IMA Journal of Numerical Analysis (2013) **33**, 343–369
doi:10.1093/imanum/drs001
Advance Access publication on March 25, 2012

Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems

ZHONG-ZHI BAI*
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, PO Box 2719*, Beijing* 100190*, People's Republic of China*
*Corresponding author: bzz@lsec.cc.ac.cn

MICHELE BENZI
Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322*, USA*
benzi@mathcs.emory.edu

FANG CHEN
Key Laboratory of Environmental Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190*, People's Republic of China*
chenfreesky@gmail.com

AND

ZENG-QI WANG
Department of Mathematics, Shanghai Jiaotong University, Shanghai 200240*, People's Republic of China*
wangzengqi@sjtu.edu.cn

[Received on 9 January 2011; revised on 1 December 2011]

We construct a *preconditioned modified Hermitian and skew-Hermitian splitting* (PMHSS) iteration scheme for solving and preconditioning a class of block two-by-two linear systems arising from the Galerkin finite element discretizations of a class of distributed control problems. The convergence theory of this class of PMHSS iteration methods is established and the spectral properties of the PMHSS-preconditioned matrix are analysed. Numerical experiments show that the PMHSS preconditioners can be quite competitive when used to precondition Krylov subspace iteration methods such as GMRES.

Keywords: block two-by-two matrices; PMHSS iteration; preconditioning; spectral properties; PDE-constrained optimization; KKT systems.

1. Introduction

Consider block two-by-two systems of linear equations of the form

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

where $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n\times n}$ are real, symmetric and positive semidefinite matrices with at least one of them, e.g., $\mathbf{W}$, being positive definite. This class of linear systems can be formally regarded as a special case of the generalized saddle point problem (Benzi & Golub, 2004; Benzi *et al.*, 2005). It frequently arises from finite element discretizations of elliptic *partial differential equation* (PDE)-constrained optimization problems such as distributed control problems (Lions, 1968; Lass *et al.*, 2009; Rees *et al.*, 2010; Rees & Stoll, 2010; Bai, 2011) and also from real equivalent formulations of complex symmetric linear systems (Axelsson & Kucherov, 2000; Day & Heroux, 2001).

Based on the *Hermitian and skew-Hermitian splitting* (HSS)[1]

$$\mathbf{A} = \mathbf{H} + \mathbf{S}$$

of the matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$, with

$$\mathbf{H} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^*) = \begin{pmatrix} \mathbf{W} & 0 \\ 0 & \mathbf{W} \end{pmatrix} \quad \text{and} \quad \mathbf{S} = \frac{1}{2}(\mathbf{A} - \mathbf{A}^*) = \begin{pmatrix} 0 & -\mathbf{T} \\ \mathbf{T} & 0 \end{pmatrix}$$

being the Hermitian and skew-Hermitian parts and $\mathbf{A}^*$ being the conjugate transpose of the matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$, we can apply the HSS iteration method (Bai *et al.*, 2003) or its preconditioned variant *PHSS* (i.e., the preconditioned HSS; see Bai *et al.*, 2004) to compute an approximate solution of the block two-by-two linear system (1.1); see also Benzi & Golub (2004), Benzi *et al.* (2005) and Bai *et al.* (2007). With this alternating splitting iteration approach, at each step we need to solve two linear subsystems of the following forms:

$$(\alpha \mathbf{V}_{\mathrm{D}} + \mathbf{H})\mathbf{x} = \mathbf{r}^{(1)} \quad \text{and} \quad (\alpha \mathbf{V}_{\mathrm{D}} + \mathbf{S})\mathbf{x} = \mathbf{r}^{(2)},$$

where $\mathbf{V}_{\mathrm{D}} \in \mathbb{R}^{2n\times 2n}$ is a given symmetric positive definite matrix of the block diagonal form $\mathbf{V}_{\mathrm{D}} = \mathrm{Diag}(\mathbf{V}, \mathbf{V})$ with $\mathbf{V} \in \mathbb{R}^{n\times n}$, α is a prescribed positive constant, and $\mathbf{r}^{(j)}$, $j = 1, 2$, are two known vectors. In matrix–vector form, the above linear subsystems can be equivalently written as

$$\begin{pmatrix} \alpha\mathbf{V} + \mathbf{W} & 0 \\ 0 & \alpha\mathbf{V} + \mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_a^{(1)} \\ \mathbf{r}_b^{(1)} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \alpha\mathbf{V} & -\mathbf{T} \\ \mathbf{T} & \alpha\mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_a^{(2)} \\ \mathbf{r}_b^{(2)} \end{pmatrix},$$

where

$$\mathbf{r}^{(j)} = \begin{pmatrix} \mathbf{r}_a^{(j)} \\ \mathbf{r}_b^{(j)} \end{pmatrix}, \quad \text{with} \quad \mathbf{r}_a^{(j)}, \mathbf{r}_b^{(j)} \in \mathbb{R}^n, \quad j = 1, 2.$$

The matrix $\alpha\mathbf{V}+\mathbf{W}$ is symmetric positive definite, so the first linear subsystem may be effectively solved either exactly by the Cholesky factorization or inexactly by some inner iterative scheme. The matrix

$$\begin{pmatrix} \alpha\mathbf{V} & -\mathbf{T} \\ \mathbf{T} & \alpha\mathbf{V} \end{pmatrix}$$

is nonsymmetric and positive definite. The second linear subsystem may be solved in principle through Schur complement reduction (block triangular factorization) by first computing the solution subvector $\mathbf{z}$ from

$$(\alpha^2\mathbf{V} + \mathbf{T}\mathbf{V}^{-1}\mathbf{T})\mathbf{z} = \alpha\mathbf{r}_b^{(2)} - \mathbf{T}\mathbf{V}^{-1}\mathbf{r}_a^{(2)},$$

[1] In the real case this becomes the symmetric and skew-symmetric splitting. In this case we use $(\cdot)^{\mathrm{T}}$ to denote the transpose of the corresponding matrix.

and then computing another solution subvector $\mathbf{y}$ from

$$\mathbf{y} = \frac{1}{\alpha}\mathbf{V}^{-1}(\mathbf{T}\mathbf{z} + \mathbf{r}_{\alpha}^{(2)}).$$

The problem with this approach is that in most cases the Schur complement $\alpha^2\mathbf{V} + \mathbf{T}\mathbf{V}^{-1}\mathbf{T}$ will be a large dense matrix that cannot be easily managed.

The system of linear equations (1.1) may be solved by the preconditioned GMRES method, or it may be first symmetrized and then solved by the preconditioned MINRES method; see Chen (2005) and Saad (2003). If MINRES is used, a symmetric positive definite preconditioner is required. A typical choice is to use a block diagonal preconditioner (Rees *et al.*, 2010; Pearson & Wathen, 2010); such preconditioners necessitate a good approximation to the Schur complement $\mathbf{W} \pm \mathbf{T}\mathbf{W}^{-1}\mathbf{T}$, which may be difficult or expensive to construct (Bai & Ng , 2005). We return to this issue at the end of Section 3.

In this paper we develop a new approach to solving block linear systems of the form (1.1) that avoids Schur complements. The new schemes are based on a class of methods for solving linear systems with complex symmetric matrices introduced in Bai *et al.* (2010, 2011) and exploit the fact that real systems of the form (1.1) are formally identical to linear systems with complex coefficient matrix $\mathbf{A}_c := \mathbf{W}+\mathrm{i}\mathbf{T} \in \mathbb{C}^{n\times n}$, where $\mathrm{i} = \sqrt{-1}$ denotes the imaginary unit. The algorithm described in Bai *et al.* (2010) is a *modified Hermitian and skew-Hermitian splitting* (MHSS) iteration method; the one introduced in Bai *et al.* (2011) is a preconditioned variant of MHSS, called *preconditioned modified Hermitian and skew-Hermitian splitting* (PMHSS).

In the following, we develop a PMHSS iterative method for solving the block two-by-two linear system (1.1), establish its convergence theory and analyse the spectral properties of the corresponding preconditioned matrix. The PMHSS iteration scheme is, in spirit, analogous to the preconditioned HSS iteration methods discussed above (see also Benzi & Golub, 2004; Bai *et al.*, 2004; Bertaccini *et al.*, 2005; Bai *et al.*, 2007, for detailed treatments) for solving non-Hermitian positive definite linear systems. Convergence analysis has shown that the PMHSS iteration method is convergent when both $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite and at least one of them is positive definite. For certain special cases of the PMHSS iteration method, this convergence condition can be weakened to the requirement that both matrices $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite satisfying $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$, where $\mathrm{null}(\cdot)$ represents the null space of the corresponding matrix. Also, the PMHSS iteration method naturally leads to a preconditioning matrix for the block two-by-two matrix $\mathbf{A}$. For the above-mentioned special PMHSS preconditioners, the eigenvalues of the preconditioned matrices are clustered within complex disks centred at 1 with radii $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$, where $\alpha > 0$ is the iteration parameter, and the matrices of the corresponding eigenvectors are unitary and hence have condition numbers equal to 1.

In this paper the PMHSS iteration method is applied to a class of KKT linear systems arising from a finite element discretization of a class of distributed control problems (Lions, 1968; Lass *et al.*, 2009; Rees *et al.*, 2010; Bai, 2011). In order to apply the method, we first eliminate the Lagrange multiplier from the original KKT system to obtain a block two-by-two system, which is then diagonally scaled so as to obtain linear systems of the form (1.1). The PMHSS iteration method for such a scaled block two-by-two linear system is then transformed back to the original variables, resulting in the PMHSS iteration method and, therefore, the PMHSS preconditioner, for the (nonscaled) block two-by-two linear system. Numerical results in Section 4 show that PMHSS iteration methods, when used to precondition Krylov subspace methods such as GMRES (Saad, 2003), lead to rapid convergence and tend to outperform the constraint preconditioners proposed in Rees *et al.* (2010); see also Keller *et al.* (2000), Bai *et al.* (2009) and the references therein for additional information on constraint preconditioning. Furthermore, our experiments show that PMHSS compares favourably with MINRES with block diagonal preconditioning and different Schur complement approximations, such as those proposed in Pearson & Wathen (2010) and Rees *et al.* (2010).

The organization of the paper is as follows. In Section 2 we establish the PMHSS iteration method and analyse its convergence and preconditioning properties. In Section 3 we derive the PMHSS iteration method for a class of block two-by-two linear systems arising from distributed control problems. Numerical results are given in Section 4 to show the effectiveness of the PMHSS preconditioner, including comparisons with a state-of-the-art constraint preconditioner. Finally, in Section 5 we end the paper with some conclusions and remarks.

2. The PMHSS iteration method

The block two-by-two linear system (1.1) can be rewritten as a fixed point equation:

$$\begin{pmatrix} \alpha\mathbf{V}+\mathbf{W} & 0 \\ 0 & \alpha\mathbf{V}+\mathbf{W} \end{pmatrix}\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{V} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{V} \end{pmatrix}\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} + \begin{pmatrix} \mathbf{p} \\ \mathbf{q} \end{pmatrix}.$$

Alternatively, it can also be rewritten into another fixed point equation:

$$\begin{pmatrix} \alpha\mathbf{V}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{V}+\mathbf{T} \end{pmatrix}\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{V} & -\mathbf{W} \\ \mathbf{W} & \alpha\mathbf{V} \end{pmatrix}\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} + \begin{pmatrix} \mathbf{q} \\ -\mathbf{p} \end{pmatrix}.$$

Alternating between these two splittings leads to the (real) PMHSS iteration method for systems of the form (1.1), described as follows. Here and in the sequel, we use $(\cdot)^{\mathrm{T}}$ to denote the transpose of either a vector or a square matrix.

METHOD 2.1 (The PMHSS iteration method (real version)).
Let $(\mathbf{y}^{(0)^{\mathrm{T}}}, \mathbf{z}^{(0)^{\mathrm{T}}})^{\mathrm{T}} \in \mathbb{R}^{2n}$ be an arbitrary initial guess, with $\mathbf{y}^{(0)}, \mathbf{z}^{(0)} \in \mathbb{R}^n$. For $k = 0, 1, 2, \ldots,$ until the sequence of iterates $\{(\mathbf{y}^{(k)^{\mathrm{T}}}, \mathbf{z}^{(k)^{\mathrm{T}}})^{\mathrm{T}}\}_{k=0}^{\infty} \subset \mathbb{R}^{2n}$ converges, compute the next iterate $(\mathbf{y}^{(k+1)^{\mathrm{T}}}, \mathbf{z}^{(k+1)^{\mathrm{T}}})^{\mathrm{T}}$ according to the following procedure:

$$\begin{cases} \begin{pmatrix} \alpha\mathbf{V}+\mathbf{W} & 0 \\ 0 & \alpha\mathbf{V}+\mathbf{W} \end{pmatrix}\begin{pmatrix} \mathbf{y}^{(k+\frac{1}{2})} \\ \mathbf{z}^{(k+\frac{1}{2})} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{V} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{V} \end{pmatrix}\begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{pmatrix} + \begin{pmatrix} \mathbf{p} \\ \mathbf{q} \end{pmatrix}, \\ \begin{pmatrix} \alpha\mathbf{V}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{V}+\mathbf{T} \end{pmatrix}\begin{pmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{V} & -\mathbf{W} \\ \mathbf{W} & \alpha\mathbf{V} \end{pmatrix}\begin{pmatrix} \mathbf{y}^{(k+\frac{1}{2})} \\ \mathbf{z}^{(k+\frac{1}{2})} \end{pmatrix} + \begin{pmatrix} \mathbf{q} \\ -\mathbf{p} \end{pmatrix}, \end{cases} \tag{2.1}$$

where α is a given positive constant and $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a prescribed symmetric positive definite matrix.

