Journal of Computational Mathematics
Vol.32, No.3, 2014, 215–232.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1401-CR3

A NEW PRECONDITIONING STRATEGY FOR SOLVING A CLASS OF TIME-DEPENDENT PDE-CONSTRAINED OPTIMIZATION PROBLEMS*

Minli Zeng
School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China
College of Mathematics, Putian University, Putian 351100, China
Email: zengml12@lzu.edu.cn
Guofeng Zhang
School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China
Email: gf_zhang@lzu.edu.cn

Abstract

In this paper, by exploiting the special block and sparse structure of the coefficient matrix, we present a new preconditioning strategy for solving large sparse linear systems arising in the time-dependent distributed control problem involving the heat equation with two different functions. First a natural order-reduction is performed, and then the reduced-order linear system of equations is solved by the preconditioned MINRES algorithm with a new preconditioning techniques. The spectral properties of the preconditioned matrix are analyzed. Numerical results demonstrate that the preconditioning strategy for solving the large sparse systems discretized from the time-dependent problems is more effective for a wide range of mesh sizes and the value of the regularization parameter.

Mathematics subject classification: 65F08, 65F10, 65F50, 49M25, 49K20, 65N22.
Key words: PDE-constrained optimization, Reduced linear system of equations, Preconditioning, Saddle point problem, Krylov subspace methods.

1. Introduction

In this paper, we focus on preconditioned iterative methods for solving the large linear system arising in the time-dependent distributed control problem involving the heat equation. Specifically, we consider the following distributed control of the heat equations:

$$\begin{aligned} &\min_{y,u} J(y,u), \\ &\text{subject to} \begin{cases} \frac{\partial y}{\partial t} - \nabla^2 y = u, & \text{for } (x,t) \in \Omega \times (0,T), \\ y = f, & \text{on } \partial\Omega \times (0,T), \\ y = y_0, & \text{at } t = 0, \end{cases} \end{aligned} \tag{1.1}$$

for certain functional $J(y,u)$, where f and y_0 depend maybe on x but not on t. Two target functionals to be considered in this paper are

$$J_1(y,u) = \frac{1}{2}\int_0^T\int_\Omega (y(x,t)-\bar{y}(x,t))^2 d\Omega dt + \frac{\beta}{2}\int_0^T\int_{\partial\Omega} (u(x,t))^2 d\Omega dt,$$

* Received August 25, 2013 / Revised version received January 2, 2014 / Accepted January 15, 2014 / Published online May 22, 2014 /

and

$$J_2(y,u) = \frac{1}{2}\int_\Omega (y(x,T)-\bar{y}(x))^2 d\Omega + \frac{\beta}{2}\int_0^T\int_{\partial\Omega}(u(x,t))^2 d\Omega dt.$$

Here y, u, $\bar{y}$ and p are vectors corresponding to the state, control, desired state and adjoint at all time steps $1, 2, \cdots, N_t$, respectively, and β is the regularization parameter.

By the way of $J(y,u) = J_1(y,u)$ being applied the discretize-then-optimize approach ([20,34]), which discretizes this problem with equal-order finite element basis functions for y, u, and the adjoint variable p, results in the linear system ([34])

$$\begin{pmatrix} \tau\mathcal{M}_{1/2} & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M}_{1,1} \\ \mathcal{K} & -\tau\mathcal{M}_{1,1} & 0 \end{pmatrix}\begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,1}\bar{y} \\ 0 \\ d \end{pmatrix}, \tag{1.2}$$

where $\mathcal{K}$, $\mathcal{M}_{1/2}$ and $\mathcal{M}_{1,1}$ are all matrices in $\mathbb{R}^{(nN_t)\times(nN_t)}$, and

$$\mathcal{K} = \begin{pmatrix} M+\tau K & & & & \\ -M & M+\tau K & & & \\ & \ddots & \ddots & & \\ & & -M & M+\tau K & \\ & & & -M & M+\tau K \end{pmatrix},$$

$$\mathcal{M}_{1/2} = \begin{pmatrix} \frac{1}{2}M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \frac{1}{2}M \end{pmatrix}, \quad d = \begin{pmatrix} \mathcal{M}_{1,1}y_0 + c \\ c \\ \vdots \\ c \\ c \end{pmatrix}.$$

Here and in the following, I denotes the identity matrix. Denote by

$$\mathcal{M}_2 := \begin{pmatrix} 2M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & 2M \end{pmatrix}, \quad \mathcal{I}_s := \begin{pmatrix} sI & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & sI \end{pmatrix},$$

$$\mathcal{M}_{\gamma,\delta} := \begin{pmatrix} \gamma M & & & & \\ & \gamma M & & & \\ & & \ddots & & \\ & & & \gamma M & \\ & & & & \delta M \end{pmatrix} \in \mathbb{R}^{(nN_t)\times(nN_t)},$$

where $s = \frac{1}{2}$ or 2. In the above, N_t is the number of time steps of (constant) size τ used to discretize the PDEs, c the boundary conditions of the PDEs and M a finite element mass matrix and K a stiffness matrix on Ω, which are of the dimension $n\times n$ with n being the degrees of freedom of the finite element approximation.

If $J(y,u) = J_1(y,u)$ is alternatively used in the optimize-then-discretize approach, the linear system (1.2) becomes

$$\begin{pmatrix} \tau\mathcal{M}_{1,0} & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M}_{1,1} \\ \mathcal{K} & -\tau\mathcal{M}_{1,1} & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,0}\bar{y} \\ 0 \\ d \end{pmatrix}. \tag{1.3}$$

If $J(y,u) = J_2(y,u)$, using the discretize-then-optimal approach, the linear system becomes

$$\begin{pmatrix} \tau\mathcal{M}_{0,1} & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M}_{1,1} \\ \mathcal{K} & -\tau\mathcal{M}_{1,1} & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{0,1}\bar{y} \\ 0 \\ d \end{pmatrix}. \tag{1.4}$$

As many other problems involving optimization with constraints ([18, 21, 23, 26]), after discretization, these problems often lead to the linear system of equations (1.2), (1.3) or (1.4) of saddle point structure, see, e.g., [14–16, 25, 30, 32–34]. We see that the total size of the linear system (1.2), (1.3) or (1.4) is $(3nN_t) \times (3nN_t)$. A key fact about the discretization of these systems is that the matrix blocks may be extremely large even for rather coarse mesh discretizations because of the higher dimensional setting. Therefore iterative methods are usually employed for their solution and finding an efficient numerical method becomes very important. Moreover, this system must be properly preconditioned in order to avoid stagnation in the convergence in terms of the norm of the residual. Many efficient iterative and preconditioned methods have been studied in many literature. For example, Uzawa-like methods ([10, 16]), GSOR methods ([9]), RPCG methods ([8]), HSS-like methods ([4, 6, 7]), and so on. We refer to [3, 14] for algebraic properties for saddle point problems. However, the matrix splitting methods can not be separated from the preconditioning techniques, which can be found in many literature, e.g., [2, 3, 9, 11, 16, 17, 20, 28, 30, 34]. Thus research has recently gone into developing preconditioners ([1, 2, 5, 22, 24, 25, 27–30, 32–34]) that are insensitive to regularization parameter as well as the mesh size.

Recently, Pearson, Stoll and Wathen ([27]) built a solver for the boundary control problem, both in the time-independent Poisson control and the time-dependent heat equation control cases. It is well known that an initial reduction ([12, 32]) of the matrix size may lead to significant savings, as long as this reduction does not entail extra computational burden.

