}_{rbc}$	$2 \times (M+F), 2 \times F_p, 2 \times M_p$

Similarly, linear systems with M can be solved in 20 steps utilizing Chebyshev semi-iteration with relaxed Jacobi iteration with a damping parameter of $\omega = 32/29$. Linear systems with a coefficient matrix $M + F$ are solved utilizing AMG methods.

5 Numerical results

In this section, we compare the numerical performances of the preconditioners discussed above by solving the Stokes control model (2.1) in $\Omega = [0,\ 1]^2$ with different desired states y_d and different boundary value conditions.

Example 5.1 $y = y_d$ on the boundary and $y_d(x_1, x_2) = (y_{d,1}(x_1, x_2), y_{d,2}(x_1, x_2))$, given by $y_{d,1}(x_1, x_2) = 10\frac{\partial}{\partial x_2}(\varphi(x_1)\varphi(x_2))$ and $y_{d,2}(x_1, x_2) = -10\frac{\partial}{\partial x_1}(\varphi(x_1)\varphi(x_2))$ with $\varphi(z) = (1 - \cos(0.8\pi z))(1 - z)^2$.

Example 5.2 The desired state $y_d = x_2\mathbf{i} - x_1\mathbf{j}$ and let $y = 0$ on the boundary, except for on $x_1 = 1, 0 \le x_2 \le 1$, where $y = -\mathbf{j}$.

In our experiments, all systems with mass matrices M and M_p were solved utilizing the Chebyshev semi-iteration method. The Chebyshev semi-iteration method is performed for a maximum of 20 steps until the relative residual is reduced to 10^{-4}. We employed an AMG routine called HSL_MI20 from the Harwell Subroutine Library [8]. This routine performs two V-cycles with two pre- and post-smooth (symmetric Gauss-Seidel) steps to solve a linear system with F_p and $M + F$. Saddle point subsystems with $\mathbf{W} + \mathbf{T}$ are solved utilizing a flexible preconditioned GMRES method with a preconditioner $\mathcal{P}_{tri}$. The inner iterations are terminated once the relative residual is reduced to 10^{-4}. The symmetric preconditioners $\mathcal{P}_{nsn}$ and $\mathcal{P}_{cta}$ from (4.1) and (4.2), respectively, are utilized to precondition the MINRES method for solving a linear system with the matrix in (4.3). The non-symmetric preconditioners $\mathcal{P}_{presb}$, $\mathbf{F}$, and $\tilde{\mathbf{F}}$ from (4.4), (2.5) and, (3.2), respectively, are solved by a preconditioned GMRES method. In our implementations, all iteration processes start from initial vectors containing zeros and terminate as soon as the relative residuals of (2.2) are less than 10^{-6} or the maximum number of iterations $IT_{\max} = 500$ is reached. The number of iterations (denoted as IT) and CPU time in seconds (denoted as CPU) are reported with respect to different mesh size h and regularization parameter β. The numbers of degree of freedom (denoted as DOF) are reported also. For the PMHSS preconditioned GMRES and PRESB preconditioned GMRES methods, the average number of inner iterations (denoted as IT_inner) for solving the saddle point subsystems are listed in brackets. All tests are performed in Matlab 8.0.0 (64 bit).

In Table 2, we list the IT and CPU values of different preconditioned Krylov subspace methods for solving Example 5.1. As mentioned in previous studies, NSN preconditioned MINRES and CTA preconditioned MINRES are mesh-independent methods, but their convergence depends on a regularization parameter β. Both NSN and CTA preconditioners perform better with smaller values of β.