Let

$$\mathbf{x}^{(k)} = \begin{pmatrix} \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{pmatrix} \quad \text{and} \quad \mathbf{x}^{(k+\frac{1}{2})} = \begin{pmatrix} \mathbf{y}^{(k+\frac{1}{2})} \\ \mathbf{z}^{(k+\frac{1}{2})} \end{pmatrix}.$$

Then after straightforward derivations, we can reformulate the PMHSS iteration scheme (2.1) into the standard form

$$\mathbf{x}^{(k+1)} = \mathbf{L}(\mathbf{V};\alpha)\mathbf{x}^{(k)} + \mathbf{R}(\mathbf{V};\alpha)\mathbf{g}, \qquad k = 0, 1, 2, \ldots,$$

where

$$\mathbf{L}(\mathbf{V};\alpha)=\begin{pmatrix}\alpha\mathbf{V}+\mathbf{T} & 0\\ 0 & \alpha\mathbf{V}+\mathbf{T}\end{pmatrix}^{-1}\begin{pmatrix}\alpha\mathbf{V} & -\mathbf{W}\\ \mathbf{W} & \alpha\mathbf{V}\end{pmatrix}\begin{pmatrix}\alpha\mathbf{V}+\mathbf{W} & 0\\ 0 & \alpha\mathbf{V}+\mathbf{W}\end{pmatrix}^{-1}\begin{pmatrix}\alpha\mathbf{V} & \mathbf{T}\\ -\mathbf{T} & \alpha\mathbf{V}\end{pmatrix}$$

and

$$\mathbf{R}(\mathbf{V};\alpha)=\alpha\begin{pmatrix}\alpha\mathbf{V}+\mathbf{T} & 0\\ 0 & \alpha\mathbf{V}+\mathbf{T}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{V} & 0\\ 0 & \mathbf{V}\end{pmatrix}\begin{pmatrix}\alpha\mathbf{V}+\mathbf{W} & 0\\ 0 & \alpha\mathbf{V}+\mathbf{W}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ -\mathbf{I} & \mathbf{I}\end{pmatrix},$$

with $\mathbf{I}\in\mathbb{R}^{n\times n}$ being the identity matrix. Note that $\mathbf{L}(\mathbf{V};\alpha)$ is the iteration matrix of the PMHSS iteration method (real version), i.e., Method 2.1.

By straightforward computations, we can also show that the spectral radius of $\mathbf{L}(\mathbf{V};\alpha)$ is bounded by a quantity $\sigma(\alpha)$, which only depends on the iteration parameter α. This fact is precisely stated in the following theorem.

THEOREM 2.2 Let $\mathbf{A}\in\mathbb{R}^{2n\times 2n}$ be the block two-by-two matrix defined as in (1.1), with both $\mathbf{W}\in\mathbb{R}^{n\times n}$ and $\mathbf{T}\in\mathbb{R}^{n\times n}$ being symmetric positive semidefinite matrices, and let α be a positive constant. Then the following statements hold true:

(i) $\mathbf{A}$ is nonsingular if and only if $\operatorname{null}(\mathbf{W})\cap\operatorname{null}(\mathbf{T})=\{0\}$;

(ii) if $\mathbf{W}$ is symmetric positive definite, the spectral radius of the PMHSS iteration matrix $\mathbf{L}(\mathbf{V};\alpha)$ satisfies $\rho(\mathbf{L}(\mathbf{V};\alpha))\leqslant\sigma(\alpha)$, with

$$\sigma(\alpha)\equiv\max_{\widetilde{\lambda}_j\in\operatorname{sp}(\mathbf{V}^{-1}\mathbf{W})}\frac{\sqrt{\alpha^2+\widetilde{\lambda}_j^2}}{\alpha+\widetilde{\lambda}_j}\cdot\max_{\widetilde{\mu}_j\in\operatorname{sp}(\mathbf{V}^{-1}\mathbf{T})}\frac{\sqrt{\alpha^2+\widetilde{\mu}_j^2}}{\alpha+\widetilde{\mu}_j}$$

$$\leqslant\max_{\widetilde{\lambda}_j\in\operatorname{sp}(\mathbf{V}^{-1}\mathbf{W})}\frac{\sqrt{\alpha^2+\widetilde{\lambda}_j^2}}{\alpha+\widetilde{\lambda}_j}<1\quad\forall\alpha>0,$$

i.e., the PMHSS iteration, defined by Method 2.1, converges unconditionally to the unique solution of the block two-by-two linear system (1.1) for any initial guess, where $\operatorname{sp}(\cdot)$ denotes the spectral set of the corresponding matrix;

(iii) for the choice $\alpha_\star=\sqrt{\widetilde{\gamma}_{\min}\widetilde{\gamma}_{\max}}$, with $\widetilde{\gamma}_{\min}$ and $\widetilde{\gamma}_{\max}$ being the smallest and the largest eigenvalues of the matrix $\mathbf{V}^{-1}\mathbf{W}$, it holds that

$$\sigma(\alpha_\star)\leqslant\frac{\sqrt{\kappa_2(\mathbf{V}^{-1}\mathbf{W})+1}}{\sqrt{\kappa_2(\mathbf{V}^{-1}\mathbf{W})}+1}.$$

Proof. We first prove (i).

Necessity: if there exists a nonzero vector $\mathbf{y}\in\operatorname{null}(\mathbf{W})\cap\operatorname{null}(\mathbf{T})$, then it follows from $\mathbf{W}\mathbf{y}=0$ and $\mathbf{T}\mathbf{y}=0$ that $\mathbf{A}\mathbf{x}=0$, where $\mathbf{x}=(\mathbf{y}^{\mathrm{T}},\mathbf{y}^{\mathrm{T}})^{\mathrm{T}}\in\mathbb{R}^{2n}$ is a nonzero vector. This obviously contradicts the assumption that $\mathbf{A}\in\mathbb{R}^{2n\times 2n}$ is nonsingular. Hence, it must hold that $\operatorname{null}(\mathbf{W})\cap\operatorname{null}(\mathbf{T})=\{0\}$.

Sufficiency: let $\mathbf{x}=(\mathbf{y}^{\mathrm{T}},\mathbf{z}^{\mathrm{T}})^{\mathrm{T}}\in\mathbb{R}^{2n}$, with $\mathbf{y},\mathbf{z}\in\mathbb{R}^n$, such that $\mathbf{A}\mathbf{x}=0$, i.e.,

$$\mathbf{W}\mathbf{y}-\mathbf{T}\mathbf{z}=0\quad\text{and}\quad\mathbf{T}\mathbf{y}+\mathbf{W}\mathbf{z}=0.$$

Then we easily have

$$(\mathbf{W}+\mathbf{T})\mathbf{y}+(\mathbf{W}-\mathbf{T})\mathbf{z}=0 \quad \text{and} \quad (\mathbf{T}-\mathbf{W})\mathbf{y}+(\mathbf{W}+\mathbf{T})\mathbf{z}=0. \tag{2.2}$$

Because $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite matrices and $\operatorname{null}(\mathbf{W}) \cap \operatorname{null}(\mathbf{T}) = \{0\}$, we see that $\mathbf{W}+\mathbf{T}$ is a symmetric positive definite matrix. Therefore, by solving for $\mathbf{y}$ from the first equation in (2.2), we get

$$\mathbf{y}=(\mathbf{W}+\mathbf{T})^{-1}(\mathbf{T}-\mathbf{W})\mathbf{z},$$

and by substituting it into the second equation in (2.2), we get

$$[(\mathbf{W}+\mathbf{T})+(\mathbf{T}-\mathbf{W})(\mathbf{W}+\mathbf{T})^{-1}(\mathbf{T}-\mathbf{W})]\mathbf{z}=0.$$

Noting that the matrix $[(\mathbf{W}+\mathbf{T})+(\mathbf{T}-\mathbf{W})(\mathbf{W}+\mathbf{T})^{-1}(\mathbf{T}-\mathbf{W})]$ is symmetric positive definite, we immediately obtain $\mathbf{z}=0$ and, thus, $\mathbf{y}=0$. This shows that $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ must be nonsingular.

To demonstrate (ii), we first introduce the notation

$$\widetilde{\mathbf{W}}=\mathbf{V}^{-\frac{1}{2}}\mathbf{W}\mathbf{V}^{-\frac{1}{2}} \quad \text{and} \quad \widetilde{\mathbf{T}}=\mathbf{V}^{-\frac{1}{2}}\mathbf{T}\mathbf{V}^{-\frac{1}{2}}.$$

Then there exist orthogonal matrices $\widetilde{\mathbf{Q}}_{\mathbf{W}}, \widetilde{\mathbf{Q}}_{\mathbf{T}} \in \mathbb{R}^{n\times n}$, a positive diagonal matrix $\widetilde{\Lambda}_{\mathbf{W}} = \operatorname{diag}(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_n) \in \mathbb{R}^{n\times n}$ and a non-negative diagonal matrix $\widetilde{\Lambda}_{\mathbf{T}} = \operatorname{diag}(\widetilde{\mu}_1, \widetilde{\mu}_2, \ldots, \widetilde{\mu}_n) \in \mathbb{R}^{n\times n}$ such that

$$\widetilde{\mathbf{W}}=\widetilde{\mathbf{Q}}_{\mathbf{W}}\widetilde{\Lambda}_{\mathbf{W}}\widetilde{\mathbf{Q}}_{\mathbf{W}}^{\mathrm{T}} \quad \text{and} \quad \widetilde{\mathbf{T}}=\widetilde{\mathbf{Q}}_{\mathbf{T}}\widetilde{\Lambda}_{\mathbf{T}}\widetilde{\mathbf{Q}}_{\mathbf{T}}^{\mathrm{T}}.$$

By block scaling and matrix similarity, we know that the PMHSS iteration matrix $\mathbf{L}(\mathbf{V};\alpha)$ is similar to the matrix

$$\widetilde{\mathbf{L}}(\alpha) := \begin{pmatrix} \alpha\mathbf{I} & -\widetilde{\mathbf{W}} \\ \widetilde{\mathbf{W}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\mathbf{W}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\mathbf{W}} \end{pmatrix}^{-1} \begin{pmatrix} \alpha\mathbf{I} & \widetilde{\mathbf{T}} \\ -\widetilde{\mathbf{T}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\mathbf{T}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\mathbf{T}} \end{pmatrix}^{-1}.$$

So, it holds that $\rho(\mathbf{L}(\mathbf{V};\alpha)) = \rho(\widetilde{\mathbf{L}}(\alpha))$. It follows from straightforward computations that

$$\begin{aligned}
\rho(\widetilde{\mathbf{L}}(\alpha)) &\leqslant \left\| \begin{pmatrix} \alpha\mathbf{I} & -\widetilde{\mathbf{W}} \\ \widetilde{\mathbf{W}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\mathbf{W}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\mathbf{W}} \end{pmatrix}^{-1} \right\|_2 \left\| \begin{pmatrix} \alpha\mathbf{I} & \widetilde{\mathbf{T}} \\ -\widetilde{\mathbf{T}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\mathbf{T}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\mathbf{T}} \end{pmatrix}^{-1} \right\|_2 \\
&= \left\| \begin{pmatrix} \alpha\mathbf{I} & -\widetilde{\Lambda}_{\mathbf{W}} \\ \widetilde{\Lambda}_{\mathbf{W}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\Lambda}_{\mathbf{W}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\Lambda}_{\mathbf{W}} \end{pmatrix}^{-1} \right\|_2 \left\| \begin{pmatrix} \alpha\mathbf{I} & \widetilde{\Lambda}_{\mathbf{T}} \\ -\widetilde{\Lambda}_{\mathbf{T}} & \alpha\mathbf{I} \end{pmatrix} \begin{pmatrix} \alpha\mathbf{I}+\widetilde{\Lambda}_{\mathbf{T}} & 0 \\ 0 & \alpha\mathbf{I}+\widetilde{\Lambda}_{\mathbf{T}} \end{pmatrix}^{-1} \right\|_2 \\
&= \max_{\widetilde{\lambda}_j \in \operatorname{sp}(\widetilde{\mathbf{W}})} \left| \frac{\alpha \pm \mathrm{i}\,\widetilde{\lambda}_j}{\alpha+\widetilde{\lambda}_j} \right| \cdot \max_{\widetilde{\mu}_j \in \operatorname{sp}(\widetilde{\mathbf{T}})} \left| \frac{\alpha \pm \mathrm{i}\,\widetilde{\mu}_j}{\alpha+\widetilde{\mu}_j} \right| \\
&= \max_{\widetilde{\lambda}_j \in \operatorname{sp}(\mathbf{V}^{-1}\mathbf{W})} \frac{\sqrt{\alpha^2+\widetilde{\lambda}_j^2}}{\alpha+\widetilde{\lambda}_j} \cdot \max_{\widetilde{\mu}_j \in \operatorname{sp}(\mathbf{V}^{-1}\mathbf{T})} \frac{\sqrt{\alpha^2+\widetilde{\mu}_j^2}}{\alpha+\widetilde{\mu}_j} \\
&= \sigma(\alpha).
\end{aligned}$$

Here we have used the facts that $\mathrm{sp}(\widetilde{\mathbf{W}}) = \mathrm{sp}(\mathbf{V}^{-1}\mathbf{W})$ and $\mathrm{sp}(\widetilde{\mathbf{T}}) = \mathrm{sp}(\mathbf{V}^{-1}\mathbf{T})$. Noting that $\mathrm{sp}(\mathbf{V}^{-1}\mathbf{T}) \subseteq [0, +\infty)$, we see that for all $\widetilde{\mu}_j \in \mathrm{sp}(\mathbf{V}^{-1}\mathbf{T})$, $j = 1, 2, \ldots, n$, the bounds

$$\frac{\sqrt{\alpha^2 + \widetilde{\mu}_j^2}}{\alpha + \widetilde{\mu}_j} \leqslant 1, \qquad j = 1, 2, \ldots, n,$$

are valid. Hence, it holds that

$$\sigma(\alpha) \leqslant \max_{\widetilde{\lambda}_j \in \mathrm{sp}(\mathbf{V}^{-1}\mathbf{W})} \frac{\sqrt{\alpha^2 + \widetilde{\lambda}_j^2}}{\alpha + \widetilde{\lambda}_j} < 1.$$