In this paper, following the strategies of [12, 21, 24, 25, 27] and [32], first a particularly simple and effective reduction is performed. Thus the total size of the linear system (1.2), (1.3) or (1.4) can be reduced to $(2nN_t) \times (2nN_t)$. Then the solution of the reduced-order linear system exploits to build effective preconditioning techniques for the obtained reduced-order system. Finally, a new preconditioning needs only to be concerned with the relevant blocks in the reduced-order system is presented. Numerical experiments show that the CPU time to solve the structured linear system of equations arising in the time-dependent PDE-constrained optimization problems is significantly reduced for a wide range of mesh sizes and the value of the regularization parameter. Therefore the new preconditioning strategy in this paper is much effective.

The organization of the paper is as follows. In Section 2, a new strategy for solving the structured linear system of equations arising in time-dependent PDE-constrained optimization problems is described. First, the reduced-order linear system is established and then the preconditioning technique for the reduced-order system is presented. In Section 3, the theoretical analysis of the preconditioning approaches is given. In Section 4, numerical results for a variety of test problems to demonstrate the effectiveness of the new strategy for solving the structured linear system of equations arising in time-dependent PDE-constrained optimization problems are provided. In Section 5, some concluding remarks are given.

2. The New Solution Strategy

In this section, firstly, the reduced-order linear systems for the structured linear systems (1.2), (1.3) and (1.4) will be obtained. Then some new structured preconditioning techniques for the reduced-order linear system will be explored.

Noticing that $\mathcal{M}_{1/2}$ in (1.2) is a symmetric positive definite(SPD) matrix, and the middle block leads to $(\beta\tau\mathcal{M}_{1/2})u = (\tau\mathcal{M}_{1,1})p$. Hence we have

$$u = \frac{1}{\beta}\begin{pmatrix} 2I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & 2I \end{pmatrix} p = \frac{1}{\beta}\mathcal{I}_2 p.$$

Then the number of blocks of the linear system (1.2) can be decreased from 3×3 to 2×2, resulting in the reduced-order linear system

$$\begin{pmatrix} \tau\mathcal{M}_{1/2} & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}\tau\mathcal{M}_2 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,1}\bar{y} \\ d \end{pmatrix},$$

or equivalently,

$$\mathcal{A}_1\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A_1 & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}C \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,1}\bar{y} \\ d \end{pmatrix}, \tag{2.1}$$

where $A_1 := \tau\mathcal{M}_{1/2}$, $C := \tau\mathcal{M}_2$.

In a similar way, the reduced-order system for (1.3) can be written as

$$\begin{pmatrix} \tau\mathcal{M}_{1,0} & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}\tau\mathcal{M}_2 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,0}\bar{y} \\ d \end{pmatrix}.$$

Because $\mathcal{M}_{1,0}$ is a rank-deficient matrix (see [34]), we use $\mathcal{M}_{1,\gamma}$ instead of the matrix $\mathcal{M}_{1,0}$ and choose $0 < \gamma \ll 1$ so that the matrix $\mathcal{M}_{1,\gamma}$ is SPD and sufficiently approximate to $\mathcal{M}_{1,0}$. Then the reduced-order system for (1.3) can be rewritten as

$$\mathcal{A}_2\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A_2 & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}C \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{1,\gamma}\bar{y} \\ d \end{pmatrix}, \tag{2.2}$$

with $A_2 = \tau\mathcal{M}_{1,\gamma}$.

By the same strategies as above, we use $\mathcal{M}_{\gamma,1}$ instead of the matrix $\mathcal{M}_{0,1}$. The reduced-order system for the linear system (1.4) can be obtained as

$$\begin{pmatrix} \tau\mathcal{M}_{\gamma,1} & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}\tau\mathcal{M}_2 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{\gamma,1}\bar{y} \\ d \end{pmatrix}.$$

Also the reduced-order linear system for (1.4) can be rewritten as

$$\mathcal{A}_3\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A_3 & \mathcal{K}^T \\ \mathcal{K} & -\frac{1}{\beta}C \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau\mathcal{M}_{\gamma,1}\bar{y} \\ d \end{pmatrix}, \tag{2.3}$$

with $A_3 = \tau\mathcal{M}_{\gamma,1}$.

Next, we consider preconditioning techniques for the reduced-order linear systems (2.1), (2.2) and (2.3). Noticing that all the three reduced-order linear systems (2.2), (2.3) and (2.4) have the same form:

$$\mathcal{A}z := \begin{pmatrix} A & B^T \\ B & -\frac{1}{\beta}C \end{pmatrix} z = g, \tag{2.4}$$

where A and $C \in \mathbb{R}^{(nN_t)\times(nN_t)}$ are SPD matrices and $B \in \mathbb{R}^{(nN_t)\times(nN_t)}$ is a block lower-triangular matrix with symmetric positive definite diagonal blocks.

For the general form of the reduced-order linear system (2.4), there exist many efficient preconditioners.

When A and B are symmetric positive semi-definite and with at least one of them being positive definite, Bai ([5]) introduced an additive block diagonal preconditioner. Following this idea, we can consider the following additive block diagonal preconditioner for the reduced-order linear system (2.4), denoted by $\mathcal{P}_{ABD}$:

$$\mathcal{P}_{ABD} = \begin{pmatrix} A + \sqrt{\beta}B & \\ & \frac{1}{\beta}(C + \sqrt{\beta}B) \end{pmatrix}.$$

The block diagonal preconditioner with the Schur complement can be described as

$$\mathcal{P}_S = \begin{pmatrix} A & \\ & \frac{1}{\beta}C + BA^{-1}B^T \end{pmatrix}.$$

In practice, solving linear systems involving the Schur complement can be expensive. In an actual implementation, an application of preconditioner will involve some approximate of the Schur complement.

Based on the approach for approximation of the Schur complement developed in [28] (see also in [1, 13]), we give some new preconditioner for the linear systems (2.1)-(2.4).

We define $\mathcal{E}_1 := \frac{1}{\sqrt{\beta}}A + B_d$, $\mathcal{E}_2 := \frac{1}{\sqrt{\beta}}C + B_d$, where $B_d = \text{Blkdiag(B)}$ is the block diagonal matrix with the block the same in the diagonal block in B. We have $(B_d + \frac{1}{\sqrt{\beta}}A)A^{-1}(B_d + \frac{1}{\sqrt{\beta}}C)^T = \mathcal{E}_1 A^{-1}\mathcal{E}_2$. Then we can obtain a modified preconditioner

$$\mathcal{P}_{MS} = \begin{pmatrix} A & \\ & \mathcal{E}_1 A^{-1}\mathcal{E}_2 \end{pmatrix}. \tag{2.5}$$

P_{MS} can be seen as a block diagonal preconditioner with an approximation Schur complement, more detail about this preconditioner can be found in [27].