When $\beta = 10^{-2}$ and $h = 2^{-7}$, the CTA preconditioned MINRES method abnormally breaks down based on the inexact implementation of preconditioning. The data in Table 2 reveals that similar to the PRESB preconditioned GMRES method, the PMHSS and RBC preconditioned GMRES methods are not only mesh size independent, but also parameter independent. Furthermore, the RBC preconditioner is advantageous because it saves more than 50% of CPU time compared with the

Table 2 Problem 1: IT, IT_inner (in brackets), and CPU of different Krylov subspace methods

DOF	Method	$\beta = 10^{-2}$		$\beta = 10^{-4}$		$\beta = 10^{-6}$		$\beta = 10^{-8}$		$\beta = 10^{-10}$	
	PMHSS	10(10)	2.68e−1	13(9)	2.81e−1	13(6)	2.03e−1	10(5)	1.82e−1	6(5)	1.03e−1
1318	RBC	28	*1.17e−1*	29	*1.30e−1*	26	*1.11e−1*	20	1.28e−1	13	8.18e−2
$h = 2^{-4}$	PRESB	6(11)	1.84e−1	8(8)	1.81e−1	7(6)	1.21e−1	5(6)	*1.13e−1*	3(6)	*6.55e−2*
	NSN	71	2.48e−1	55	1.82e−1	38	1.37e−1	26	1.23e−1	18	8.39e−2
	CTA	129	4.99e−1	77	2.98e−1	50	2.03e−1	24	1.19e−1	15	7.33e−2
	PMHSS	10(11)	6.38e−1	13(10)	6.75e−1	13(7)	5.13e−1	11(6)	4.16e−1	6(5)	1.90e−1
4936	RBC	31	*2.63e−1*	29	*2.75e−1*	24	*2.48e−1*	19	2.41e−1	10	1.32e−1
$h = 2^{-5}$	PRESB	6(12)	4.01e−1	8(9)	4.18e−1	7(7)	3.03e−1	5(6)	*2.16e−1*	3(6)	*1.20e−1*
	NSN	81	5.33e−1	66	4.31e−1	42	3.11e−1	26	2.20e−1	18	1.48e−1
	CTA	160	1.32	102	8.24e−1	62	5.50e−1	28	2.76e−1	13	1.24e−1
	PMHSS	10(13)	2.10	13(10)	2.29	13(7)	1.53	11(6)	1.29	7(5)	6.55e−1
19078	RBC	33	*1.05*	29	*8.12e−1*	23	*6.38e−1*	16	*6.22e−1*	10	*3.90e−1*
$h = 2^{-6}$	PRESB	6(13)	1.53	8(11)	1.61	7(7)	9.30e−1	5(6)	8.07e−1	3(6)	4.26e−1
	NSN	89	1.92	74	1.60	46	1.01	26	7.08e−1	17	4.21e−1
	CTA	206	5.18	130	3.25	78	1.98	33	9.86e−1	15	4.15e−1
	PMHSS	10(14)	8.93	13(11)	9.77	12(8)	5.85	11(6)	6.06	8(5)	2.98
75014	RBC	34	*3.76*	29	*3.09*	23	*2.42*	16	*3.08*	10	*1.33*
$h = 2^{-7}$	PRESB	6(14)	5.40	8(12)	6.34	7(8)	3.53	5(6)	3.30	4(6)	1.87
	NSN	101	8.50	81	6.36	53	4.19	26	*3.08*	18	1.59
	CTA	—	—	158	1.54e+1	92	9.25	40	5.84	18	2.19

Table 3 Problem 2: IT, IT_inner (in brackets), and CPU of different Krylov subspace methods