The validity of (iii) follows immediately from the estimate

$$\sigma(\alpha) \leqslant \max_{\widetilde{\gamma}_{\min} \leqslant \widetilde{\lambda} \leqslant \widetilde{\gamma}_{\max}} \frac{\sqrt{\alpha^2 + \widetilde{\lambda}^2}}{\alpha + \widetilde{\lambda}},$$

and the fact that this upper bound attains the minimum at $\alpha_\star = \sqrt{\widetilde{\gamma}_{\min}\widetilde{\gamma}_{\max}}$. By making use of this $\alpha_\star$, we get

$$\sigma(\alpha_\star) \leqslant \frac{\sqrt{\widetilde{\gamma}_{\min} + \widetilde{\gamma}_{\max}}}{\sqrt{\widetilde{\gamma}_{\min}} + \sqrt{\widetilde{\gamma}_{\max}}} = \frac{\sqrt{\kappa_2(\mathbf{V}^{-1}\mathbf{W}) + 1}}{\sqrt{\kappa_2(\mathbf{V}^{-1}\mathbf{W})} + 1}.$$

. □

In addition, if we introduce matrices

$$\mathbf{F}(\mathbf{V};\alpha) = \mathbf{P}(\alpha)\begin{pmatrix} \alpha\mathbf{V} + \mathbf{W} & 0 \\ 0 & \alpha\mathbf{V} + \mathbf{W} \end{pmatrix}\begin{pmatrix} \mathbf{V}^{-1} & 0 \\ 0 & \mathbf{V}^{-1} \end{pmatrix}\begin{pmatrix} \alpha\mathbf{V} + \mathbf{T} & 0 \\ 0 & \alpha\mathbf{V} + \mathbf{T} \end{pmatrix} \tag{2.3}$$

and

$$\mathbf{G}(\mathbf{V};\alpha) = \mathbf{P}(\alpha)\begin{pmatrix} \alpha\mathbf{V} & -\mathbf{W} \\ \mathbf{W} & \alpha\mathbf{V} \end{pmatrix}\begin{pmatrix} \mathbf{V}^{-1} & 0 \\ 0 & \mathbf{V}^{-1} \end{pmatrix}\begin{pmatrix} \alpha\mathbf{V} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{V} \end{pmatrix},$$

with

$$\mathbf{P}(\alpha) = \frac{1}{2\alpha}\begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{pmatrix}, \tag{2.4}$$

then it holds that

$$\mathbf{A} = \mathbf{F}(\mathbf{V};\alpha) - \mathbf{G}(\mathbf{V};\alpha) \quad \text{and} \quad \mathbf{L}(\mathbf{V};\alpha) = \mathbf{F}(\mathbf{V};\alpha)^{-1}\mathbf{G}(\mathbf{V};\alpha). \tag{2.5}$$

Therefore, the PMHSS iteration scheme (2.1) is induced by the matrix splitting $\mathbf{A} = \mathbf{F}(\mathbf{V};\alpha) - \mathbf{G}(\mathbf{V};\alpha)$ defined in (2.5). It follows that the splitting matrix $\mathbf{F}(\mathbf{V};\alpha)$ given in (2.3) can be used as a preconditioning matrix for the block two-by-two matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ in (1.1), which will be referred to as the PMHSS preconditioner.

In actual implementations, the action of the preconditioning matrix $\mathbf{F}(\mathbf{V};\alpha)$, when used to precondition Krylov subspace methods, is often realized through solving a sequence of generalized residual equations of the form

$$\mathbf{F}(\mathbf{V};\alpha)\mathbf{v} = \mathbf{r},$$

where $\mathbf{r} = (\mathbf{r}_a^{\mathrm{T}}, \mathbf{r}_b^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{2n}$, with $\mathbf{r}_a, \mathbf{r}_b \in \mathbb{R}^n$, represents the current residual vector, while $\mathbf{v} = (\mathbf{v}_a^{\mathrm{T}}, \mathbf{v}_b^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{2n}$, with $\mathbf{v}_a, \mathbf{v}_b \in \mathbb{R}^n$, represents the generalized residual vector. By making use of the concrete structures of the matrices $\mathbf{F}(\mathbf{V};\alpha)$ in (2.3) and $\mathbf{P}(\alpha)$ in (2.4), we obtain the following procedure for computing the vector $\mathbf{v}$:

(i) compute $\widetilde{\mathbf{v}}_a, \widetilde{\mathbf{v}}_b \in \mathbb{R}^n$ by solving the systems of linear equations

$$\begin{cases} (\alpha\mathbf{V} + \mathbf{W})\widetilde{\mathbf{v}}_a = \alpha(\mathbf{r}_a + \mathbf{r}_b), \\ (\alpha\mathbf{V} + \mathbf{W})\widetilde{\mathbf{v}}_b = \alpha(-\mathbf{r}_a + \mathbf{r}_b); \end{cases}$$

(ii) compute $\widehat{\mathbf{v}}_a, \widehat{\mathbf{v}}_b \in \mathbb{R}^n$ by

$$\widehat{\mathbf{v}}_a = \mathbf{V}\widetilde{\mathbf{v}}_a \quad \text{and} \quad \widehat{\mathbf{v}}_b = \mathbf{V}\widetilde{\mathbf{v}}_b;$$

(iii) compute $\mathbf{v}_a, \mathbf{v}_b \in \mathbb{R}^n$ by solving the systems of linear equations

$$\begin{cases} (\alpha\mathbf{V} + \mathbf{T})\mathbf{v}_a = \widehat{\mathbf{v}}_a, \\ (\alpha\mathbf{V} + \mathbf{T})\mathbf{v}_b = \widehat{\mathbf{v}}_b. \end{cases}$$

Note that both matrices $\alpha\mathbf{V} + \mathbf{W}$ and $\alpha\mathbf{V} + \mathbf{T}$ are symmetric positive definite. Hence, the above four systems of linear equations can be solved effectively either exactly by Cholesky factorizations or inexactly by some conjugate gradient or multigrid scheme; see Bai *et al.* (2003, 2008).

In particular, when $\mathbf{V} = \mathbf{W}$, we have

$$\mathbf{L}(\alpha) := \mathbf{L}(\mathbf{W};\alpha) = \frac{1}{\alpha+1}\begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix}^{-1}\begin{pmatrix} \alpha\mathbf{W} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{W} \end{pmatrix}\begin{pmatrix} \alpha\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \alpha\mathbf{I} \end{pmatrix} \tag{2.6}$$

and

$$\mathbf{R}(\alpha) := \mathbf{R}(\mathbf{W};\alpha) = \frac{\alpha}{\alpha+1}\begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}.$$

We note that the PMHSS iteration scheme in the real version is induced from the matrix splitting

$$\mathbf{A} = \mathbf{F}(\alpha) - \mathbf{G}(\alpha),$$

where

$$\mathbf{F}(\alpha) := \mathbf{F}(\mathbf{W};\alpha) = (\alpha+1)\mathbf{P}(\alpha)\begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix} \tag{2.7}$$

and

$$\mathbf{G}(\alpha) := \mathbf{G}(\mathbf{W};\alpha) = \mathbf{P}(\alpha)\begin{pmatrix} \alpha\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \alpha\mathbf{I} \end{pmatrix}\begin{pmatrix} \alpha\mathbf{W} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{W} \end{pmatrix},$$

with $\mathbf{P}(\alpha)$ being defined in (2.4).

Moreover, it holds that

$$\rho(\mathbf{L}(\alpha)) \leqslant \frac{\sqrt{\alpha^2+1}}{\alpha+1} \qquad \forall \alpha > 0;$$

see Theorem 2.2(ii). We can further prove the convergence of this PMHSS iteration method under weaker conditions without imposing the restriction that the matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ is positive definite. This result is precisely stated in the following theorem.

THEOREM 2.3 Let $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ be the block two-by-two matrix defined as in (1.1), with $\mathbf{W} \in \mathbb{R}^{n\times n}$ and $\mathbf{T} \in \mathbb{R}^{n\times n}$ being symmetric positive semidefinite matrices satisfying $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$, and let α be a positive constant. Then the spectral radius of the PMHSS iteration matrix $\mathbf{L}(\alpha)$ satisfies $\rho(\mathbf{L}(\alpha)) \leqslant \sigma(\alpha)$, with

$$\sigma(\alpha) = \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \max_{\mu^{(\alpha)} \in \mathrm{sp}(\widetilde{\mathbf{Z}}^{(\alpha)})} \sqrt{\frac{1+|\mu^{(\alpha)}|^2}{2}},$$

where $\widetilde{\mathbf{Z}}^{(\alpha)} = (\alpha\mathbf{W}+\mathbf{T})^{-1}(\alpha\mathbf{W}-\mathbf{T})$. Therefore, it holds that

$$\rho(\mathbf{L}(\alpha)) \leqslant \sigma(\alpha) \leqslant \frac{\sqrt{\alpha^2+1}}{\alpha+1} \qquad \forall \alpha > 0;$$

i.e., the PMHSS iteration converges unconditionally to the unique solution of the block two-by-two linear system (1.1) for any initial guess.

Proof. Because $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite matrices, $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$, and $\alpha > 0$, we know that the matrix $\alpha\mathbf{W}+\mathbf{T}$ is symmetric positive definite. Based on the identities

$$\begin{pmatrix} \alpha\mathbf{W} & \mathbf{T} \\ -\mathbf{T} & \alpha\mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & \mathbf{T}-\alpha\mathbf{W} \\ \alpha\mathbf{W}-\mathbf{T} & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} \alpha\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \alpha\mathbf{I} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} (\alpha+1)\mathbf{I} & (\alpha-1)\mathbf{I} \\ (1-\alpha)\mathbf{I} & (\alpha+1)\mathbf{I} \end{pmatrix},$$

from (2.6) we immediately have

$$\mathbf{L}(\alpha) = \frac{1}{2} \begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & 0 \\ 0 & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix}^{-1} \begin{pmatrix} \alpha\mathbf{W}+\mathbf{T} & \mathbf{T}-\alpha\mathbf{W} \\ \alpha\mathbf{W}-\mathbf{T} & \alpha\mathbf{W}+\mathbf{T} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I} \\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I} \end{pmatrix},$$

which is similar to the matrix

$$\widehat{\mathbf{L}}(\alpha) = \frac{1}{2} \begin{pmatrix} \mathbf{I} & -\widehat{\mathbf{Z}}^{(\alpha)} \\ \widehat{\mathbf{Z}}^{(\alpha)} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I} \\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I} \end{pmatrix},$$

with

$$\widehat{\mathbf{Z}}^{(\alpha)} = (\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}(\alpha\mathbf{W}-\mathbf{T})(\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}.$$

By noting

$$\begin{pmatrix} \mathbf{I} & -\widehat{\mathbf{Z}}^{(\alpha)} \\ \widehat{\mathbf{Z}}^{(\alpha)} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\widehat{\mathbf{Z}}^{(\alpha)} \\ \widehat{\mathbf{Z}}^{(\alpha)} & \mathbf{I} \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} \mathbf{I} + \widehat{\mathbf{Z}}^{(\alpha)^2} & 0 \\ 0 & \mathbf{I} + \widehat{\mathbf{Z}}^{(\alpha)^2} \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I} \\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I} \\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I} \end{pmatrix}^{\mathrm{T}} = \frac{2(\alpha^2+1)}{(\alpha+1)^2}\mathbf{I},$$

we obtain

$$\widehat{\mathbf{L}}(\alpha)\widehat{\mathbf{L}}(\alpha)^{\mathrm{T}} = \frac{\alpha^2+1}{2(\alpha+1)^2} \begin{pmatrix} \mathbf{I} + \widehat{\mathbf{Z}}^{(\alpha)^2} & 0 \\ 0 & \mathbf{I} + \widehat{\mathbf{Z}}^{(\alpha)^2} \end{pmatrix}.$$

In addition, straightforward computations show that $\widehat{\mathbf{L}}(\alpha)\widehat{\mathbf{L}}(\alpha)^{\mathrm{T}} = \widehat{\mathbf{L}}(\alpha)^{\mathrm{T}}\widehat{\mathbf{L}}(\alpha)$, so $\widehat{\mathbf{L}}(\alpha)$ is a normal matrix. Therefore, it holds that

$$\rho(\widehat{\mathbf{L}}(\alpha)) = \|\widehat{\mathbf{L}}(\alpha)\|_2 = \sqrt{\rho(\widehat{\mathbf{L}}(\alpha)\widehat{\mathbf{L}}(\alpha)^{\mathrm{T}})} = \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \sqrt{\frac{1+\rho(\widehat{\mathbf{Z}}^{(\alpha)})^2}{2}}. \tag{2.8}$$

As $\widehat{\mathbf{Z}}^{(\alpha)}$ is similar to $\widetilde{\mathbf{Z}}^{(\alpha)}$, we know that $\rho(\widehat{\mathbf{Z}}^{(\alpha)}) = \rho(\widetilde{\mathbf{Z}}^{(\alpha)})$. Hence, from (2.8) we further obtain

$$\rho(\mathbf{L}(\alpha)) = \rho(\widehat{\mathbf{L}}(\alpha)) = \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \max_{\mu^{(\alpha)} \in \mathrm{sp}(\widehat{\mathbf{Z}}^{(\alpha)})} \sqrt{\frac{1+|\mu^{(\alpha)}|^2}{2}} = \sigma(\alpha).$$

For all $\mu^{(\alpha)} \in \mathrm{sp}(\widetilde{\mathbf{Z}}^{(\alpha)})$, it easily follows from $\mathrm{sp}(\widetilde{\mathbf{Z}}^{(\alpha)}) \subseteq [-1,\ 1]$ that $\frac{1}{2}(1+|\mu^{(\alpha)}|^2) \leqslant 1$ and, therefore,

$$\sigma(\alpha) \leqslant \frac{\sqrt{\alpha^2+1}}{\alpha+1}.$$

□

The spectral properties of the preconditioned matrix $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ are established in the following theorem.