Based on the approach of [1, 5, 13] and [27], we propose a new $ABD-$like preconditioner for the reduced-order linear system (2.4):

$$\mathcal{P}_{ABD-like} = \begin{pmatrix} \sqrt{\beta}\mathcal{E}_1 & \\ & \frac{1}{\sqrt{\beta}}\mathcal{E}_2 \end{pmatrix}. \tag{2.6}$$

Let

$$E_{12} := (M+\tau K) + \frac{1}{2\sqrt{\beta}}\tau M, \quad E_{21} := (M+\tau K) + \frac{2}{\sqrt{\beta}}\tau M,$$
$$E_{\gamma} = (M+\tau K) + \frac{\gamma}{\sqrt{\beta}}\tau M, \qquad E := (M+\tau K) + \frac{1}{\sqrt{\beta}}\tau M.$$

With the preconditioner $P_{ABD-like}$ defined in (2.6) being applied to the three reduced-order linear systems (2.1)-(2.3), we can obtain immediately the preconditioners $\mathcal{P}_{1new}, \mathcal{P}_{2new}$ and $\mathcal{P}_{3new}$ for the linear systems (2.1), (2.2) and (2.3), respectively. Here

$$\mathcal{P}_{1new} = \text{blkdiag}(\underbrace{\sqrt{\beta}E_{12}, \sqrt{\beta}E, \cdots, \sqrt{\beta}E, \sqrt{\beta}E_{12}}_{N_t}, \underbrace{\frac{1}{\sqrt{\beta}}E_{21}, \frac{1}{\sqrt{\beta}}E, \cdots, \frac{1}{\sqrt{\beta}}E, \frac{1}{\sqrt{\beta}}E_{21}}_{N_t}),$$
$$\mathcal{P}_{2new} = \text{blkdiag}(\underbrace{\sqrt{\beta}E, \sqrt{\beta}E, \cdots, \sqrt{\beta}E, \sqrt{\beta}E_{\gamma}}_{N_t}, \underbrace{\frac{1}{\sqrt{\beta}}E_{21}, \frac{1}{\sqrt{\beta}}E, \cdots, \frac{1}{\sqrt{\beta}}E, \frac{1}{\sqrt{\beta}}E_{21}}_{N_t}),$$
$$\mathcal{P}_{3new} = \text{blkdiag}(\underbrace{\sqrt{\beta}E_{\gamma}, \sqrt{\beta}E_{\gamma}, \cdots, \sqrt{\beta}E_{\gamma}, \sqrt{\beta}E}_{N_t}, \underbrace{\frac{1}{\sqrt{\beta}}E_{21}, \frac{1}{\sqrt{\beta}}E, \cdots, \frac{1}{\sqrt{\beta}}E, \frac{1}{\sqrt{\beta}}E_{21}}_{N_t}).$$

3. Spectral Analysis of the Reduced-Order Linear Systems

In this section, we will analyze the spectral properties of the reduced-order and the preconditioned reduced-order linear system. Some theoretical bounds on the spectrum of the preconditioned matrix are established.

First some basic notations and definition will be introduced. Let A and $C \in \mathbb{R}^{(nN_t)\times(nN_t)}$ be SPD matrices and $B \in \mathbb{R}^{(nN_t)\times(nN_t)}$ be block lower-triangular matrix with symmetric positive definite diagonal blocks. Define $S := B^TB$.

Here and in the sequel, $\lambda(\cdot)$, $\alpha.$, $\beta.$ and $\lambda.$ denote the eigenvalue, the minimum eigenvalue, the maximum eigenvalue and the Rayleigh quotients of the matrix $(\cdot)$, respectively. sp$(\cdot)$ denotes the set of the eigenvalues of the matrix $(\cdot)$. Suppose that

$$\alpha_A \le \lambda_A \le \beta_A, \quad \alpha_C \le \lambda_C \le \beta_C, \quad \alpha_S \le \lambda_S \le \beta_S.$$

First, we consider the spectral properties for the general form of the reduced-order linear systems (2.4). The result straightforwardly follows from [31] or the Proposition 2.1 in [32].

Theorem 3.1. *Assume that the matrix $\mathcal{A}$ is the coefficient matrix defined in* (2.4)*, then*

$$sp(\mathcal{A}) \subseteq \mathcal{I}^- \cup \mathcal{I}^+,$$

where

$$\mathcal{I}^- = \left[\frac{1}{2}(-\frac{1}{\beta}\beta_A - \sqrt{\frac{1}{\beta^2}\beta_A^2 + 4\beta_S}, \ \ \frac{1}{2}(\beta_A - \sqrt{\beta_A^2 + 4\alpha_S})\right],$$
$$\mathcal{I}^+ = \left[\frac{1}{2}(-\frac{1}{\beta}\beta_A + \sqrt{\frac{1}{\beta^2}\beta_A^2 + 4\alpha_S}, \ \ \frac{1}{2}(\beta_A + \sqrt{\beta_A^2 + 4\beta_S})\right].$$

From Theorem 3.1, we can see that the eigenvalues of the coefficient matrix before preconditioning is dependent of $O(\frac{1}{\beta})$, so they increase rapidly when $\beta \ll 1$. Therefore a preconditioning technique must be taken to change the ill-conditioning property of the linear system. Next we give the spectral properties of the preconditioned matrix $\mathcal{P}_{ABD-like}^{-1}\mathcal{A}$.

We give bounds about the preconditioned matrices for the preconditioned linear systems (2.1), (2.2) and (2.3), respectively.

Theorem 3.2. *Let μ be the eigenvalue of the preconditioned matrix $\mathcal{P}_{ABD-like}^{-1}\mathcal{A}$, $\mathcal{A}$ and $\mathcal{P}_{ABD-like}$ be defined in (2.4) and (2.6), respectively. Assume that the eigenvector corresponding to the eigenvalue μ is $(x^T, y^T)^T$. Then*

$$\mu = \frac{1}{2}\Big(\frac{\tilde{b}}{a} - \frac{b}{a}\Big) \pm \frac{1}{2}\sqrt{\Big(\frac{\tilde{b}}{a} - \frac{b}{a}\Big)^2 + 4\Big(\frac{1}{\beta}\frac{c}{a} + \frac{1}{a}\Big)}, \tag{3.1}$$

where

$$\begin{cases} a := \dfrac{y^T B^{-T}\mathcal{E}_1 B^{-1}\mathcal{E}_2 y}{y^T y}, & b := \dfrac{y^T B^{-T}\mathcal{E}_1 B^{-1} C y}{y^T y}, \\ c := \dfrac{y^T B^{-T} A B^{-1} C y}{y^T y}, & \tilde{b} := \dfrac{y^T B^{-T} A B^{-1}\mathcal{E}_2 y}{y^T y}. \end{cases} \tag{3.2}$$

Proof. Noting the properties of the matrices A, B and C, we know that $x \neq 0$ and $y \neq 0$. Then we have

$$\begin{cases} Ax + B^T y = \sqrt{\beta}\mu\mathcal{E}_1 x, \\ Bx - \frac{1}{\beta}Cy = \mu\frac{1}{\sqrt{\beta}}\mathcal{E}_2 y. \end{cases}$$

By substituting $x = \frac{1}{\beta}B^{-1}(\mu\sqrt{\beta}\mathcal{E}_2 + C)y$ into the first equation, after some simple algebra computations, we have

$$\mu^2\frac{y^T B^{-T}\mathcal{E}_1 B^{-1}\mathcal{E}_2 y}{y^T y} + \frac{\mu}{\sqrt{\beta}}\left(\frac{y^T B^{-T}\mathcal{E}_1 B^{-1} C y}{y^T y} - \frac{y^T B^{-T} A B^{-1}\mathcal{E}_2 y}{y^T y}\right) - \left(\frac{1}{\beta}\frac{y^T B^{-T} A B^{-1} C y}{y^T y} + 1\right) = 0.$$

Using the definitions of a, b, $\tilde{b}$ and c in (3.2), we can rewrite the above equation as

$$\mu^2 a + \frac{\mu}{\sqrt{\beta}}(b - \tilde{b}) - \Big(\frac{1}{\beta}c + 1\Big) = 0.$$