DOF	Method	$\beta = 10^{-2}$		$\beta = 10^{-4}$		$\beta = 10^{-6}$		$\beta = 10^{-8}$		$\beta = 10^{-10}$	
	PMHSS	10(11)	2.88e−1	12(9)	2.99e−1	11(7)	2.05e−1	9(6)	1.97e−1	6(6)	1.28e−1
1318	RBC	26	*9.61e−2*	30	*1.15e−1*	28	1.18e−1	27	1.69e−1	21	1.32e−1
$h = 2^{-4}$	PRESB	5(11)	1.46e−1	7(9)	1.66e−1	6(7)	1.17e−1	5(7)	*1.18e−1*	5(7)	1.34e−1
	NSN	79	2.43e−1	49	1.50e−1	35	*1.15e−1*	27	1.19e−1	23	*1.02e−1*
	CTA	117	4.44e−1	72	2.72e−1	46	1.85e−1	28	1.42e−1	22	1.11e−1
	PMHSS	10(12)	6.71e−1	11(10)	6.36e−1	11(8)	5.21e−1	9(6)	3.96e−1	6(6)	2.43e−1
4936	RBC	26	*2.30e−1*	27	*2.49e−1*	25	*2.55e−1*	24	3.12e−1	21	2.63e−1
$h = 2^{-5}$	PRESB	5(12)	3.41e−1	6(10)	3.18e−1	6(8)	2.95e−1	5(7)	*2.29e−1*	5(7)	*2.35e−1*
	NSN	81	5.11e−1	55	3.47e−1	41	2.91e−1	29	2.45e−1	23	1.94e−1
	CTA	139	1.14	89	7.31e−1	54	4.97e−1	32	3.28e−1	21	2.16e−1
	PMHSS	8(12)	1.66	10(11)	1.87	10(9)	1.50	9(7)	1.39	7(6)	1.00
19078	RBC	26	*7.60e−1*	25	*7.12e−1*	20	*5.62e−1*	19	7.26e−1	18	6.32e−1
$h = 2^{-6}$	PRESB	4(13)	9.15e−1	6(11)	1.10	5(9)	7.34e−1	5(7)	*7.08e−1*	4(6)	5.09e−1
	NSN	83	1.66	57	1.13	45	8.79e−1	29	7.15e−1	19	*4.51e−1*
	CTA	164	4.46	106	2.97	62	1.72	36	1.23	18	5.81e−1
	PMHSS	8(12)	6.63	10(12)	8.03	9(9)	5.65	8(7)	5.82	7(6)	3.40
75014	RBC	25	*2.68*	23	*2.41*	17	*1.90*	14	*2.58*	12	1.58
$h = 2^{-7}$	PRESB	4(14)	3.70	5(12)	3.82	5(10)	3.09	4(7)	2.93	4(6)	1.84
	NSN	83	6.59	59	4.62	45	3.57	33	4.05	15	*1.43*
	CTA	191	2.18e+1	123	1.35e+1	71	7.98	41	6.28	16	1.88

PMHSS preconditioner in cases with small mesh sizes. The RBC preconditioner outperforms the other preconditioners, including the PRESB preconditioner, in terms of CPU time in most cases. Regarding the number of iterations, the PRESB preconditioned GMRES method outperforms the other iteration methods. However, it

Table 4 The cost functional of the distributed control problem 5.1

Method	β	IT	$\|u\|_2$	$\|y-\hat{y}\|_2$	$\|y-\hat{y}\|_2/\|\hat{y}\|_2$	$\mathcal{J}$	$\|b-\mathcal{A}x\|_2/\|b\|_2$	time
	10^{-2}	33	6.96e+1	3.69e+1	9.66e−1	7.06e+2	8.54e−7	1.05
	10^{-4}	29	1.58e+3	8.91	2.33e−1	1.65e+2	7.66e−7	8.12e−1
RBC	10^{-6}	23	2.13e+3	2.62e−1	6.85e−3	2.31	8.23e−7	6.38e−1
	10^{-8}	16	2.18e+3	1.22e−2	3.18e−4	2.39e−2	8.80e−7	6.22e−1
	10^{-10}	10	2.19e+3	4.99e−4	1.31e−5	2.39e−4	6.76e−7	3.90e−1
	10^{-2}	6	6.96e+1	3.69e+1	9.66e−1	7.06e+2	2.84e−7	1.53
	10^{-4}	8	1.58e+3	8.91	2.33e−1	1.65e+2	9.70e−7	1.61
PRESB	10^{-6}	7	2.13e+3	2.62e−1	6.85e−3	2.31	6.77e−7	9.30e−1
	10^{-8}	5	2.18e+3	1.22e−2	3.18e−4	2.39e−2	5.02e−7	8.07e−1
	10^{-10}	3	2.19e+3	4.64e−4	1.21e−5	2.39e−4	3.11e−7	4.26e−1