THEOREM 2.4 Let $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ be the block two-by-two matrix defined as in (1.1), with $\mathbf{W} \in \mathbb{R}^{n\times n}$ and $\mathbf{T} \in \mathbb{R}^{n\times n}$ being symmetric positive semidefinite matrices satisfying $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$, and let α be a positive constant. Define $\mathbf{Z}^{(\alpha)} = (\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}(\mathbf{W}-\alpha\mathbf{T})(\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}$. Denote by $\mu_1^{(\alpha)}, \mu_2^{(\alpha)}, \ldots, \mu_n^{(\alpha)}$ the eigenvalues of the symmetric matrix $\mathbf{Z}^{(\alpha)} \in \mathbb{R}^{n\times n}$, and by $\mathbf{q}_1^{(\alpha)}, \mathbf{q}_2^{(\alpha)}, \ldots, \mathbf{q}_n^{(\alpha)}$ the corresponding $(\alpha\mathbf{W}+\mathbf{T})^{-1}$-orthogonal eigenvectors. Then the eigenvalues of the matrix $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ are

$$\lambda_{\pm}^{(\alpha,j)} = \frac{\alpha}{\alpha^2+1}\left(1 \pm \mathrm{i}\frac{\alpha-1}{\alpha+1}\right)\left(1 \pm \mathrm{i}\mu_j^{(\alpha)}\right), \qquad j = 1, 2, \ldots, n,$$

and the corresponding eigenvectors are

$$\widetilde{\mathbf{x}}_j^{(\alpha)} = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{x}_j^{(\alpha)} \\ \mathrm{i}\,\mathbf{x}_j^{(\alpha)} \end{pmatrix}, \quad \widetilde{\mathbf{x}}_{n+j}^{(\alpha)} = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{x}_j^{(\alpha)} \\ -\mathrm{i}\,\mathbf{x}_j^{(\alpha)} \end{pmatrix}, \qquad j = 1, 2, \ldots, n,$$

with $\mathbf{x}_j^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{q}_j^{(\alpha)}$. Therefore, we have $\mathbf{F}(\alpha)^{-1}\mathbf{A} = \widetilde{\mathbf{X}}^{(\alpha)}\Lambda^{(\alpha)}\widetilde{\mathbf{X}}^{(\alpha)^{-1}}$, where

$$\widetilde{\mathbf{X}}^{(\alpha)} = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{X}^{(\alpha)} & \mathbf{X}^{(\alpha)} \\ \mathrm{i}\,\mathbf{X}^{(\alpha)} & -\mathrm{i}\,\mathbf{X}^{(\alpha)} \end{pmatrix}, \qquad \mathbf{X}^{(\alpha)} = \left(\mathbf{x}_1^{(\alpha)}, \mathbf{x}_2^{(\alpha)}, \ldots, \mathbf{x}_n^{(\alpha)}\right) \in \mathbb{R}^{n\times n},$$

and

$$\Lambda^{(\alpha)} = \begin{pmatrix} \Lambda_+^{(\alpha)} & 0 \\ 0 & \Lambda_-^{(\alpha)} \end{pmatrix}, \qquad \Lambda_\pm^{(\alpha)} = \mathrm{diag}\left(\lambda_\pm^{(\alpha,1)}, \lambda_\pm^{(\alpha,2)}, \ldots, \lambda_\pm^{(\alpha,n)}\right) \in \mathbb{C}^{n\times n}.$$

Moreover, $\mathbf{X}^{(\alpha)} \in \mathbb{R}^{n\times n}$ is orthogonal and $\widetilde{\mathbf{X}}^{(\alpha)} \in \mathbb{C}^{2n\times 2n}$ is unitary, so it holds that

$$\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = \kappa_2(\mathbf{X}^{(\alpha)}) = 1,$$

where $\kappa_2(\cdot)$ represents the condition number in the Euclidean norm.

Proof. Define matrices

$$\mathbf{Q}^{(\alpha)} = \left(\mathbf{q}_1^{(\alpha)}, \mathbf{q}_2^{(\alpha)}, \ldots, \mathbf{q}_n^{(\alpha)}\right) \in \mathbb{R}^{n\times n}$$

and

$$\Xi^{(\alpha)} = \mathrm{diag}\left(\mu_1^{(\alpha)}, \mu_2^{(\alpha)}, \ldots, \mu_n^{(\alpha)}\right) \in \mathbb{R}^{n\times n}.$$

Then it holds that

$$\mathbf{Z}^{(\alpha)}\mathbf{Q}^{(\alpha)} = \mathbf{Q}^{(\alpha)}\Xi^{(\alpha)} \quad \text{and} \quad \mathbf{X}^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{Q}^{(\alpha)}.$$

Because the inverse of the matrix $\mathbf{P}(\alpha)$ defined in (2.4) is given by

$$\mathbf{P}(\alpha)^{-1} = \alpha\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}$$

and the matrices

$$\begin{pmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}$$

commute, from (1.1) and (2.7) we easily get

$$\mathbf{F}(\alpha)^{-1}\mathbf{A} = \frac{\alpha}{\alpha+1}\begin{pmatrix} \alpha\mathbf{W} + \mathbf{T} & 0 \\ 0 & \alpha\mathbf{W} + \mathbf{T} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}. \tag{2.9}$$

Based on the identities

$$\begin{pmatrix} \mathbf{W} & -\mathbf{T} \\ \mathbf{T} & \mathbf{W} \end{pmatrix}\begin{pmatrix} \alpha\mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \alpha\mathbf{I} \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{W} + \mathbf{T} & \mathbf{W} - \alpha\mathbf{T} \\ \alpha\mathbf{T} - \mathbf{W} & \alpha\mathbf{W} + \mathbf{T} \end{pmatrix}$$

and

$$\begin{pmatrix}\alpha\mathbf{I} & \mathbf{I}\\ -\mathbf{I} & \alpha\mathbf{I}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ -\mathbf{I} & \mathbf{I}\end{pmatrix} = \frac{1}{\alpha^2+1}\begin{pmatrix}(\alpha+1)\mathbf{I} & (\alpha-1)\mathbf{I}\\ (1-\alpha)\mathbf{I} & (\alpha+1)\mathbf{I}\end{pmatrix},$$

we further obtain from (2.9) that

$$\begin{aligned}\mathbf{F}(\alpha)^{-1}\mathbf{A} &= \frac{\alpha}{\alpha^2+1}\begin{pmatrix}\alpha\mathbf{W}+\mathbf{T} & 0\\ 0 & \alpha\mathbf{W}+\mathbf{T}\end{pmatrix}^{-1}\begin{pmatrix}\alpha\mathbf{W}+\mathbf{T} & \mathbf{W}-\alpha\mathbf{T}\\ \alpha\mathbf{T}-\mathbf{W} & \alpha\mathbf{W}+\mathbf{T}\end{pmatrix}\begin{pmatrix}\mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I}\\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I}\end{pmatrix}\\
&= \frac{\alpha}{\alpha^2+1}\begin{pmatrix}\alpha\mathbf{W}+\mathbf{T} & 0\\ 0 & \alpha\mathbf{W}+\mathbf{T}\end{pmatrix}^{-\frac{1}{2}}\begin{pmatrix}\mathbf{I} & \mathbf{Z}^{(\alpha)}\\ -\mathbf{Z}^{(\alpha)} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\alpha\mathbf{W}+\mathbf{T} & 0\\ 0 & \alpha\mathbf{W}+\mathbf{T}\end{pmatrix}^{\frac{1}{2}}\widehat{\mathbf{J}}^{(\alpha)}\\
&= \frac{\alpha}{\alpha^2+1}\begin{pmatrix}\mathbf{X}^{(\alpha)} & 0\\ 0 & \mathbf{X}^{(\alpha)}\end{pmatrix}\begin{pmatrix}\mathbf{I} & \varXi^{(\alpha)}\\ -\varXi^{(\alpha)} & \mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{X}^{(\alpha)} & 0\\ 0 & \mathbf{X}^{(\alpha)}\end{pmatrix}^{-1}\widehat{\mathbf{J}}^{(\alpha)}\\
&= \frac{\alpha}{\alpha^2+1}\widehat{\mathbf{X}}^{(\alpha)}\widehat{\varPsi}^{(\alpha)}\widehat{\mathbf{X}}^{(\alpha)^{-1}}\widehat{\mathbf{J}}^{(\alpha)}\\
&= \frac{\alpha}{\alpha^2+1}\widehat{\mathbf{X}}^{(\alpha)}\widehat{\varPsi}^{(\alpha)}\widehat{\mathbf{J}}^{(\alpha)}\widehat{\mathbf{X}}^{(\alpha)^{-1}}, \qquad (2.10)\end{aligned}$$

where

$$\widehat{\mathbf{X}}^{(\alpha)} = \begin{pmatrix}\mathbf{X}^{(\alpha)} & 0\\ 0 & \mathbf{X}^{(\alpha)}\end{pmatrix}, \quad \widehat{\varPsi}^{(\alpha)} = \begin{pmatrix}\mathbf{I} & \varXi^{(\alpha)}\\ -\varXi^{(\alpha)} & \mathbf{I}\end{pmatrix} \quad \text{and} \quad \widehat{\mathbf{J}}^{(\alpha)} = \begin{pmatrix}\mathbf{I} & \frac{\alpha-1}{\alpha+1}\mathbf{I}\\ \frac{1-\alpha}{\alpha+1}\mathbf{I} & \mathbf{I}\end{pmatrix}.$$

Let

$$\mathbf{U}^{(\alpha)} = \frac{1}{\sqrt{2}}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ \mathrm{i}\,\mathbf{I} & -\mathrm{i}\,\mathbf{I}\end{pmatrix}.$$

Then by straightforward computation, we have

$$\mathbf{U}^{(\alpha)^*}\widehat{\varPsi}^{(\alpha)}\mathbf{U}^{(\alpha)} = \begin{pmatrix}\mathbf{I}+\mathrm{i}\,\varXi^{(\alpha)} & 0\\ 0 & \mathbf{I}-\mathrm{i}\,\varXi^{(\alpha)}\end{pmatrix}$$

and

$$\mathbf{U}^{(\alpha)^*}\widehat{\mathbf{J}}^{(\alpha)}\mathbf{U}^{(\alpha)} = \begin{pmatrix}\frac{(\alpha+1)+(\alpha-1)\mathrm{i}}{\alpha+1}\mathbf{I} & 0\\ 0 & \frac{(\alpha+1)-(\alpha-1)\mathrm{i}}{\alpha+1}\mathbf{I}\end{pmatrix}.$$

Now it follows from (2.10) that

$$\begin{aligned}\mathbf{F}(\alpha)^{-1}\mathbf{A} &= \frac{\alpha}{(\alpha+1)(\alpha^2+1)}\widehat{\mathbf{X}}^{(\alpha)}\mathbf{U}^{(\alpha)}\begin{pmatrix}\mathbf{I}+\mathrm{i}\,\varXi^{(\alpha)} & 0\\ 0 & \mathbf{I}-\mathrm{i}\,\varXi^{(\alpha)}\end{pmatrix}\\
&\quad\begin{pmatrix}[(\alpha+1)+\mathrm{i}(\alpha-1)]\mathbf{I} & 0\\ 0 & [(\alpha+1)-\mathrm{i}(\alpha-1)]\mathbf{I}\end{pmatrix}\mathbf{U}^{(\alpha)^*}\widehat{\mathbf{X}}^{(\alpha)^{-1}}\\
&= \widetilde{\mathbf{X}}^{(\alpha)}\varLambda^{(\alpha)}\widetilde{\mathbf{X}}^{(\alpha)^{-1}},\end{aligned}$$

where $\widetilde{\mathbf{X}}^{(\alpha)} = \widehat{\mathbf{X}}^{(\alpha)}\mathbf{U}^{(\alpha)}$ and

$$\varLambda_{+}^{(\alpha)} = \frac{\alpha[(\alpha+1)+\mathrm{i}(\alpha-1)]}{(\alpha+1)(\alpha^2+1)}(\mathbf{I}+\mathrm{i}\,\varXi^{(\alpha)}), \quad \varLambda_{-}^{(\alpha)} = \frac{\alpha[(\alpha+1)-\mathrm{i}(\alpha-1)]}{(\alpha+1)(\alpha^2+1)}(\mathbf{I}-\mathrm{i}\,\varXi^{(\alpha)}).$$

Moreover, as $\mathbf{Q}^{(\alpha)} \in \mathbb{R}^{n\times n}$ is $(\alpha\mathbf{W}+\mathbf{T})^{-1}$-orthogonal and $\mathbf{U}^{(\alpha)} \in \mathbb{C}^{2n\times 2n}$ is unitary, we immediately find that $\mathbf{X}^{(\alpha)} \in \mathbb{R}^{n\times n}$ and $\widehat{\mathbf{X}}^{(\alpha)} \in \mathbb{R}^{2n\times 2n}$ are orthogonal and, hence, $\widetilde{\mathbf{X}}^{(\alpha)} \in \mathbb{C}^{2n\times 2n}$ is unitary. It then follows that $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = \kappa_2(\mathbf{X}^{(\alpha)}) = 1$. □

REMARK 2.5 If $\alpha = 1$, then Theorem 2.3 leads to $\sigma(1) \leqslant \frac{\sqrt{2}}{2}$. This shows that when

$$\mathbf{F} := \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{W}+\mathbf{T} & 0 \\ 0 & \mathbf{W}+\mathbf{T} \end{pmatrix}$$

is used to precondition the matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$, the eigenvalues of the preconditioned matrix $\mathbf{F}^{-1}\mathbf{A}$ are contained within the complex disk centred at 1 with radius $\frac{\sqrt{2}}{2}$. Moreover, Theorem 2.4 indicates that the matrix $\mathbf{F}^{-1}\mathbf{A}$ is diagonalizable, with the matrix $\widetilde{\mathbf{X}}^{(1)}$, formed by its eigenvectors, being unitary. Hence, the preconditioned Krylov subspace iteration methods, when employed to solve the block two-by-two linear system (1.1), can be expected to converge rapidly and independently of both problem data and mesh size.