The roots of the above equation gives (3.1). □

Theorem 3.3. *Let A be one of the matrices A_1, A_2 or A_3, where A_1, A_2, A_3 and A are defined in (2.1), (2.2), (2.3) and (2.4), respectively. Let μ be an eigenvalue of the preconditioned matrix $\mathcal{P}_{ABD-like}^{-1}\mathcal{A}$ with $\mathcal{A}$ being defined in (2.4). Then*

$$\mu \in \left[\frac{1}{2}\Big(-\frac{1}{\beta}b_2 - \sqrt{\frac{1}{\beta^2}b_2^2 + 4d_2}\Big), \frac{1}{2}\Big(a_2 - \sqrt{a_2^2 + 4c_2}\Big)\right] \bigcup \left[\frac{1}{2}\Big(-\frac{1}{\beta}b_2 + \sqrt{\frac{1}{\beta^2}b_2^2 + 4c_2}\Big), \frac{1}{2}\Big(a_2 + \sqrt{a_2^2 + 4d_2}\Big)\right],$$

where

$$a_2 = \frac{1}{1+\sqrt{\beta}(\frac{1}{\tau}+\frac{\alpha_K}{\beta_M})}, \qquad b_2 = \frac{\beta}{1+\frac{\sqrt{\beta}}{2\tau}(1+\tau\frac{\alpha_K}{\beta_M})},$$
$$c_2 = \frac{\alpha_S}{(\tau\beta_M+\sqrt{\beta}(\beta_M+\tau\beta_K))(\frac{2\tau}{\beta}\beta_M+\frac{1}{\sqrt{\beta}}(\beta_M+\tau\beta_K))},$$
$$d_2 = \frac{\beta_S}{(r\tau\alpha_M+\sqrt{\beta}(\alpha_M+\tau\alpha_K))(\frac{\tau}{\beta}\alpha_M+\frac{1}{\sqrt{\beta}}(\alpha_M+\tau\alpha_K))}.$$

Here $r=\frac{1}{2}$ *when* $A=A_1$, *and* $r=\gamma$ *when* $A=A_2$ *or* A_3.

Proof. Denote by

$$\tilde{\mathcal{A}} := \mathcal{P}_{ABD-like}^{-\frac{1}{2}}\mathcal{A}\mathcal{P}_{ABD-like}^{-\frac{1}{2}} = \begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & -\frac{1}{\beta}\tilde{C} \end{pmatrix} = \begin{pmatrix} P^{-\frac{1}{2}}AP^{-\frac{1}{2}} & P^{-\frac{1}{2}}B^TR^{-\frac{1}{2}} \\ R^{-\frac{1}{2}}BP^{-\frac{1}{2}} & -\frac{1}{\beta}R^{-\frac{1}{2}}CR^{-\frac{1}{2}} \end{pmatrix},$$

where

$$\mathcal{P}_{ABD-like} = \begin{pmatrix} P & \\ & R \end{pmatrix} = \begin{pmatrix} \sqrt{\beta}\mathcal{E}_1 & \\ & \frac{1}{\sqrt{\beta}}\mathcal{E}_2 \end{pmatrix}.$$

From Theorem 3.1, we can get bounds for the eigenvalues of the preconditioned matrix $\mathcal{P}_{ABD-like}^{-1}\mathcal{A}$ as $\tilde{\mathcal{I}^-}\cup\tilde{\mathcal{I}^+}$, where

$$\begin{aligned} \tilde{\mathcal{I}}^- &= \left[\frac{1}{2}\Big(-\frac{1}{\beta}\beta_{\tilde{C}}-\sqrt{\frac{1}{\beta^2}\beta_{\tilde{C}}^2+4\beta_{\tilde{S}}}\Big),\ \frac{1}{2}\Big(\beta_{\tilde{A}}-\sqrt{\beta_{\tilde{A}}^2+4\alpha_{\tilde{S}}}\Big)\right], \\ \tilde{\mathcal{I}}^+ &= \left[\frac{1}{2}\Big(-\frac{1}{\beta}\beta_{\tilde{C}}+\sqrt{\frac{1}{\beta^2}\beta_{\tilde{C}}^2+4\alpha_{\tilde{S}}}\Big),\ \frac{1}{2}\Big(\beta_{\tilde{A}}+\sqrt{\beta_{\tilde{A}}^2+4\beta_{\tilde{S}}}\Big)\right]. \end{aligned} \tag{3.3}$$

In order to obtain bounds for the eigenvalues of the preconditioned matrix corresponding to $A=A_1$ or $A=A_2$ (or A_3), we first compute the largest eigenvalues of $\tilde{A}$ and $\tilde{C}$, and the largest and smallest eigenvalues of the matrix $\tilde{S}=\tilde{B}^T\tilde{B}$.

When $A=A_1$, $r=\frac{1}{2}$, we have $P=\sqrt{\beta}\mathcal{E}_1$ and $R=\frac{1}{\sqrt{\beta}}\mathcal{E}_2$. Then

$$\begin{aligned} \frac{y^T\tilde{A}y}{y^Ty} &= \frac{x^TAx}{x^T(A+\sqrt{\beta}B_d)x} = \frac{1}{1+\sqrt{\beta}\cdot\frac{x^TB_dx}{x^TAx}} \le \frac{1}{1+\sqrt{\beta}(\frac{1}{\tau}+\frac{\alpha_K}{\beta_M})} := \beta_{\tilde{A}}, \\ \frac{y^T\tilde{C}y}{y^Ty} &= \frac{1}{\frac{1}{\beta}+\frac{1}{\sqrt{\beta}}\cdot\frac{x^TB_dx}{x^TCx}} \le \frac{\beta}{1+\frac{\sqrt{\beta}}{2\tau}(1+\tau\frac{\alpha_K}{\beta_M})} := \beta_{\tilde{C}}. \end{aligned}$$

Noticing that

$$B^T(\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-1}B \sim (\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-\frac{1}{2}}BB^T(\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-\frac{1}{2}},$$

we see that

$$\begin{aligned} \frac{y^T\tilde{B}^T\tilde{B}y}{y^Ty} &= \frac{x^TB^T(\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-1}Bx}{x^T(A+\sqrt{\beta}B_d)x} \\ &= \frac{p^T(\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-\frac{1}{2}}BB^T(\frac{1}{\beta}C+\frac{1}{\sqrt{\beta}}B_d)^{-\frac{1}{2}}p}{p^Tp}\cdot\frac{x^Tx}{x^T(A+\sqrt{\beta}B_d)x} \end{aligned}$$

$$
\begin{aligned}
&= \frac{z^T B B^T z}{z^T z} \frac{z^T z}{z^T(\frac{1}{\beta} C + \frac{1}{\sqrt{\beta}} B_d) z} \frac{x^T x}{x^T (A + \sqrt{\beta} B_d) x} \\
&\leq \frac{\beta_S}{(r\tau\alpha_M + \sqrt{\beta}(\alpha_M + \tau\alpha_K))(\frac{\tau}{\beta}\alpha_M + \frac{1}{\sqrt{\beta}}(\alpha_M + \tau\alpha_K))} := \beta_{\tilde{S}},
\end{aligned}
$$

and

$$
\frac{y^T \tilde{B}^T \tilde{B} y}{y^T y} \geq \frac{\alpha_S}{(\tau\beta_M + \sqrt{\beta}(\beta_M + \tau\beta_K))(\frac{2\tau}{\beta}\beta_M + \frac{1}{\sqrt{\beta}}(\beta_M + \tau\beta_K))} := \alpha_{\tilde{S}}.
$$

Substituting $\beta_{\tilde{A}} := a_2$, $\beta_{\tilde{C}} := b_2$, $\alpha_{\tilde{S}} := c_2$ and $\beta_{\tilde{S}} := d_2$ into (3.3) gives the designed result. □

For the case $A = A_2$ or A_3, we only need to replace $r = \frac{1}{2}$ by $r = \gamma$ in the above proof.