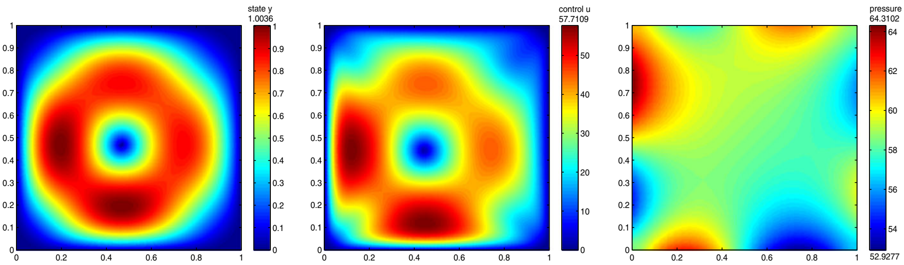

Fig. 1 Problem 1: State y, control u, and pressure p for $h = 2^{-6}$ and $\beta = 10^{-6}$

has to solve several saddle point problems utilizing inner loops. The same is true for the PMHSS preconditioned GMRES method. The data in Table 3 support these conclusions.

We report the cost functional $\mathcal{J}$ of distributed control problem in Example 5.1 for $h = 2^{-6}$ in Table 4. The results generated by the two methods have no inapparent difference. With the decreasing of β, the cost functional decreases and the state y goes close to the desired state $\hat{y}$. However, the control u becomes large. It indicates that $\beta = 10^{-10}$ is a proper regularization parameter for Example 5.1.

We plot the velocity, control, and pressure values for Examples 5.1 and 5.2 for the cases of $h = 2^{-6}$ and $\beta = 10^{-6}$ based on data calculated utilizing the RBC preconditioned GMRES method in Figs. 1 and 2. The maximum value of velocity y in the 2-norm is 1.00 and the maximum value of control u in the 2-norm is 57.7. These results are consistent with the results in [1, 25]. Additionally, the images of pressure and velocity in Fig. 2 coincide with the results in [1].

6 Conclusion

In this paper, we study PMHSS-type iteration methods and corresponding preconditioners for solving the Stokes control problems. By approximating the saddle point matrix in the PMHSS preconditioner as a product of lower and upper matrices, a more practical preconditioner was designed. This new preconditioner avoids solving saddle point systems in every iteration. The eigenvalue distribution of the preconditioned

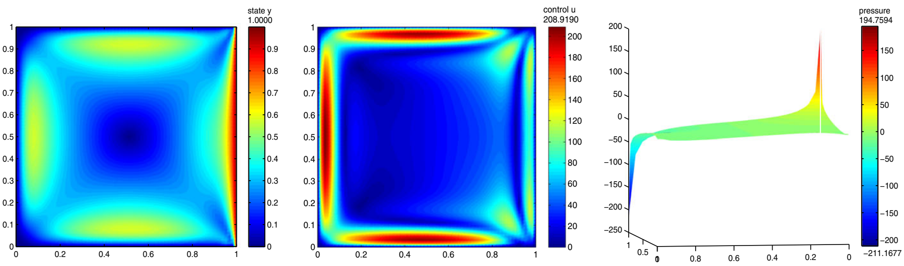

Fig. 2 Problem 2: State y, control u, and pressure p for $h = 2^{-6}$ and $\beta = 10^{-6}$

matrix is favorable and numerical experiments demonstrated that preconditioning behaviour is independent of the mesh size of discretization, as well as of the regularization parameter. The proposed preconditioning can also be extended to solving more complicated Navier-Stokes control problems.

Funding information This research is supported by the National Natural Science Foundation of China (11371022).