REMARK 2.6 For the PMHSS preconditioning matrix $\mathbf{F}_c(\alpha)$ defined in (2.7) in Bai *et al.* (2011) for complex symmetric linear systems of the form $\mathbf{A}_c\mathbf{x}_c = \mathbf{b}_c$, with $\mathbf{A}_c = \mathbf{W} + \mathrm{i}\mathbf{T} \in \mathbb{C}^{n\times n}$ and $\mathbf{b}_c \in \mathbb{C}^n$, in an analogous fashion to the proof of Theorem 2.4 we can also demonstrate that $\mathbf{F}_c(\alpha)^{-1}\mathbf{A}_c$ is diagonalizable by the orthogonal matrix $\mathbf{X}^{(\alpha)} \in \mathbb{R}^{n\times n}$, where

$$\mathbf{X}^{(\alpha)} = \left(\mathbf{x}_1^{(\alpha)}, \mathbf{x}_2^{(\alpha)}, \ldots, \mathbf{x}_n^{(\alpha)}\right), \quad \text{with} \quad \mathbf{x}_j^{(\alpha)} = (\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}\mathbf{q}_j^{(\alpha)},$$

and $\mathbf{q}_1^{(\alpha)}, \mathbf{q}_2^{(\alpha)}, \ldots, \mathbf{q}_n^{(\alpha)}$ are the $(\alpha\mathbf{W}+\mathbf{T})^{-1}$-orthogonal eigenvectors of the matrix $\mathbf{Z}^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}(\mathbf{W} - \alpha\mathbf{T})(\alpha\mathbf{W}+\mathbf{T})^{-\frac{1}{2}}$. Hence, the preconditioned Krylov subspace iteration methods such as GMRES, when employed to solve the complex symmetric linear system $\mathbf{A}_c\mathbf{x}_c = \mathbf{b}_c$, can be expected to converge very rapidly. This result improves Theorems 3.2 and 3.3 in Bai *et al.* (2011).

3. Applications to the numerical solution of distributed control problems

Consider the distributed control problem

$$\min_{u,f} \frac{1}{2}\|u-u_*\|_2^2 + \beta\|f\|_2^2 \tag{3.1}$$

$$\text{subject to} \quad -\nabla^2 u = f \quad \text{in} \quad \varOmega, \tag{3.2}$$

$$\text{with} \quad u = g \quad \text{on} \quad \partial\varOmega_1 \quad \text{and} \quad \frac{\partial u}{\partial \mathrm{n}} = g \quad \text{on} \quad \partial\varOmega_2, \tag{3.3}$$

where $\varOmega$ is a domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, $\partial\varOmega$ is the boundary of $\varOmega$, and $\partial\varOmega_1$ and $\partial\varOmega_2$ are two parts of $\partial\varOmega$ satisfying $\partial\varOmega_1 \cup \partial\varOmega_2 = \partial\varOmega$ and $\partial\varOmega_1 \cap \partial\varOmega_2 = \emptyset$. The parameter $\beta > 0$ in (3.1) is a regularization parameter; some authors use $\frac{\beta}{2}$ instead of β, e.g., Pearson & Wathen (2010). Such problems, introduced by Lions (1968), consist of a cost functional (3.1) to be minimized subject to a PDE problem (3.2–3.3) posed on the domain $\varOmega$. Here the function u_* (the 'desired state') is known, and we want to find u, which satisfies the PDE problem and is as close to u_* as possible in the L_2-norm sense. For recent references on this topic, see, e.g., Ito & Kunisch (2008), Lass *et al.* (2009), Bai (2011), Takacs & Zulehner (2011) and Tröltzsch (2010).

When the PDE-constrained optimization problem (3.1–3.3) is treated with the discretize-then-optimize approach (Rees *et al.*, 2010) through application of a Galerkin finite element method to its weak formulation, we obtain a KKT system in the following saddle point form:

$$\begin{pmatrix} 2\beta M & 0 & -M \\ 0 & M & K^{\mathrm{T}} \\ -M & K & 0 \end{pmatrix} \begin{pmatrix} f \\ u \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ b \\ d \end{pmatrix}, \tag{3.4}$$

where $M \in \mathbb{R}^{m\times m}$ is the mass matrix, $K \in \mathbb{R}^{m\times m}$ is the stiffness matrix (the discrete Laplacian), $\beta > 0$ is the regularization parameter, $d \in \mathbb{R}^m$ contains the terms coming from the boundary values of the discrete solution and $b \in \mathbb{R}^m$ is the Galerkin projection of the discrete state u_*. In addition, λ is a vector of Lagrange multipliers.

The saddle point linear system (3.4) can be equivalently rewritten in block element form as

$$\begin{cases} 2\beta Mf - M\lambda = 0, \\ Mu + K^{\mathrm{T}}\lambda \quad = b, \\ -Mf + Ku \quad = d. \end{cases} \tag{3.5}$$

From the first equation in (3.5) we easily get

$$\lambda = 2\beta f. \tag{3.6}$$

By substituting this expression with respect to λ into the other two equations in (3.5), we obtain

$$\begin{cases} Mu + 2\beta K^{\mathrm{T}} f = b, \\ Ku - Mf \qquad = d, \end{cases}$$

or equivalently,

$$Ax \equiv \begin{pmatrix} \frac{1}{2\beta}M & K^{\mathrm{T}} \\ -K & M \end{pmatrix} \begin{pmatrix} u \\ f \end{pmatrix} = \begin{pmatrix} \frac{1}{2\beta}b \\ -d \end{pmatrix} \equiv g. \tag{3.7}$$

We remark that the block two-by-two linear system (3.7), together with (3.6), is equivalent to the saddle point linear system (3.4); see also Simoncini (2011). Recall that $M \in \mathbb{R}^{m\times m}$ is the mass matrix and is, thus, symmetric positive definite. Therefore, the matrix $A \in \mathbb{R}^{2m\times 2m}$ is positive real, i.e., its symmetric part is positive definite.

In the remainder of this section, we assume that the stiffness matrix K is symmetric and positive semidefinite; this assumption is slightly more general than needed for the treatment of the particular distributed control problem (3.1–3.3), for which K is positive definite. Through symmetric block-scaling by the diagonal matrix

$$D = \begin{pmatrix} -\sqrt{2\beta}I & 0 \\ 0 & I \end{pmatrix}, \tag{3.8}$$

where $I \in \mathbb{R}^{m\times m}$ represents the identity matrix, we can reformulate the block two-by-two linear system (3.7) into the form of (1.1), with

$$\mathbf{A} = DAD = \begin{pmatrix} M & -\sqrt{2\beta}K \\ \sqrt{2\beta}K & M \end{pmatrix}, \qquad \text{i.e.,} \quad \mathbf{W} = M \quad \text{and} \quad \mathbf{T} = \sqrt{2\beta}K, \tag{3.9}$$

and

$$\begin{cases} \mathbf{x} = D^{-1}x = \begin{pmatrix} -\frac{1}{\sqrt{2\beta}}u \\ f \end{pmatrix}, & \text{i.e., } \mathbf{y} = -\frac{1}{\sqrt{2\beta}}u \quad \text{and} \quad \mathbf{z} = f, \\ \mathbf{g} = Dg \quad = \begin{pmatrix} -\frac{1}{\sqrt{2\beta}}b \\ -d \end{pmatrix}, & \text{i.e., } \mathbf{p} = -\frac{1}{\sqrt{2\beta}}b \quad \text{and} \quad \mathbf{q} = -d. \end{cases} \tag{3.10}$$

Now first applying Method 2.1 directly to the block two-by-two linear system (1.1), and then transforming the induced iteration scheme back to the original variables through (3.9) and (3.10), with the notational replacement $\mathbf{V} = V$, we obtain the PMHSS iteration method for solving the block two-by-two linear system (3.7) as follows.

METHOD 3.1. (The PMHSS iteration method for (3.7)).
Let $(u^{(0)^{\mathrm{T}}}, f^{(0)^{\mathrm{T}}})^{\mathrm{T}} \in \mathbb{R}^{2m}$ be an arbitrary initial guess, with $u^{(0)}, f^{(0)} \in \mathbb{R}^m$. For $k = 0, 1, 2, \ldots,$ until the sequence of iterates $\{(u^{(k)^{\mathrm{T}}}, f^{(k)^{\mathrm{T}}})^{\mathrm{T}}\}_{k=0}^{\infty} \subset \mathbb{R}^{2m}$ converges, compute the next iterate $(u^{(k+1)^{\mathrm{T}}},$ $f^{(k+1)^{\mathrm{T}}})^{\mathrm{T}}$ according to the following procedure:

$$\begin{cases} \begin{pmatrix} \alpha V + M & 0 \\ 0 & \alpha V + M \end{pmatrix} \begin{pmatrix} u^{(k+\frac{1}{2})} \\ f^{(k+\frac{1}{2})} \end{pmatrix} = \begin{pmatrix} \alpha V & -2\beta K \\ K & \alpha V \end{pmatrix} \begin{pmatrix} u^{(k)} \\ f^{(k)} \end{pmatrix} + \begin{pmatrix} b \\ -d \end{pmatrix}, \\ \begin{pmatrix} \alpha V + \sqrt{2\beta}K & 0 \\ 0 & \alpha V + \sqrt{2\beta}K \end{pmatrix} \begin{pmatrix} u^{(k+1)} \\ f^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha V & \sqrt{2\beta}M \\ -\frac{1}{\sqrt{2\beta}}M & \alpha V \end{pmatrix} \begin{pmatrix} u^{(k+\frac{1}{2})} \\ f^{(k+\frac{1}{2})} \end{pmatrix} + \begin{pmatrix} \sqrt{2\beta}d \\ \frac{1}{\sqrt{2\beta}}b \end{pmatrix}, \end{cases}$$

where α is a given positive constant and $V \in \mathbb{R}^{m \times m}$ is a prescribed symmetric positive definite matrix.

We easily see that the convergence rate of the PMHSS iteration sequence generated by Method 3.1 is bounded by

$$\sigma_o(\alpha) \equiv \max_{\widetilde{\lambda}_j \in \mathrm{sp}(V^{-1}M)} \frac{\sqrt{\alpha^2 + \widetilde{\lambda}_j^2}}{\alpha + \widetilde{\lambda}_j} \cdot \max_{\widetilde{\mu}_j \in \mathrm{sp}(V^{-1}K)} \frac{\sqrt{\alpha^2 + 2\beta\widetilde{\mu}_j^2}}{\alpha + \sqrt{2\beta}\widetilde{\mu}_j}.$$

Moreover, the PMHSS preconditioner induced from Method 3.1 for the block two-by-two linear system (3.7) is given by

$$\begin{aligned} F(V; \alpha) &= D^{-1}\mathbf{F}(\mathbf{V}; \alpha)D^{-1} \\ &= P(\alpha; \beta) \begin{pmatrix} \alpha V + M & 0 \\ 0 & \alpha V + M \end{pmatrix} \begin{pmatrix} V^{-1} & 0 \\ 0 & V^{-1} \end{pmatrix} \begin{pmatrix} \alpha V + \sqrt{2\beta}K & 0 \\ 0 & \alpha V + \sqrt{2\beta}K \end{pmatrix}, \end{aligned} \tag{3.11}$$

where

$$P(\alpha; \beta) = D^{-1}\mathbf{P}(\alpha)D^{-1} = \frac{1}{4\alpha\beta} \begin{pmatrix} I & \sqrt{2\beta}I \\ -\sqrt{2\beta}I & 2\beta I \end{pmatrix}; \tag{3.12}$$

see (2.3), (2.4) and (3.8) for the definitions of the matrices $\mathbf{F}(\mathbf{V}; \alpha)$, $\mathbf{P}(\alpha)$ and D, respectively. Here we have applied the relationships in (3.9).

In actual implementations, the action of the preconditioning matrix $F(V;\alpha)$, when used to precondition the Krylov subspace iteration methods, is often realized through solving a sequence of generalized residual equations of the form

$$F(V;\alpha)v = r,$$

where $r = (r_a^{\mathrm{T}}, r_b^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{2m}$, with $r_a, r_b \in \mathbb{R}^m$, represents the current residual vector, while $v = (v_a^{\mathrm{T}}, v_b^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{2m}$, with $v_a, v_b \in \mathbb{R}^m$, represents the generalized residual vector. By making use of the concrete structure of the matrix $F(V;\alpha)$, we obtain the following procedure for computing the vector v:

(i) compute $\widetilde{v}_a, \widetilde{v}_b \in \mathbb{R}^m$ by solving the systems of linear equations

$$\begin{cases} (\alpha V + M)\widetilde{v}_a = 2\alpha\beta\left(r_a - \frac{1}{\sqrt{2\beta}} r_b\right), \\ (\alpha V + M)\widetilde{v}_b = \alpha(\sqrt{2\beta} r_a + r_b); \end{cases}$$

(ii) compute $\widehat{v}_a, \widehat{v}_b \in \mathbb{R}^m$ by

$$\widehat{v}_a = V\widetilde{v}_a \quad \text{and} \quad \widehat{v}_b = V\widetilde{v}_b;$$

(iii) compute $v_a, v_b \in \mathbb{R}^m$ by solving the systems of linear equations

$$\begin{cases} (\alpha V + \sqrt{2\beta} K)v_a = \widehat{v}_a, \\ (\alpha V + \sqrt{2\beta} K)v_b = \widehat{v}_b. \end{cases}$$

Note that both matrices $\alpha V + M$ and $\alpha V + \sqrt{2\beta} K$ are symmetric positive definite. Hence, the above four systems of linear equations can be solved effectively either exactly by Cholesky factorizations or inexactly by some conjugate gradient or multigrid scheme; see Bai *et al.* (2003, 2008).

In particular, when $V = M$, the PMHSS preconditioning matrix $F(V;\alpha)$ defined in (3.11–3.12) reduces to

$$F(\alpha) := F(M;\alpha) = (\alpha+1)P(\alpha;\beta)\begin{pmatrix} \alpha M + \sqrt{2\beta} K & 0 \\ 0 & \alpha M + \sqrt{2\beta} K \end{pmatrix}. \tag{3.13}$$

If $\alpha = 1$, then we further have

$$F := P\begin{pmatrix} M + \sqrt{2\beta} K & 0 \\ 0 & M + \sqrt{2\beta} K \end{pmatrix}, \tag{3.14}$$

where

$$P = \frac{1}{2\beta}\begin{pmatrix} I & \sqrt{2\beta} I \\ -\sqrt{2\beta} I & 2\beta I \end{pmatrix}.$$

We stress that our solution approach is not limited to the special (and rather simple) model problem (3.1–3.3), and that it can handle any kind of distributed control problem leading to KKT systems of the form (3.4) with K symmetric and positive (semi-)definite. Hence, a broad class of elliptic PDE constraints can be accommodated besides Poisson's equation.