4. Numerical Experiments

In this section, some numerical examples are given to support the theoretical results. We examine our new strategy by using the following time-dependent PDE-constrained optimization problem, which is used in [27]:

$$
\begin{aligned}
&\min_{y,u} J(y,u), \\
&\text{subject to} \begin{cases} \frac{\partial y}{\partial t} - \nabla^2 y = u, & \text{for } (x,t) \in \Omega \times (0,T), \\ y = f, & \text{on } \partial\Omega \times (0,T), \\ y = y_0, & \text{at } t = 0. \end{cases}
\end{aligned} \tag{4.1}
$$

The experiments are performed for $T = 1$ and $\tau = 0.05$, i.e., 20 time-steps, the domain considered $\Omega = [0,1]^2$ is a unit square, the desired state is given by

$$
\bar{y}(x_1, x_2) = \begin{cases} (2x_1 - 1)^2 (2x_2 - 1)^2, & \text{if } (x_1, x_2) \in [0, \frac{1}{2}]^2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}
$$

The zero Dirichlet boundary condition for the distributed control problem, with $f = 0$ and $y_0 = 1$, is considered in all the experiments.

In our implementations, all iteration processes are terminated once the Euclidean norms of the current relative residuals are reduced by a factor of 10^{-4} from those of the initial residuals. The relative residual error (denoted as err) is defined as

$$
err = \frac{\|g - \mathcal{A} z^{(k)}\|_2}{\|g - \mathcal{A} z^{(0)}\|_2}.
$$

Table 4.1: Information about the experimental environments.

Hardware and Software	Details
Computer	Microsoft Window XP, Professional, Service Pack 3
	AMD Phenom(tm) II X4 830 Processor
	$2.79GHz, Memory\ 3.00GB$
Version of Matlab	R2009b(7.9.0.529)
	32-bit(win32)

We also use the degree of freedom (**DoF**) to represent the order of matrix tested in our performance, IT is the number of iteration steps and CPU is the elapsed CPU time in seconds. $\overline{P}_S$ and $\overline{P}_{MS}$ are the preconditioners for the original linear system (before order-reduced):

$$\overline{P}_S = \begin{pmatrix} \tau\mathcal{H} & & \\ & \beta\tau\mathcal{M}_{1/2} & \\ & & \frac{1}{\tau}\mathcal{K}\mathcal{H}^{-1}\mathcal{K} + \frac{\tau}{\beta}\mathcal{M}_{1,1}(\mathcal{M}_{1/2})^{-1}\mathcal{M}_{1,1} \end{pmatrix},$$

$$\overline{P}_{MS} = \begin{pmatrix} \tau\mathcal{H} & & \\ & \beta\tau\mathcal{M}_{1/2} & \\ & & \frac{1}{\tau}(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1,1})\mathcal{H}^{-1}(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1,1}) \end{pmatrix},$$

where $\mathcal{H}$ is referred to $\mathcal{M}_{1/2}$ in Example 4.1, $\mathcal{M}_{1,\gamma}$ in Example 4.2, and $\mathcal{M}_{\gamma,1}$ in Example 4.3, respectively. For details about this example, we refer to [19, 27].

Example 4.1. The time-dependent distributed control problem is defined by (4.1)-(4.2).

In Example 4.1, we minimize $J = J_1$ with **discretize-then-optimize** approach. We refer to [19, 27] for a more detailed description about this example. The problem is discretized in time using a backward Euler implicit time-stepping method with N_t time steps of size τ, which results in the linear system (1.2).

Example 4.2. The time-dependent distributed control problem is the same as Example 4.1.

In Example 4.2, we use the **optimize-then-discretize** approach ([27]) with $J(y,u) = J_1(y,u)$ and discretize in time using a backward Euler implicit time-stepping method with N_t time steps of size τ, which leads to the linear system (1.3).

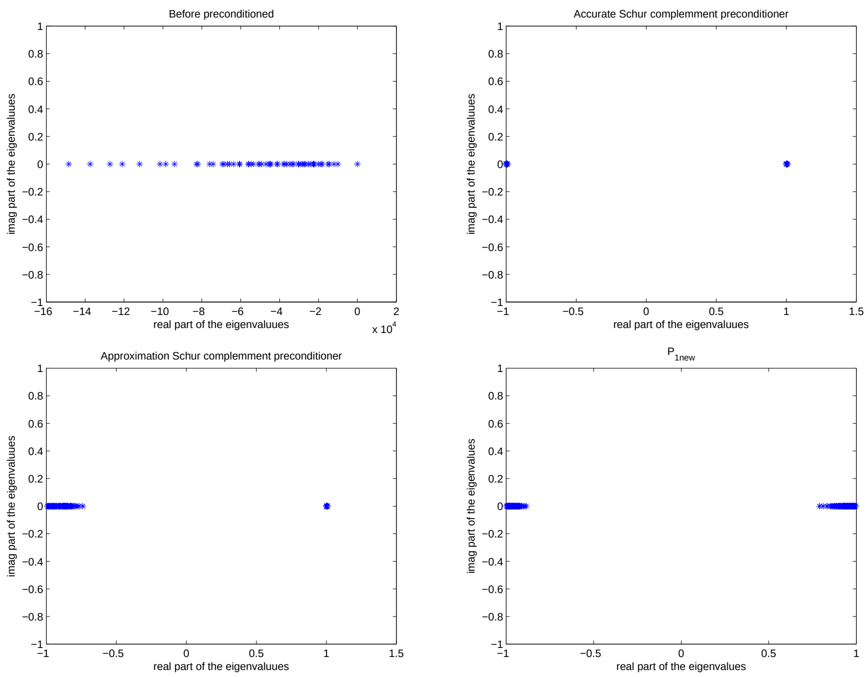

Fig. 4.1. Eigenvalue distributions for Example 4.1 with $\beta = 10^{-8}$. Before preconditioning(top left corner), preconditioned by $\mathcal{P}_S$(top right corner), $\mathcal{P}_{MS}$ (left bottom), $\mathcal{P}_{1new}$ (right bottom).

Table 4.2: Preconditioned by $\overline{P}_S$, $\overline{P}_{MS}$ and the new preconditioning strategy for Example 4.1.

DoF	β		$\overline{P}_S$	$\overline{P}_{MS}$	$\mathcal{P}_{1new}$
2940	10^{-2}	IT	2	14	19
		CPU	0.15	0.61	0.06
	10^{-4}	IT	2	20	12
		CPU	0.15	0.84	0.04
	10^{-6}	IT	2	26	13
		CPU	0.15	1.04	0.04
13500	10^{-2}	IT	2	14	16
		CPU	3.24	13.21	0.32
	10^{-4}	IT	2	22	13
		CPU	3.24	19.76	0.27
	10^{-6}	IT	2	26	12
		CPU	3.29	22.73	0.25

Example 4.3. The time-dependent distributed control problem is described as the above. For this Example, applying the **discretize-then-optimize** formulation with $J(y,u) = J_2(y,u)$ approach, by the same discretizing way as Example 4.1, we obtain the linear system (1.4).

Eigenvalue distributions of the preconditioned matrices are very important to analyze the

Table 4.3: Numerical results with different β for Example 4.1.

DoF	β		$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{1new}$
2940	10^{-2}	IT	6	10	19
		CPU	0.28	0.40	0.06
	10^{-4}	IT	13	13	12
		CPU	0.53	0.51	0.04
	10^{-6}	IT	10	15	13
		CPU	0.43	0.58	0.04
	10^{-10}	IT	1	4	4
		CPU	0.10	0.20	0.02
13500	10^{-2}	IT	6	10	16
		CPU	6.34	9.62	0.32
	10^{-4}	IT	13	13	13
		CPU	13.27	11.90	0.27
	10^{-6}	IT	13	17	12
		CPU	12.27	15.50	0.25
	10^{-10}	IT	3	4	6
		CPU	3.99	4.90	0.14

Table 4.4: Numerical results for larger DoF for Example 4.1.