References

1. Axelsson, O., Farouq, S., Neytcheva, M.: Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems Stokes control. Numer. Algorithms **74**, 19–37 (2017)
2. Axelsson, O., Neytcheva, M., Ahmad, B.: A comparison of iterative methods to solve complex valued linear algebraic systems. Numer. Algorithms **66**(4), 811–841 (2014)
3. Axelsson, O., Davod, S.K.: A new version of a preconditioning method for certain two-by-two block matrices with square blocks. BIT Numer. Math. **59**(2), 321–342 (2019)
4. Bai, Z.-Z., Benzi, M., Chen, F., Wang, Z.-Q.: Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA J. Numer. Anal. **33**(1), 343–369 (2013)
5. Bai, Z.-Z.: On preconditioned iteration methods for complex linear systems. J. Eng. Math. **93**(1), 41–60 (2015)
6. Bai, Z.-Z., Parlett, B.N., Wang, Z.-Q.: On generalized successive overrelaxation methods for augmented linear systems. Numerische Mathematik **102**(1), 1–38 (2005)
7. Bai, Z.-Z., Wang, Z.-Q.: On parameterized inexact Uzawa methods for generalized saddle point problems. Linear Algebra and its Applications **428**, 2900–2932 (2008)
8. Boyle, J., Mihajlovic, M.D., Scott, J.A.: HSL_MI20 An Efficient AMG preconditioner for finite element problems in 3D. Int. J. Numer. Methods Eng. **82**(1), 64–98 (2009)
9. Cahouet, J., Chabard, J.P.: Some finite element solvers for the generalized Stokes problem. Int. J. Numer. Methods Fluids **8**(8), 869–895 (1988)
10. Elman, H.C., Golub, G.H.: Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J. Numer. Anal. **31**(6), 1645–1661 (1994)
11. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics. Oxford University Press, USA (2014)
12. Jiang, E.: Bounds for the smallest singular value of a Jordan block with an application to eigenvalue perturbation. Linear Algebra and its Applications **197**, 691–707 (1994)
13. Liao, L., Zhang, G.-F.: The generalized C-to-R method for solving complex symmetric indefinite linear systems. Linear and Multilinear Algebra **67**, 1–9 (2019)
14. Pearson, J.W.: On the development of parameter-robust preconditioners and commutator arguments for solving Stokes control problems. Electornic Trans. Numer. Anal. **44**, 53–72 (2015)
15. Rees, T., Sue Dollar, H., Wathen, A.J.: Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. **32**(1), 271–298 (2010)
16. Ren, Z.-R., Cao, Y., Zhang, L.-L.: On preconditioned MHSS real-valued iteration methods for a class of complex symmetric indefinite linear systems. East Asian J. Appl. Math. **6**(2), 192–210 (2016)
17. Shen, Q.-Q., Shi, Q.: A variant of the HSS preconditioner for complex symmetric indefinite linear systems. Computers & Mathematics with Applications **75**(3), 850–863 (2018)
18. Wang, Z.eng.-Q.i.: Restrictively preconditioned Chebyshev method for solving systems of linear equations. J. Eng. Math. **93**(1), 61–76 (2015)
19. Wang, Z.-Q.: On a Chebyshev accelerated splitting iteration method with application to two-by-two block linear systems. Numerical Linear Algebra with Applications **25**(5), e2712 (2018)
20. Wathen, A.J.: Realistic eigenvalue bounds for the Galerkin mass matrix. IMA J. Numer. Anal. **7**(4), 449–457 (1987)
21. Wu, S.-L., Li, C.-X.: Modified complex-symmetric and skew-Hermitian splitting iteration method for a class of complex-symmetric indefinite linear systems. Numer. Algorithms **76**(1), 93–107 (2017)

22. Xu, W.: A generalization of preconditioned MHSS iteration method for complex symmetric indefinite linear systems. Appl. Math. Comput. **219**(21), 10510–10517 (2013)
23. Zhang, J.-H., Dai, H.: A new block preconditioner for complex symmetric indefinite linear systems. Numer. Algorithms **74**(3), 889–903 (2017)
24. Zhang, J.-L., Fan, H.-T., Gu, C.-Q.: An improved block splitting preconditioner for complex symmetric indefinite linear systems. Numer. Algorithms **77**(2), 451–478 (2018)
25. Zulehner, W.: Nonstandard norms and robust estimates for saddle point problems. SIAM Journal on Matrix Analysis and Applications **32**(2), 536–560 (2011)
26. Zulehner, W.: Efficient solvers for saddle point problems with applications to PDE-constrained optimization, pp. 197–216. Springer, Berlin (2013)