Finally, the definition in (3.14) shows that up to the factor P, the preconditioner F is nothing but a block diagonal preconditioner. Recently, different block diagonal preconditioners for optimal control of Poisson's equation have been developed by several authors; see, e.g., Rees *et al.* (2010), Pearson & Wathen (2010) and Zulehner (2011). In Rees *et al.* (2010), it was found that various block diagonal preconditioners for MINRES are robust with respect to h, but their performance deteriorates for decreasing values of β. Furthermore, the results in Rees *et al.* (2010) demonstrate that the *preconditioned projected conjugate gradient* (PPCG) method with constraint preconditioning is systematically faster than MINRES with block diagonal preconditioning in terms of solution times, often by a large margin. Other more robust preconditioners have been recently introduced in Zulehner (2011) and Pearson & Wathen (2010). It is interesting to note that the diagonal blocks in these preconditioners are scalar combinations of the mass matrix M and the stiffness matrix K, similar to the coefficient matrices in (3.4) and (3.7), respectively. These preconditioners appear to be robust with respect to both h and β and have costs comparable to the PMHSS preconditioner developed here. For instance, the block diagonal preconditioner in Pearson & Wathen (2010) requires at each iteration the (approximate) solution of two linear systems with coefficient matrix $K + \frac{1}{\sqrt{2\beta}}M$, in addition to three linear systems involving the mass matrix. (Recall that our β corresponds to $\beta/2$ in Pearson & Wathen, 2010; hence, our $K + \frac{1}{\sqrt{2\beta}}M$ corresponds to the matrix $K + \frac{1}{\sqrt{\beta}}M$ in Pearson & Wathen, 2010.) The PMHSS preconditioner (with $\alpha = 1$) requires the (approximate) solution of two linear systems with coefficient matrix $M + \sqrt{2\beta}K$ at each iteration. Hence, apart from the (inexpensive) mass matrix solves, the two preconditioners require almost identical computational effort.

4. Numerical results

In this section, we use the following example to examine the numerical behaviour of the PMHSS preconditioning matrix and the corresponding preconditioned Krylov subspace iteration methods.

EXAMPLE 4.1 (Rees *et al.*, 2010). Let $\Omega = [0, 1]^2$ be a unit square and consider the distributed control problem (3.1–3.3) with $\partial\Omega_2 = \emptyset$, $g = u_*$ and

$$u_* = \begin{cases} (2x-1)^2(2y-1)^2 & \text{if } (x, y) \in [0, \frac{1}{2}]^2, \\ 0 & \text{otherwise.} \end{cases}$$

Our aim is to compare the PMHSS scheme with some of the best existing solvers for this problem. To this end, we solve the system of linear equations (3.4) by the projected conjugate gradient method preconditioned with the constraint preconditioning matrix P_C and its approximation (inexact variant) $P_C^{(\mathrm{app})}$ (see, e.g., Keller *et al.*, 2000) and solve the system of linear equations (3.7) by the GMRES method preconditioned with the PMHSS preconditioning matrices $F(\alpha)$ and F, defined by (3.13) and (3.14), and their inexact variants $F^{(\mathrm{app})}(\alpha)$ and $F^{(\mathrm{app})}$. Also, we solve the system of linear equations (3.4) and its block two-by-two variant

$$A_s x \equiv \begin{pmatrix} \frac{1}{2\beta}M & K^{\mathrm{T}} \\ K & -M \end{pmatrix} \begin{pmatrix} u \\ f \end{pmatrix} = \begin{pmatrix} \frac{1}{2\beta}b \\ d \end{pmatrix} \equiv g_s \qquad (4.1)$$

by the MINRES method preconditioned with the block diagonal preconditioning matrices

$$P_D = \mathrm{Diag}\left(2\beta M,\ M,\ \frac{1}{2\beta}M + KM^{-1}K\right) \quad \text{and} \quad P_{D2} = \mathrm{Diag}\left(\frac{1}{2\beta}M,\ M + 2\beta KM^{-1}K\right),$$

respectively. We also consider the corresponding inexact variants of the block diagonal preconditioners

$$P_D^{(\mathrm{app})} = \mathrm{Diag}\left(2\beta M^{(\mathrm{app})},\ M^{(\mathrm{app})},\ K^{(\mathrm{app})}M^{-1}K^{(\mathrm{app})}\right)$$

and

$$P_{D2}^{(\mathrm{app})} = \mathrm{Diag}\left(\frac{1}{2\beta}M^{(\mathrm{app})},\ 2\beta K^{(\mathrm{app})}M^{-1}K^{(\mathrm{app})}\right);$$

see Keller *et al.* (2000), Rees *et al.* (2010) and Bai (2011). Here, in applying the preconditioners $P_C^{(\mathrm{app})}$, $P_D^{(\mathrm{app})}$ and $P_{D2}^{(\mathrm{app})}$, the approximation $K^{(\mathrm{app})}$ to the matrix K is implicitly defined by two *algebraic multigrid* (**AMG**) V-cycles with two damped Jacobi pre- and post-smoothing steps obtained by the `amg` operator in the software COMSOL Multiphysics.[2] The approximation $M^{(\mathrm{app})}$ to the matrix M is implicitly defined by 20 steps of Chebyshev semiiteration approximation; see Rees *et al.* (2010). In computing the actions of the inverses of $F^{(\mathrm{app})}(\alpha)$ and $F^{(\mathrm{app})}$, the inverses of the approximations $G^{(\mathrm{app})}(\alpha)$ and $G^{(\mathrm{app})}$ corresponding to the matrices $G(\alpha) := \alpha M + \sqrt{2\beta}K$ and $G := M + \sqrt{2\beta}K$ are also implemented by 20 steps of Chebyshev semiiteration. This choice is justified later in this section.

In our implementations, all iteration processes are terminated once the Euclidean norms of the current residuals are reduced by a factor of 10^4 from those of the initial residuals, and the iteration parameters α_{opt} adopted in the PMHSS preconditioners $F(\alpha)$ and $F^{(\mathrm{app})}(\alpha)$ are the experimentally found optimal ones that minimize the total iteration steps of the corresponding iteration processes; see Table 1.

In Tables 2 and 3, we list the numbers of iteration steps and the computing times (in parentheses) with respect to the constraint preconditioner, the PMHSS preconditioners and their inexact variants, which are employed to precondition the projected CG and the GMRES methods, respectively. The optimal iteration parameters α_{opt} used in these two tables are those given in Table 1.

From Table 2 we see that the number of iteration steps of the PPCG method, with the constraint preconditioner, is independent of the discretization mesh size h, with $h = \frac{1}{\sqrt{m}+1}$, when $\beta = 10^{-2}$ and 10^{-4}. It is mildly dependent on the discretization mesh size h when $\beta = 10^{-6}$ and is strongly dependent on h when $\beta = 10^{-8}$. The number of iteration steps of the PMHSS-preconditioned GMRES method, with the preconditioner $F(\alpha_{\mathrm{opt}})$ or F, is roughly independent of the discretization mesh size h for all tested values of β. In addition, as β decreases from 10^{-2} to 10^{-8}, the number of iteration steps of the PPCG method changes drastically, while that of the PMHSS-preconditioned GMRES method remains nearly constant. These observations indicate that the PMHSS preconditioner shows h- and β-independent convergence properties, whereas the constraint preconditioner does not.

For β ranging from 10^{-2} to 10^{-6}, the computing times of the PMHSS-preconditioned GMRES method are comparable with those of the PPCG method when $h = 2^{-j}$, $j = 2, 3, 4, 5$; they are, however, one order of magnitude less than those of the PPCG method when $h = 2^{-6}$. For $\beta = 10^{-8}$, the computing times of the PMHSS-preconditioned GMRES method are comparable with those of

[2] For more details about the COMSOL Multiphysics simulation software, we refer to the website http://www.comsol.com/.

TABLE 1 *Experimental optimal parameters for $F(\alpha)$- and $F^{(\mathrm{app})}(\alpha)$-preconditioned GMRES methods*

β	h	α_{opt} for $F(\alpha)$	α_{opt} for $F^{(\mathrm{app})}(\alpha)$
10^{-2}	2^{-2}	[0.01, 1.00]	[0.01, 1.00]
	2^{-3}	[0.15, 0.18]	[0.14, 0.16]
	2^{-4}	[0.15, 0.18]	[0.27, 0.30]
	2^{-5}	[0.15, 0.18]	[0.01, 0.53]
	2^{-6}	[0.15, 0.19]	[0.01, 1.00]
10^{-4}	2^{-2}	[0.83, 1.00]	[0.83, 1.00]
	2^{-3}	[0.44, 0.83]	[0.44, 0.83]
	2^{-4}	[0.21, 0.85]	[0.16, 0.95]
	2^{-5}	[0.54, 0.84]	[0.37, 0.40]
	2^{-6}	[0.55, 0.55]	[0.01, 0.10]
10^{-6}	2^{-2}	[0.66, 0.85]	[0.66, 0.85]
	2^{-3}	[0.85, 0.86]	[0.85, 0.86]
	2^{-4}	[0.98, 1.00]	[0.98, 1.00]
	2^{-5}	[0.94, 0.98]	[0.94, 0.98]
	2^{-6}	[0.94, 1.00]	[0.76, 0.76]
10^{-8}	2^{-2}	[0.92, 1.00]	[0.96, 1.00]
	2^{-3}	[0.58, 0.85]	[0.58, 0.85]
	2^{-4}	[0.85, 0.86]	[0.85, 0.86]
	2^{-5}	[0.90, 1.00]	[0.90, 1.00]
	2^{-6}	[0.96, 1.00]	[0.96, 1.00]

the PPCG method when $h = 2^{-j}$, $j = 2, 3, 4$; they are, however, one order of magnitude less than those of the PPCG method when $h = 2^{-5}$ and 2^{-6}. This shows that the performance of the PMHSS-preconditioned GMRES method is comparable to or better than that of the PPCG method.

Moreover, and importantly, the iteration steps and the computing times with respect to both preconditioners $F(\alpha_{\mathrm{opt}})$ and F are almost the same. This implies that in actual implementations of the PMHSS preconditioning matrix, one should simply take the iteration parameter α to be 1, resulting in a parameter-free method; see Figs 1 and 2.

Comparing the results in Tables 2 and 3, we observe that the inexact PMHSS preconditioner $F^{(\mathrm{app})}(\alpha_{\mathrm{opt}})$ yields almost the same iteration steps and computing times as the exact PMHSS preconditioner $F(\alpha_{\mathrm{opt}})$ when $\beta = 10^{-6}$ and 10^{-8}. The same observation applies to the inexact and the exact constraint preconditioners $P_C^{(\mathrm{app})}$ and P_C when $\beta = 10^{-2}$ and 10^{-4}. On the other hand, the inexact constraint preconditioner suffers a significant deterioration as h is decreased when $\beta = 10^{-6}$ and 10^{-8}, and the inexact PMHSS preconditioner shows some deterioration as h is decreased when $\beta = 10^{-2}$ and 10^{-4}. The reason for the latter phenomenon is that for small values of β, the coefficient matrix $G(\alpha) = \alpha M + \sqrt{2\beta}K$, with $\alpha = \alpha_{\mathrm{opt}}$ or $\alpha = 1$, is dominated by the term involving the mass matrix M, hence, it is well conditioned (independently of h) and 20 steps of Chebyshev semiiteration suffice to approximate the action of its inverse. For larger values of β, however, the stiffness matrix K makes a significant contribution and the conditioning of $G(\alpha)$, with $\alpha = \alpha_{\mathrm{opt}}$ or $\alpha = 1$, deteriorates as $h \to 0$. Hence, a fixed number of Chebyshev semiiteration is no longer sufficient to approximate the action of

TABLE 2 *Iteration counts and CPU timings (in seconds) for the constraint preconditioner and for the PMHSS preconditioners with optimal α and with $\alpha = 1$*

β	h	P_C	$F(\alpha_{\text{opt}})$	F
10^{-2}	2^{-2}	3 (3.82×10^{-2})	9 (6.00×10^{-2})	9 (1.93×10^{-2})
	2^{-3}	3 (3.30×10^{-2})	9 (2.62×10^{-2})	11 (2.17×10^{-2})
	2^{-4}	3 (3.88×10^{-2})	9 (4.56×10^{-2})	11 (4.11×10^{-2})
	2^{-5}	3 (1.49×10^{-1})	9 (1.13×10^{-1})	11 (1.15×10^{-1})
	2^{-6}	3 (1.60)	9 (4.34×10^{-1})	11 (4.81×10^{-1})
10^{-4}	2^{-2}	5 (1.22×10^{-2})	9 (2.29×10^{-2})	9 (1.83×10^{-2})
	2^{-3}	5 (1.69×10^{-2})	12 (2.68×10^{-2})	13 (2.24×10^{-2})
	2^{-4}	5 (4.00×10^{-2})	14 (5.17×10^{-2})	16 (5.08×10^{-2})
	2^{-5}	5 (1.78×10^{-1})	14 (1.35×10^{-1})	16 (1.41×10^{-1})
	2^{-6}	5 (1.70)	14 (6.07×10^{-1})	18 (6.96×10^{-1})
10^{-6}	2^{-2}	6 (1.23×10^{-2})	11 (2.43×10^{-2})	12 (1.95×10^{-2})
	2^{-3}	11 (2.05×10^{-2})	13 (2.87×10^{-2})	16 (2.46×10^{-2})
	2^{-4}	14 (7.66×10^{-2})	14 (5.19×10^{-2})	14 (4.44×10^{-2})
	2^{-5}	14 (3.06×10^{-1})	13 (1.29×10^{-1})	14 (1.22×10^{-1})
	2^{-6}	14 (2.40)	14 (6.02×10^{-1})	14 (5.60×10^{-1})
10^{-8}	2^{-2}	6 (1.19×10^{-2})	8 (2.46×10^{-2})	8 (1.88×10^{-2})
	2^{-3}	15 (2.31×10^{-2})	11 (2.73×10^{-2})	12 (2.38×10^{-2})
	2^{-4}	25 (1.23×10^{-1})	13 (4.98×10^{-2})	14 (4.42×10^{-2})
	2^{-5}	37 (6.20×10^{-1})	14 (1.39×10^{-1})	14 (1.24×10^{-1})
	2^{-6}	47 (4.84)	14 (6.05×10^{-1})	14 (5.59×10^{-1})

$G(\alpha_{\text{opt}})^{-1}$ or G^{-1} in such a way that the h-independent convergence of GMRES is preserved. This is confirmed by the numerical experiments described in Table 4. Here, we report the errors $\|x^{(20)} - x_*\|_2$ (and CPU times, in parentheses) after 20 Chebyshev semiiterations for $\alpha = 1$ and for the optimal parameter $\alpha = \alpha_{\text{opt}}$, with different h and β. In these experiments, the exact solution x_* to $G(\alpha)x = b$ is the vector of all ones and the initial guess is the zero vector. The results clearly show that the quality of the approximation deteriorates as h decreases and β increases, as expected.