DoF	$\mathcal{P}_{1new}$	$\beta = 10^{-2}$	$\beta = 10^{-4}$	$\beta = 10^{-8}$
57660	IT	15	12	13
	CPU	1.89	1.55	1.66
238140	IT	13	12	13
	CPU	10.15	9.49	10.17
253500	IT	16	13	12
	CPU	33.62	21.11	19.32

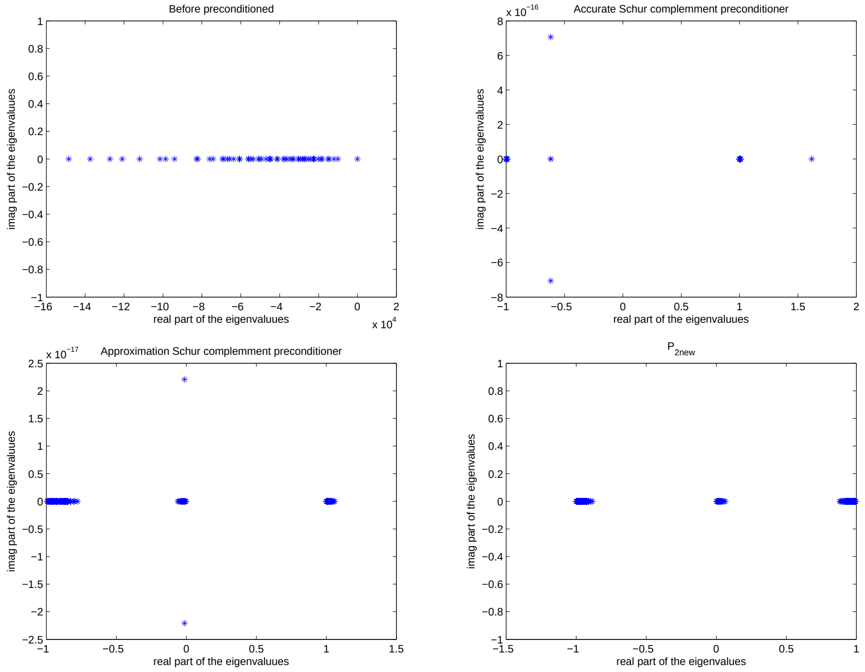

Fig. 4.2. Eigenvalue distributions for Example 4.2 with $\beta = 10^{-8}, \gamma = \tau\beta$ and DoF=2940). Before preconditioning(top left corner), preconditioned by $\mathcal{P}_S$(top right corner), $\mathcal{P}_{MS}$ (left bottom), $\mathcal{P}_{2new}$ (right bottom).

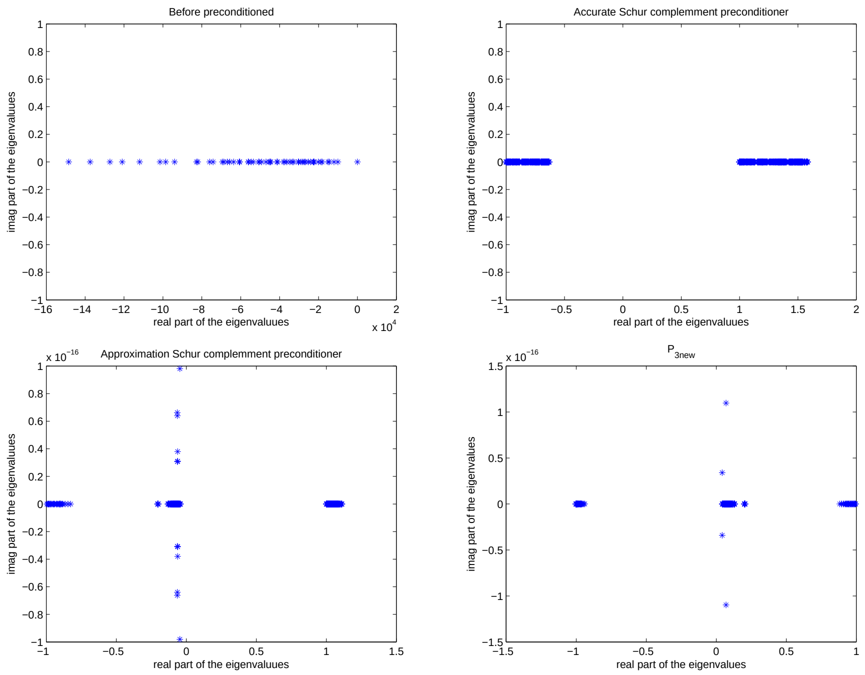

Fig. 4.3. Eigenvalue distributions for Example 4.3 with $\beta = 10^{-8}, \gamma = 10^{-3}$. Before preconditioning(top left corner), preconditioned by $\mathcal{P}_S$(top right corner), $\mathcal{P}_{MS}$ (left bottom), $\mathcal{P}_{3new}$ (right bottom).

Table 4.5: Preconditioned by $\overline{P}_S$, $\overline{P}_{MS}$ and the new preconditioning strategy for Example 4.2.

DoF	β	$\gamma = 0.5$	$\overline{P}_S$	$\overline{P}_{MS}$	$\mathcal{P}_{2new}$
2940	10^{-2}	IT	2	22	19
		CPU	0.15	0.90	0.06
	10^{-4}	IT	2	16	12
		CPU	0.15	0.68	0.05
	10^{-6}	IT	2	26	13
		CPU	0.14	1.04	0.04
13500	10^{-2}	IT	2	16	16
		CPU	3.24	14.89	0.33
	10^{-4}	IT	2	24	13
		CPU	3.19	21.09	0.28
	10^{-6}	IT	2	26	12
		CPU	3.25	22.92	0.26

Table 4.6: Numerical results with different γ for Example 4.2.

DoF : 2940	$\beta = 10^{-4}$	$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{2new}$
$\gamma = \tau$	IT	14	21	19
	CPU	0.54	0.34	0.061
$\gamma = \tau\beta$	IT	14	35	20
	CPU	0.55	0.55	0.07
$\gamma = \beta^2$	IT	14	42	20
	CPU	0.55	0.73	0.08
$\gamma = (\tau\beta)^2$	IT	14	46	20
	CPU	0.55	0.69	0.06

Table 4.7: Numerical results with different γ for Example 4.2.

DoF:13500	$\beta = 10^{-4}$	$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{2new}$
$\gamma = \tau\beta$	IT	14	37	22
	CPU	12.85	9.15	0.44
$\gamma = \beta^2$	IT	14	53	22
	CPU	12.92	12.61	0.44
$\gamma = (\tau\beta)^2$	IT	14	44	22
	CPU	12.01	10.43	0.44

Table 4.8: Numerical results with different β for Example 4.2.

DoF:13500	$\gamma = \tau$	$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{2new}$
$\beta = 10^{-2}$	IT	7	13	16
	CPU	6.31	3.23	0.33
$\beta = 10^{-4}$	IT	14	21	17
	CPU	12.87	5.39	0.35
$\beta = 10^{-6}$	IT	17	26	19
	CPU	14.57	6.67	0.40
$\beta = 10^{-10}$	IT	4	17	12
	CPU	3.98	4.28	0.26

Table 4.9: Numerical results for larger DoF using new preconditioning strategy for Example 4.2.

$\mathcal{P}_{2new}, DoF$	$\gamma=\tau$	$\beta=10^{-2}$	$\beta=10^{-4}$	$\beta=10^{-8}$
57660	IT	15	17	22
	CPU	1.90	2.11	2.68
238140	IT	13	17	21
	CPU	10.22	12.93	15.73
253500	IT	13	17	22
	CPU	33.56	35.11	53.32

Table 4.10: Preconditioned by $\overline{P}_S$, $\overline{P}_{MS}$ and the new preconditioning strategy for Example 4.3.

DoF	β	$\gamma=0.5$	$\overline{P}_S$	$\overline{P}_{MS}$	$\mathcal{P}_{3new}$
2940	$\beta=10^{-2}$	IT	2	16	19
		CPU	0.15	0.66	0.06
	$\beta=10^{-4}$	IT	2	28	12
		CPU	0.15	1.12	0.04
	$\beta=10^{-6}$	IT	2	32	11
		CPU	0.15	1.27	0.03
13500	$\beta=10^{-2}$	IT	2	14	17
		CPU	3.33	13.24	0.34
	$\beta=10^{-4}$	IT	2	24	12
		CPU	3.30	21.20	0.27
	$\beta=10^{-6}$	IT	2	28	11
		CPU	3.29	24.63	0.24

Table 4.11: Numerical results with different β for Example 4.3.

$\gamma=\tau$	DoF:2940	$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{3new}$
$\beta=10^{-8}$	IT	9	18	15
	CPU	0.37	0.29	0.05
$\beta=10^{-6}$	IT	17	27	18
	CPU	0.66	0.43	0.06
$\beta=10^{-4}$	IT	13	28	20
	CPU	0.51	0.43	0.06
$\beta=10^{-2}$	IT	5	37	19
	CPU	0.24	0.58	0.06

Table 4.12: Numerical results with different β and γ for Example 4.3.

(β,γ)	DoF:2940	$\mathcal{P}_S$	$\mathcal{P}_{MS}$	$\mathcal{P}_{3new}$
$(10^{2}, 0.01)$	IT	1	5	21
	CPU	0.10	0.63	0.07
$(10^{-2}, 0.01)$	IT	21	30	19
	CPU	1.96	2.67	0.07
$(10^{-4}, 0.0001)$	IT	13	43	29
	CPU	0.51	3.85	0.09
$(10^{-10}, 0.001)$	IT	18	25	14
	CPU	1.71	2.21	0.05

Table 4.13: Numerical results for larger DoF using new preconditioning strategy for Example 4.3.

DoF	$\mathcal{P}_{3new}$	$\beta=10^{-8},\gamma=0.05$	$\beta=10^{-8},\gamma=0.1$	$\beta=10^{-8},\gamma=0.5$
13500	IT	20	16	11
	CPU	0.99	0.33	0.24
57660	IT	21	18	13
	CPU	2.63	2.27	1.72
238140	IT	20	17	11
	CPU	15.08	13.03	8.97
253500	IT	20	17	12
	CPU	57.92	50.55	35.90

Table 4.14: Numerical results with the stopping criterion 10^{-6} using new preconditioning strategy.

	γ	β	DoF:	2940	13500	57660	238140
Example 4.1	–	10^{-2}	IT	27	25	24	23
			CPU	0.33	2.09	13.08	84.85
		10^{-4}	IT	20	20	20	18
			CPU	0.25	1.70	11.09	68.71
		10^{-8}	IT	13	17	20	19
			CPU	0.21	1.43	11.07	72.46
Example 4.2	0.05	10^{-2}	IT	27	26	24	23
			CPU	0.33	2.15	13.06	85.75
		10^{-4}	IT	27	28	27	26
			CPU	0.33	2.27	14.58	94.45
		10^{-8}	IT	23	32	32	31
			CPU	0.32	2.64	17.08	102.63
	0.5	10^{-2}	IT	27	25	24	23
			CPU	0.34	2.13	12.58	82.01
		10^{-4}	IT	20	20	20	18
			CPU	0.25	1.71	10.69	65.45
		10^{-8}	IT	12	17	19	19
			CPU	0.21	1.41	10.22	69.75
Example 4.3	0.05	10^{-8}	IT	24	28	29	29
			CPU	0.36	2.49	16.44	110.97
	0.1	10^{-8}	IT	19	24	26	25
			CPU	0.29	1.98	15.18	96.22
	0.5	10^{-8}	IT	13	18	18	18
			CPU	0.22	1.49	10.64	70.63

preconditions properties. In Figs. 4.1-4.3, we depict the eigenvalue distributions in the complex plane.

Fig. 4.1 shows the eigenvalue distributions of the coefficient matrix without preconditioning and preconditioned by P_S, P_{MS} and P_{1new} for Example 4.1, respectively.

Figs. 4.2 and 4.3 depict the eigenvalue distributions of the coefficient matrix without preconditioning and preconditioned by P_S, P_{MS} and P_{2new} or P_{3new} for Example 4.2 or Example 4.3, respectively.

From Theorem 3.3, we can obtain the real intervals of the eigenvalues of the matrices $\mathcal{P}_{1new}^{-1}\mathcal{A}_1$, $\mathcal{P}_{2new}^{-1}\mathcal{A}_2$ and $\mathcal{P}_{3new}^{-1}\mathcal{A}_3$ are $[-1.0692,-2.0048\times10^{-6}]\cup[2.0008\times10^{-6},1.0674]$ (when $r=\frac{1}{2}$), $[-2.1107,-2.0048\times10^{-6}]\cup[2.0008\times10^{-6},2.1094]$ (when $r=\tau\beta$ and $\beta=10^{-8}$) and

Table 4.15: Numerical results with the stopping criterion 10^{-6} preconditioned by P_S or P_{MS}.

	γ	β	DoF:	2940		13500	
			Pre.:	P_S	P_{MS}	P_S	P_{MS}
Example 4.1	–	10^{-2}	IT	8	10	8	29
			CPU	0.48	0.73	13.69	10.98
		10^{-4}	IT	19	20	19	20
			CPU	0.99	0.50	28.35	7.92
		10^{-8}	IT	5	9	10	17
			CPU	0.33	0.26	16.29	7.01
Example 4.2	0.05	10^{-2}	IT	10	29	10	29
			CPU	0.57	0.70	16.44	11.33
		10^{-4}	IT	21	29	21	29
			CPU	1.08	0.68	31.17	11.06
		10^{-8}	IT	9	23	18	32
			CPU	0.53	0.55	29.26	13.12
	0.5	10^{-2}	IT	8	29	8	27
			CPU	0.47	0.68	13.52	10.20
		10^{-4}	IT	19	20	19	20
			CPU	1.01	0.49	28.62	8.12
		10^{-8}	IT	5	9	10	16
			CPU	0.33	0.28	17.28	6.72
Example 4.3	0.05	10^{-8}	IT	13	25	21	36
			CPU	0.73	0.61	31.96	13.93
	0.1	10^{-8}	IT	10	19	19	30
			CPU	0.58	0.47	29.19	11.78
	0.5	10^{-8}	IT	7	10	12	18
			CPU	1.86	0.28	19.83	7.42

$[-2.0691, -2.0048\times10^{-6}]\cup[2.0008\times10^{-6}, 2.0678]$ (when $r = 10^{-3}$ and $\beta = 10^{-8}$), respectively.

From Figs. 4.1-4.3 we see that the experimental results are in agreement with theoretical analysis. In Tables 4.2-4.13, we list the numbers of iteration steps, the computing times with respect to the preconditioners P_S, P_{MS} and $P_{ABD-like}$, which are employed to precondition MINRES with the stopping criterion 10^{-4}. Tables 4.14-4.15 list the numerical results (numbers of iteration steps, the computing times) with respect to the preconditioners P_S, P_{MS} and $P_{ABD-like}$ with the stopping criterion 10^{-6}.