One way to avoid this difficulty is to approximate the action of $G(\alpha_{\text{opt}})^{-1}$ or G^{-1} with one or more AMG V-cycles. This approach, while yielding robustness with respect to both h and β, is however more expensive in terms of CPU time than the nonoptimal Chebyshev-based approach, at least for the range of h values considered here. It is expected that for finer grids and/or three-dimensional problems, the use of AMG instead of Chebyshev semiiteration will be the recommended approach for larger values of β.

Next, from Table 3, we see that $F^{(\text{app})}$ performs as well as $F^{(\text{app})}(\alpha_{\text{opt}})$, as both of them produce nearly the same number of iterations and computing times. Therefore, in actual computations, we may adopt the inexact preconditioning matrices rather than the exact ones to accelerate the PPCG and the GMRES methods and, instead of the experimentally found optimal parameters α_{opt}, we may simply take $\alpha = 1$ to obtain good performance and effectively eliminate the need for parameter tuning.

In Table 5, we show the results obtained with MINRES preconditioned with the block diagonal preconditioners P_{D2}, $P_{D2}^{(\text{app})}$, P_D and $P_D^{(\text{app})}$ described earlier. From these results, we can see that $P_{D2}^{(\text{app})}$

TABLE 3 *Iteration counts and CPU timings (in seconds) for the inexact constraint preconditioner and for the inexact PMHSS preconditioners with optimal α and with $\alpha = 1$*

β	h	$P_C^{(\mathrm{app})}$	$F^{(\mathrm{app})}(\alpha_{\mathrm{opt}})$	$F^{(\mathrm{app})}$
10^{-2}	2^{-2}	3 (8.28×10^{-2})	9 (1.02×10^{-2})	9 (9.64×10^{-3})
	2^{-3}	3 (2.96×10^{-2})	9 (1.08×10^{-2})	11 (1.22×10^{-2})
	2^{-4}	3 (4.43×10^{-2})	11 (2.41×10^{-2})	13 (2.68×10^{-2})
	2^{-5}	4 (1.82×10^{-1})	21 (2.13×10^{-1})	23 (2.22×10^{-1})
	2^{-6}	7 (1.64)	48 (8.60×10^{-1})	48 (8.29×10^{-1})
10^{-4}	2^{-2}	5 (2.71×10^{-2})	9 (9.48×10^{-3})	9 (9.43×10^{-2})
	2^{-3}	5 (3.07×10^{-2})	12 (1.38×10^{-2})	13 (1.40×10^{-2})
	2^{-4}	6 (5.57×10^{-2})	14 (2.98×10^{-1})	16 (3.26×10^{-2})
	2^{-5}	6 (2.16×10^{-1})	15 (1.55×10^{-1})	19 (1.80×10^{-1})
	2^{-6}	6 (1.56)	31 (5.56×10^{-1})	34 (5.82×10^{-1})
10^{-6}	2^{-2}	7 (2.83×10^{-2})	11 (1.09×10^{-2})	12 (1.16×10^{-2})
	2^{-3}	14 (3.89×10^{-2})	13 (1.44×10^{-2})	16 (1.75×10^{-2})
	2^{-4}	34 (1.30×10^{-1})	14 (2.92×10^{-2})	14 (2.94×10^{-2})
	2^{-5}	38 (6.21×10^{-1})	13 (1.35×10^{-1})	14 (1.37×10^{-1})
	2^{-6}	41 (2.93)	13 (2.81×10^{-1})	18 (3.38×10^{-1})
10^{-8}	2^{-2}	6 (2.84×10^{-2})	8 (8.69×10^{-2})	8 (9.37×10^{-2})
	2^{-3}	22 (4.90×10^{-2})	11 (1.31×10^{-2})	12 (1.45×10^{-2})
	2^{-4}	88 (2.71×10^{-1})	13 (2.72×10^{-2})	14 (2.77×10^{-2})
	2^{-5}	283 (3.83)	14 (1.42×10^{-1})	14 (1.43×10^{-1})
	2^{-6}	178 (8.26)	14 (3.02×10^{-1})	14 (2.86×10^{-1})

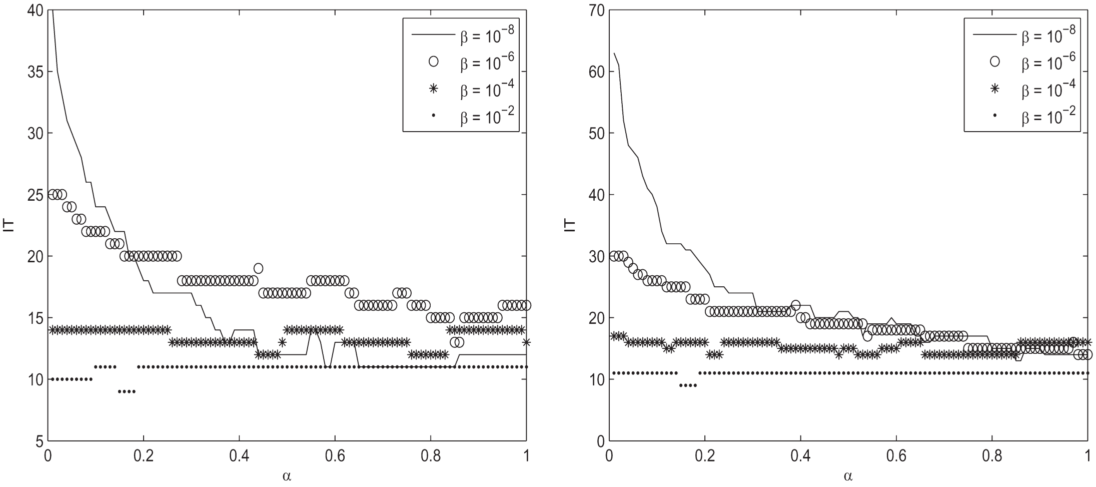

FIG. 1. Pictures of IT versus α for $F(\alpha)$-preconditioned GMRES when $\beta = 10^{-2}$, 10^{-4}, 10^{-6} and 10^{-8}, with $h = 2^{-3}$ (left) and $h = 2^{-4}$ (right).

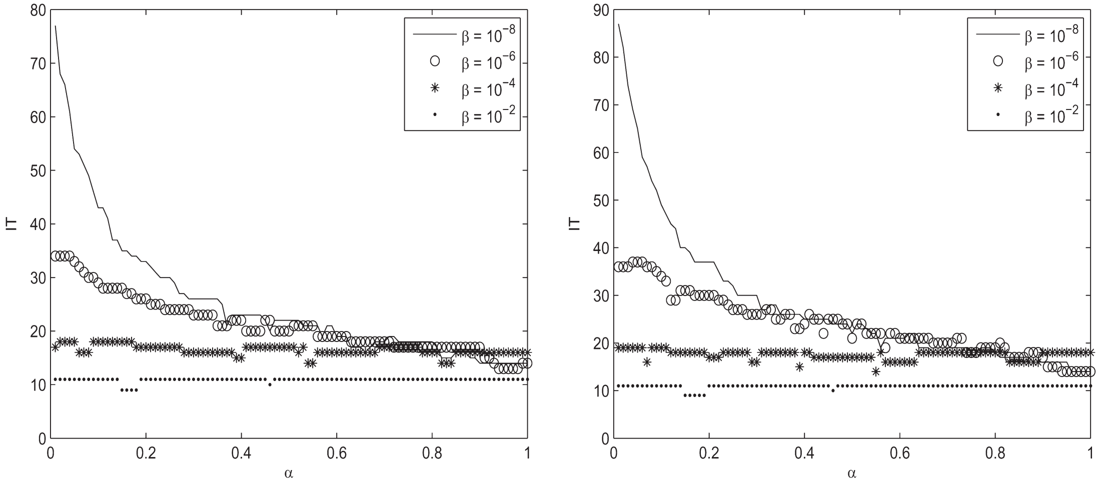

FIG. 2. Pictures of IT versus α for $F(\alpha)$-preconditioned GMRES when $\beta = 10^{-2}$, 10^{-4}, 10^{-6} and 10^{-8}, with $h = 2^{-5}$ (left) and $h = 2^{-6}$ (right).

used with MINRES is competitive for $\beta = 10^{-2}$ and h sufficiently small, while it is generally inferior to inexact PMHSS and constraint preconditioning in virtually all the remaining cases. Also, MINRES with the other block diagonal preconditioners is not competitive on this test problem. In particular, the results show that while the exact preconditioners P_D and, to a lesser extent, P_{D2} are very robust with respect to regularization and discretization parameters, their cost is too high to make them competitive as $h \to 0$. Moreover, their inexact variants lack robustness and drastically deteriorate as the regularization parameter is decreased, with no convergence within 1000 MINRES iterations in several cases (denoted by a dash in Table 5).

These results confirm the findings in Pearson & Wathen (2010), namely, that approximating the inverse of $K^{-1}MK^{-1}$ by replacing the exact inverse of the stiffness matrix K by a spectrally equivalent multilevel preconditioner gives poor results. As shown in Pearson & Wathen (2010), more favourable results can be expected by using the block diagonal preconditioner $\widehat{P}_1$ based on the following approximation of the Schur complement:

$$S \approx \widehat{S} = \left(K + \frac{1}{\sqrt{2\beta}}M\right) M^{-1} \left(K + \frac{1}{\sqrt{2\beta}}M\right);$$

see Pearson & Wathen (2010, (5) and (11)). In turn, the action of $\left(K + \frac{1}{\sqrt{2\beta}}M\right)^{-1}$ can be approximated by a suitable iterative method.

In the application of the preconditioning matrix $\widehat{P}_1^{(\text{app})}$, the inverse of M is again approximated by 20 steps of Chebyshev semiiteration. As for the inverse of $K + \frac{1}{\sqrt{2\beta}}M$, we find that the AMG implementation in COMSOL fails due to the fact that most off-diagonal entries are positive; in other words, the matrix is too far from being an M-matrix and the construction of the preconditioner breaks down in the coarsening phase. Similar problems were reported in Pearson & Wathen (2010) (which uses a different code for AMG) for certain values of h and β. To circumvent this problem, we use two different approximations. As before, we approximate the inverse of $K + \frac{1}{\sqrt{2\beta}}M$ by 20 steps of Chebyshev

TABLE 4 *The solution accuracy and CPU timings (in seconds) of 20-step Chebyshev approximation with optimal α and with $\alpha = 1$*

β			10^{-2}		10^{-4}		10^{-6}		10^{-8}	
		2^{-2}	2.17×10^{-6}	(3.28×10^{-3})	2.47×10^{-6}	(2.81×10^{-3})	2.56×10^{-6}	(2.75×10^{-3})	6.76×10^{-7}	(2.92×10^{-3})
		2^{-3}	1.36×10^{-1}	(2.86×10^{-3})	2.08×10^{-6}	(2.32×10^{-3})	5.43×10^{-6}	(2.81×10^{-3})	3.44×10^{-6}	(2.78×10^{-3})
$\alpha = \alpha_{\text{opt}}$	h	2^{-4}	5.18	(2.96×10^{-3})	3.72×10^{-1}	(3.02×10^{-3})	1.09×10^{-5}	(3.15×10^{-3})	1.10×10^{-5}	(3.08×10^{-3})
		2^{-5}	2.13×10^{1}	(5.84×10^{-3})	1.60×10^{1}	(5.54×10^{-3})	5.39×10^{-4}	(5.45×10^{-3})	2.50×10^{-5}	(5.32×10^{-3})
		2^{-6}	5.26×10^{1}	(7.36×10^{-3})	5.26×10^{1}	(7.27×10^{-3})	1.28×10^{1}	(7.05×10^{-3})	5.48×10^{-5}	(7.23×10^{-3})
		2^{-2}	2.17×10^{-6}	(3.09×10^{-3})	2.47×10^{-6}	(2.77×10^{-3})	1.51×10^{-6}	(2.32×10^{-3})	6.76×10^{-7}	(2.76×10^{-3})
		2^{-3}	3.68×10^{-2}	(2.45×10^{-3})	6.08×10^{-6}	(2.92×10^{-3})	3.96×10^{-6}	(2.97×10^{-3})	1.23×10^{-6}	(2.41×10^{-3})
$\alpha = 1$	h	2^{-4}	4.15	(3.16×10^{-3})	2.00×10^{-1}	(2.80×10^{-3})	1.09×10^{-5}	(3.30×10^{-3})	8.40×10^{-6}	(3.22×10^{-3})
		2^{-5}	2.00×10^{1}	(5.66×10^{-3})	1.02×10^{1}	(5.35×10^{-3})	3.22×10^{-4}	(5.41×10^{-3})	2.50×10^{-5}	(5.72×10^{-3})
		2^{-6}	5.26×10^{1}	(7.10×10^{-3})	4.46×10^{1}	(7.19×10^{-3})	7.91	(7.20×10^{-3})	5.48×10^{-5}	(7.64×10^{-3})

TABLE 5 *Iteration counts and CPU timings (in seconds) for block-diagonal preconditioned MINRES methods applied to solving the linear systems (3.4) and (4.1)*