For Examples 4.1-4.3, we can apply $\overline{P}_S$, $\overline{P}_{MS}$, P_S, P_{MS} and P_{1new} (or P_{2new} or P_{3new}) to serve as preconditioners for MINRES.

Table 4.2 shows the numerical results for Example 4.1 using preconditioner $\overline{P}_S$, $\overline{P}_{MS}$ and P_{1new}. Table 4.3 gives the numerical results for Example 4.1 using preconditioner P_S, P_{MS} and P_{1new}. And Table 4.4 list the numerical results for Example 4.1 preconditioned by and P_{1new} for larger DoF.

From Tables 4.1-4.3 we see that the new preconditioning strategy requires larger number of iteration steps in some cases, but costs less computing time. Moreover, the difference in computing time becomes more significant for larger DoF.

Similarly, Tables 4.5-4.9 and Tables 4.10-4.13 show the numerical results for Example 4.2 and Example 4.3, respectively.

From Tables 4.14-4.15 we see that the results with the stopping criterion 10^{-4} can be extended to the case when the stopping criterion is 10^{-4}. Besides, when preconditioned by P_S or P_{MS}, if the matrix dimension DOF is more than 57660, the computer will be out of memory. However, when using new preconditioning strategy preconditioned, it works well.

From Tables 4.2-4.15, we observe that in terms of computing time new preconditioning strategy perform significantly better than the other solvers (such as preconditioned by P_S and P_{MS})in solving these problems. The new preconditioning strategy costs less computing time than other solvers and the difference becomes much greater when the mesh is more refined. And the new preconditioning strategy seems to be more effective for a wider range of regularization parameter values as well as mesh sizes for solving the large sparse system which is discretized from time-dependent PDE-constrained optimization problem.

5. Conclusions

By utilizing the algebraic properties and the sparse structures of the coefficient matrix, we present a new preconditioning strategy for solving the large sparse system arising in the time-dependent PDE-constrained optimization problems. By using a particularly simple and effective reduction, we first obtain the order-reduced structural linear system. Then a new effective preconditioner is proposed for the reduced-order linear system. Spectral analysis of the original and the preconditioned reduced-order linear system are discussed. Numerical examples illustrate that the new preconditioning strategy shows great advantage in the CPU time for solving this kind of problems for a wide range of mesh sizes and regularization parameter.

Acknowledgments. The work was supported by the National Natural Science Foundation of China (11271174). The authors would like to thank the referees for the comments and constructive suggestions, which are valuable in improving the quality of the manuscript.

References

[1] O. Axelsson, M. Neytcheva, B. Ahmad, A comparison of iterative methods to solve complex valued linear algebraic systems, TR 2013-005.

[2] Z.-Z. Bai, Block preconditioners for elliptic PDE-constrained optimization, *Computing*, **91** (2011), 379–395.

[3] Z.-Z. Bai, Structured preconditioners for nonsingular matrices of block two-by-two structures, *Math. Comput.*, **75** (2006), 791–815.

[4] Z.-Z. Bai, Eigenvalue estimates for saddle point matrices of Hermitian and indefinite leading blocks, *J. Comput. Appl. Math.*, **237** (2013), 295–306.

[5] Z.-Z. Bai, F. Chen, Z.-Q. Wang, Additive block diagonal preconditioning for block two-by-two linear systems of skew-Hamiltonian coefficient matrices, *Numer. Algor.*, **62** (2013), 655–675.

[6] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM J. Matrix Anal. Appl.*, **24** (2003), 603–626.

[7] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, *Numer. Math.*, **98** (2004), 1–32.

[8] Z.-Z. Bai, G.-Q. Li, Restrictively preconditioned conjugate gradient methods for systems of linear equations, *IMA J. Numer. Anal.*, **23** (2003), 561–580.

[9] Z.-Z. Bai, B.N. Parlett, Z.-Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, *Numer. Math.*, **102** (2005), 1–38.

[10] Z.-Z. Bai, Z.-Q. Wang, On parameterized inexact Uzawa methods for generalized saddle point problems, *Linear Algebra Appl.*, **428** (2008), 2900–2932.

[11] Z.-Z. Bai, M.K. Ng, Z.-Q. Wang, Constraint preconditioners for symmetric indefinite matrices, *SIAM J. Matrix Anal. Appl.*, **31** (2009), 410–433.

[12] Z.-Z. Bai, M. Benzi, F. Chen, Z.-Q. Wang, 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** (2013), 343–369.

[13] Z.-Z. Bai, On preconditioned iteration methods for complex linear systems, *J. Engrg. Math.*, 2013.

[14] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, **14** (2005), 1–137.

[15] M. Benzi, V. Simoncini, On the eigenvalues of a class of saddle point matrices, *Numer. Math.*, **103** (2006), 173–196.

[16] J.H. Bramble, J.E. Pascisk, Analysis of the inexact Uzawa algorithm for saddle point problems, *Comput. Optim. Appl.*, **34** (1997), 1072–1092.

[17] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer, New York, 1991.

[18] M. Gunzburger, C. Trenchea, Optimal control of the time-periodic MHD equations, *Nonlinear Anal.*, **63** (2005), 1687–1699.

[19] H.C. Elman, A. Ramage, D.J. Silvester, Algorithm 866: IFISS, A Matlab toolbox for modelling incompressible flow, *ACM Trans. Math. Softw.*, **33** (2007).

[20] M. Hinze, Optimization with PDE constraints, Springer, 2009.

[21] M. Kollmann, M. Kolmbauer, U. Langer, M. Wolfmayr, W. Zulehner, A robust finite element solver for a multiharmonic parabolic optimal control problem, *Comput. Math. Appl.*, **65** (2013), 469–486.

[22] M. Kolmbauer, Efficient solvers for multiharmonic eddy current optimal control problems with various constraints and their analysis, *IMA J. Numer. Anal.*, 2012, doi: 10.1093/imanum/drs025.

[23] M.Kollmann, M. Kolmbauer, A preconditioned MinRes solver for time-periodic parabolic optimal control problems, *Numer. Linear Algebra Appl.*, 2012, doi: 10.1002/nla.1842.

[24] M. Kolmbauer, U. Langer, A robust preconditioned Minres solver for distributed time-periodic eddy current optimal control problems, *SIAM J. Sci. Comput.*, **34** (2012), 785–809.

[25] W. Krendl, V. Simoncini, W. Zulehner, Stability estimates and structural spectral properties of saddle point problems, *Numer. Math.*, **124** (2013), 183–213.

[26] J.L. Lions, Optimal control of systems governed by partial differential equations, Berlin: Springer, 1971.

[27] J.W. Pearson, M. Stoll, A.J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, *SIAM J. Matrix Anal. Appl.*, **33** (2012), 1126–1152.

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

[29] T. Rees, M. Stoll, Block-triangular preconditioners for PDE-constrained optimization, *Numer. Linear Algebra Appl.*, **17** (2010), 977–996.

[30] J. Schöberl, W. Zulehner, Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems, *SIAM J. Matrix Anal. Appl.*, **29** (2007), 752–773.

[31] D.J. Silvester, A.J. Wathen, Fast iterative solution of stabilized Stokes systems, Part II: Using general block preconditioners, *SIAM J. Numer. Anal.*, **31** (1994), 1352–1367.

[32] V. Simoncini, Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems, *Compu. Opt. Appl.*, **53** (2012), 591–617.

[33] M. Stoll, A.J. Wathen, All-at-once solution of time-dependent Stokes control, *J. Comput. Phy.*, **232** (2013), 498–515.

[34] F. Tröltzsch, Optimal Control of Partial Differential Equations: Theory, Methods, and Applications, *Amer. Math. Soc.*, 2010.