β	h	P_{D2}	$P_{D2}^{(\mathrm{app})}$	P_D	$P_D^{(\mathrm{app})}$
10^{-2}	2^{-2}	7 (5.32×10^{-2})	7 (1.13×10^{-1})	4 (2.07×10^{-2})	8 (1.17×10^{-2})
	2^{-3}	7 (2.98×10^{-2})	7 (2.51×10^{-2})	4 (2.02×10^{-2})	8 (1.34×10^{-2})
	2^{-4}	7 (1.30×10^{-1})	7 (4.38×10^{-2})	4 (5.05×10^{-2})	6 (2.65×10^{-2})
	2^{-5}	7 (5.96×10^{-1})	8 (1.29×10^{-1})	4 (5.39×10^{-1})	8 (1.55×10^{-1})
	2^{-6}	6 (1.67×10^{1})	11 (5.41×10^{-1})	4 (1.61×10^{1})	14 (1.29)
10^{-4}	2^{-2}	11 (2.50×10^{-2})	12 (2.30×10^{-2})	4 (1.72×10^{-2})	13 (1.46×10^{-2})
	2^{-3}	13 (3.61×10^{-2})	17 (3.66×10^{-2})	4 (2.02×10^{-2})	17 (2.29×10^{-2})
	2^{-4}	13 (8.35×10^{-2})	25 (1.01×10^{-1})	4 (4.60×10^{-2})	24 (7.65×10^{-2})
	2^{-5}	13 (7.24×10^{-1})	— (1.21×10^{1})	4 (5.44×10^{-1})	26 (3.87×10^{-1})
	2^{-6}	13 (2.01×10^{1})	27 (1.26)	4 (1.61×10^{1})	24 (1.66)
10^{-6}	2^{-2}	6 (2.58×10^{-2})	11 (2.41×10^{-2})	4 (1.78×10^{-2})	14 (1.66×10^{-2})
	2^{-3}	11 (3.13×10^{-2})	63 (8.74×10^{-2})	4 (2.00×10^{-2})	63 (7.19×10^{-2})
	2^{-4}	13 (8.04×10^{-2})	675 (2.18)	4 (4.60×10^{-2})	172 (4.94×10^{-1})
	2^{-5}	13 (7.24×10^{-1})	— (1.21×10^{1})	4 (5.30×10^{-1})	— (1.26×10^{1})
	2^{-6}	13 (2.01×10^{1})	— (4.45×10^{1})	4 (1.62×10^{1})	170 (7.31)
10^{-8}	2^{-2}	4 (2.43×10^{-2})	10 (2.24×10^{-2})	4 (1.77×10^{-2})	14 (1.55×10^{-2})
	2^{-3}	4 (2.63×10^{-2})	73 (9.90×10^{-2})	4 (2.03×10^{-2})	116 (1.32×10^{-1})
	2^{-4}	6 (6.01×10^{-2})	675 (2.20)	4 (4.65×10^{-2})	675 (1.90)
	2^{-5}	11 (6.80×10^{-1})	— (1.21×10^{1})	4 (5.33×10^{-1})	— (1.27×10^{1})
	2^{-6}	11 (1.91×10^{1})	— (4.47×10^{1})	4 (1.62×10^{1})	— (3.93×10^{1})

semiiteration (which works well for smaller values of β); alternatively, we replace the mass matrix in $K + \frac{1}{\sqrt{2\beta}}M$ by its diagonal, leading to a new approximation of the Schur complement, namely,

$$S \approx \widetilde{S} = \left(K + \frac{1}{\sqrt{2\beta}}\,\mathrm{diag}(M)\right) M^{-1} \left(K + \frac{1}{\sqrt{2\beta}}\,\mathrm{diag}(M)\right).$$

The replacement of $\widehat{S}$ by $\widetilde{S}$ in $\widehat{P}_1$ then leads to a preconditioner $\widetilde{P}_1$ for the coefficient matrix of the linear system (3.4). As the number of positive off-diagonal elements of the matrix $K + \frac{1}{\sqrt{2\beta}}\mathrm{diag}(M)$ is small relative to the number of nonpositive ones, we can use the algebraic multigrid method described earlier to approximate the inverse of the matrix $K + \frac{1}{\sqrt{2\beta}}\,\mathrm{diag}(M)$. This results in an approximation $\widetilde{P}_1^{(\mathrm{app})}$ to the matrix $\widetilde{P}_1$. The numerical results for the block diagonal preconditioning matrices $\widehat{P}_1$, $\widehat{P}_1^{(\mathrm{app})}$ and $\widetilde{P}_1^{(\mathrm{app})}$ applied to precondition the MINRES method for solving the linear system (3.4) are given in Table 6. These results show fairly good robustness of the preconditioners $\widehat{P}_1$ and $\widetilde{P}_1^{(\mathrm{app})}$.

Comparing the results reported in Table 6 with those for the inexact PMHSS preconditioners $F^{(\mathrm{app})}(\alpha_{\mathrm{opt}})$ and $F^{(\mathrm{app})}$ in Table 3, we find in nearly all cases that PMHSS results in faster convergence and smaller CPU timings.

TABLE 6 *Iteration counts and CPU timings (in seconds) for block diagonal preconditioned MINRES methods in Pearson & Wathen (2010) applied to solving the linear system (3.4)*

β	h	$\widehat{P}_1$	$\widehat{P}_1^{(\text{app})}$	$\widetilde{P}_1^{(\text{app})}$
10^{-2}	2^{-2}	10 (3.22×10^{-2})	10 (1.65×10^{-2})	8 (2.87×10^{-1})
	2^{-3}	11 (3.30×10^{-2})	11 (2.01×10^{-2})	10 (5.57×10^{-2})
	2^{-4}	12 (7.39×10^{-2})	14 (5.33×10^{-2})	10 (7.85×10^{-2})
	2^{-5}	12 (1.80×10^{-1})	24 (3.69×10^{-1})	12 (2.64×10^{-1})
	2^{-6}	12 (1.48)	66 (2.19)	15 (1.44)
10^{-4}	2^{-2}	10 (2.89×10^{-2})	10 (1.63×10^{-2})	12 (5.59×10^{-2})
	2^{-3}	12 (3.59×10^{-2})	12 (2.34×10^{-2})	12 (5.71×10^{-2})
	2^{-4}	14 (7.48×10^{-2})	14 (4.68×10^{-2})	12 (8.20×10^{-2})
	2^{-5}	14 (2.06×10^{-1})	20 (3.13×10^{-1})	15 (2.99×10^{-1})
	2^{-6}	14 (1.59)	50 (1.87)	15 (1.41)
10^{-6}	2^{-2}	10 (2.82×10^{-2})	10 (1.66×10^{-2})	14 (5.81×10^{-2})
	2^{-3}	12 (3.38×10^{-2})	12 (2.35×10^{-2})	32 (8.12×10^{-2})
	2^{-4}	12 (6.92×10^{-2})	12 (4.42×10^{-2})	25 (1.19×10^{-1})
	2^{-5}	12 (1.82×10^{-1})	12 (2.17×10^{-1})	22 (3.83×10^{-1})
	2^{-6}	14 (1.58)	14 (1.04)	20 (1.61)
10^{-8}	2^{-2}	8 (3.05×10^{-2})	8 (1.44×10^{-2})	14 (5.89×10^{-2})
	2^{-3}	10 (3.25×10^{-2})	10 (1.87×10^{-2})	45 (9.47×10^{-2})
	2^{-4}	12 (6.86×10^{-2})	12 (4.09×10^{-2})	62 (2.23×10^{-1})
	2^{-5}	12 (1.77×10^{-1})	12 (2.20×10^{-1})	42 (6.23×10^{-1})
	2^{-6}	12 (1.47)	12 (1.01)	28 (1.90)

5. Concluding remarks

The PMHSS iteration method provides a fairly general framework for solving block two-by-two linear systems arising from the discretization of a class of distributed control problems. For this class of linear systems, choosing $\mathbf{V} = \mathbf{W}$ (which is just a mass matrix) one can construct high-quality preconditioners for which the eigenvalues of the preconditioned matrices are clustered within complex disks centred at 1 with radii $\delta(\alpha) := \frac{\sqrt{\alpha^2+1}}{\alpha+1}$, and the matrices of the corresponding eigenvectors are unitary, where $\alpha > 0$ is the iteration parameter. Note that when $\alpha = 1$, it holds that $\delta(1) = \frac{\sqrt{2}}{2}$. In practice, the preconditioner is applied inexactly and these bounds no longer hold. Nevertheless, numerical experiments on a standard test problem indicate that the proposed preconditioner compares favourably with some of the best existing solvers, including constrained and block diagonal preconditioning.

In general, the PMHSS iteration method can be applied to solve a reasonably broad class of block two-by-two linear systems, for which the symmetric matrices $\mathbf{W}$ and $\mathbf{T}$ are only required to satisfy the conditions that each of them is either positive or negative semidefinite and that $\text{null}(\mathbf{W}) \cap \text{null}(\mathbf{T}) = \{0\}$. Under these conditions, we can analogously demonstrate that the block two-by-two matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ is nonsingular and the PMHSS iteration method is unconditionally convergent; moreover, for the special case when $\mathbf{V} = \mathbf{W}$ or $\mathbf{V} = \mathbf{T}$ (or $\mathbf{V} = -\mathbf{W}$ or $\mathbf{V} = -\mathbf{T}$) depending on whether $\mathbf{W}$ or $\mathbf{T}$ is positive (or negative) semidefinite, we can demonstrate that the convergence factor of the PMHSS iteration method is at most $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$, the eigenvalues of the PMHSS-preconditioned matrix are included in a complex disk centred at 1 with radius $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$ and the matrix of eigenvectors of the PMHSS-preconditioned matrix is unitary with condition number equal to 1. More generally, the block two-by-two matrix $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$ is nonsingular and Theorem 2.4 holds true even when the matrix $\alpha\mathbf{W} + \mathbf{T}$ is symmetric and is either positive or negative definite, which may allow the real symmetric matrix $\mathbf{W}$ or $\mathbf{T}$ to be indefinite. In fact, if the matrix $\alpha\mathbf{W} + \mathbf{T}$ is symmetric negative definite, we may simply multiply -1 through both sides of the linear system (1.1), obtaining a new block two-by-two linear system satisfying the requirement that $\alpha\mathbf{W} + \mathbf{T}$ is a symmetric positive definite matrix. These observations equally apply to the PMHSS iteration method introduced and discussed in Bai *et al.* (2011) for solving complex symmetric linear systems of the form $\mathbf{A}_c\mathbf{x}_c = \mathbf{b}_c$, with $\mathbf{A}_c = \mathbf{W} + \mathrm{i}\,\mathbf{T} \in \mathbb{C}^{n\times n}$ and $\mathbf{b}_c = \mathbf{p} + \mathrm{i}\,\mathbf{q} \in \mathbb{C}^n$.

Acknowledgements

The authors are very much indebted to Tyrone Rees and Sue Thorne for help with the MATLAB code for generating the test problems, and to the Editor and anonymous referees for helpful suggestions.

Funding

The National Natural Science Foundation for Innovative Research Groups (No. 11021101); The Hundred Talent Project of the Chinese Academy of Sciences; and The National Basic Research Program (No. 2011CB309703), People's Republic of China.

References

Axelsson, O. & Kucherov, A. (2000) Real valued iterative methods for solving complex symmetric linear systems. *Numer. Linear Algebra Appl.*, **7**, 197–218.

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

Bai, Z.-Z., Benzi, M. & Chen, F. (2010) Modified HSS iteration methods for a class of complex symmetric linear systems. *Computing*, **87**, 93–111.

Bai, Z.-Z., Benzi, M. & Chen, F. (2011) On preconditioned MHSS iteration methods for complex symmetric linear systems. *Numer. Algorithms*, **56**, 297–317.

Bai, Z.-Z., Golub, G. H. & Li, C.-K. (2007) Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. *Math. Comp.*, **76**, 287–298.

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

Bai, Z.-Z., Golub, G. H. & Ng, M. K. (2008) On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *Linear Algebra Appl.*, **428**, 413–440.

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

Bai, Z.-Z. & Ng, M. K. (2005) On inexact preconditioners for nonsymmetric matrices. *SIAM J. Sci. Comput.*, **26**, 1710–1724.

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

Benzi, M. & Golub, G. H. (2004) A preconditioner for generalized saddle point problems. *SIAM J. Matrix Anal. Appl.*, **26**, 20–41.

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

Bertaccini, D., Golub, G. H., Capizzano, S. S. & Tablino Possio, C. (2005) Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation. *Numer. Math.*, **99**, 441–484.

Chen, K. (2005) *Matrix Preconditioning Techniques and Applications*. Cambridge, UK: Cambridge University Press.

Day, D. D. & Heroux, M. A. (2001) Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, **23**, 480–498.

Ito, K. & Kunisch, K. (2008) *Lagrange Multiplier Approach to Variational Problems and Applications*. Philadelphia, PA: SIAM.

Keller, C., Gould, N. I. M. & Wathen, A. J. (2000) Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, **21**, 1300–1317.

Lass, O., Vallejos, M., Borzì, A. & Douglas, C. C. (2009) Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems. *Computing*, **84**, 27–48.

Lions, J. L. (1968) *Optimal Control of Systems Governed by Partial Differential Equations*. Berlin, Germany: Springer.

Pearson, J. W. & Wathen, A. J. (2010) A new approximation of the Schur complement in preconditioners for PDE constrained optimization. *Report no. 10/14*. Oxford: University of Oxford, Oxford University Mathematical Institute.

Rees, T., Dollar, H. S. & Wathen, A. J. (2010) Optimal solvers for PDE-constrained optimization. *SIAM J. Sci. Comput.*, **32**, 271–298.

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

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, 2nd edn. Philadelphia, PA: SIAM.

Simoncini, V. (2011) Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems. *Preprint*.

Takacs, S. & Zulehner, W. (2011) Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. *NuMa-Report No. 2011-02*. Linz, Austria: Johannes Kepler University.

Tröltzsch, F. (2010) *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Providence, RI: American Mathematical Society.

Zulehner, W. (2011) Nonstandard norms and robust estimates for saddle point problems. *SIAM J. Matrix Anal. Appl.*, **32**, 536–560.