On Preconditioned MHSS Iteration Methods for Complex Symmetric Linear Systems*

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, P.O. Box 2719, Beijing 100190, P.R.China
Email: bzz@lsec.cc.ac.cn

Michele Benzi
Department of Mathematics and Computer Science
Emory University, Atlanta, GA 30322, USA
Email: benzi@mathcs.emory.edu

Fang Chen
Department of Mathematics and Physics
Xi'an University of Post and Telecommunications
Xi'an 710121, P.R. China
Email: chenfreesky@gmail.com

November 25, 2010

Abstract

We propose a preconditioned variant of the *modified HSS* (**MHSS**) iteration method for solving a class of complex symmetric systems of linear equations. Under suitable conditions, we prove the convergence of the *preconditioned MHSS* (**PMHSS**) iteration method and discuss the spectral properties of the PMHSS-preconditioned matrix. Numerical implementations show that the resulting PMHSS preconditioner leads to fast convergence when it is used to precondition Krylov subspace iteration methods such as GMRES and its restarted variants. In particular, both the stationary PMHSS iteration and PMHSS-preconditioned GMRES show meshsize-independent and parameter-insensitive convergence behavior for the tested numerical examples.

Keywords: complex symmetric linear system, MHSS iteration, preconditioning, convergence theory, spectral properties.

AMS(MOS) Subject Classifications: 65F10, 65F50; CR: G1.3.

*Supported by The National Natural Science Foundation for Innovative Research Groups (No. 11021101), by The Hundred Talent Project of Chinese Academy of Sciences, by The National Basic Research Program (No. 2011CB309703), P.R. China., and by the US National Science Foundation grant DMS-0810862.

1 Introduction

Consider the iterative solution of the system of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \qquad \mathbf{A} \in \mathbb{C}^{n\times n} \quad \text{and} \quad \mathbf{x}, \mathbf{b} \in \mathbb{C}^n, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{C}^{n\times n}$ is a complex symmetric matrix of the form

$$\mathbf{A} = \mathbf{W} + \imath\, \mathbf{T}, \tag{1.2}$$

and $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semidefinite matrices with at least one of them, e.g., $\mathbf{W}$, being positive definite. Here and in the sequel, we use $\imath = \sqrt{-1}$ to denote the imaginary unit. For more details about the practical backgrounds of this class of problems, we refer to [1, 10, 9, 4] and the references therein.

The Hermitian and skew-Hermitian parts of the complex symmetric matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ are given by

$$\mathbf{H} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^*) = \mathbf{W} \quad \text{and} \quad \mathbf{S} = \frac{1}{2}(\mathbf{A} - \mathbf{A}^*) = \imath\, \mathbf{T},$$

respectively. Hence, when $\mathbf{W}$ is symmetric positive definite, $\mathbf{A} \in \mathbb{C}^{n\times n}$ is a non-Hermitian, but positive definite matrix. Here $\mathbf{A}^*$ is used to denote the conjugate transpose of the matrix $\mathbf{A}$. Based on the *Hermitian and skew-Hermitian splitting* (**HSS**)

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

of the matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, we can straightforwardly employ the HSS iteration method, introduced in Bai, Golub and Ng [6], to compute an approximate solution for the complex symmetric linear system (1.1)-(1.2). Recently, by making use of the special structure of the coefficient matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, Bai, Benzi and Chen designed in [4] a *modified HSS* (**MHSS**) iteration method, which is much more efficient than the HSS iteration method for solving the complex symmetric linear system (1.1)-(1.2). This MHSS iteration method is algorithmically described in the following.

Method 1.1. (The MHSS Iteration Method)
Let $\mathbf{x}^{(0)} \in \mathbb{C}^n$ be an arbitrary initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges, compute the next iterate $\mathbf{x}^{(k+1)}$ according to the following procedure:

$$\begin{cases} (\alpha\mathbf{I} + \mathbf{W})\mathbf{x}^{(k+\frac{1}{2})} &= (\alpha\mathbf{I} - \imath\, \mathbf{T})\mathbf{x}^{(k)} + \mathbf{b}, \\ (\alpha\mathbf{I} + \mathbf{T})\mathbf{x}^{(k+1)} &= (\alpha\mathbf{I} + \imath\, \mathbf{W})\mathbf{x}^{(k+\frac{1}{2})} - \imath\, \mathbf{b}, \end{cases}$$

where α is a given positive constant and $\mathbf{I}$ represents the identity matrix.

As $\mathbf{W} \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $\mathbf{T} \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite, and $\alpha \in \mathbb{R}$ is positive, we see that both matrices $\alpha\mathbf{I} + \mathbf{W}$ and $\alpha\mathbf{I} + \mathbf{T}$ are symmetric positive definite. Hence, the two linear sub-systems involved in each step of the MHSS iteration can be solved effectively using mostly real arithmetic either exactly by a Cholesky factorization or inexactly by some conjugate gradient or multigrid scheme. This is different from the HSS iteration method,

in which a shifted skew-Hermitian linear sub-system with coefficient matrix $\alpha\mathbf{I} + \imath\,\mathbf{T}$ needs to be solved at every iteration step; see [6, 7]. If sparse triangular factorizations are used to solve the linear sub-systems involved at each step, the MHSS iteration method is likely to require considerably less storage than the HSS iteration method since only two triangular factors rather than three have to be computed and stored. For more details, we refer to [6, 7, 4].

Theoretical analysis in [4] has indicated that the MHSS iteration converges to the unique solution of the complex symmetric linear system (1.1)-(1.2) for any initial guess, and its asymptotic convergence rate is bounded by

$$\sigma(\alpha) \equiv \max_{\lambda_j \in \mathrm{sp}(\mathbf{W})} \frac{\sqrt{\alpha^2+\lambda_j^2}}{\alpha+\lambda_j} \cdot \max_{\mu_j \in \mathrm{sp}(\mathbf{T})} \frac{\sqrt{\alpha^2+\mu_j^2}}{\alpha+\mu_j} \leq \max_{\lambda_j \in \mathrm{sp}(\mathbf{W})} \frac{\sqrt{\alpha^2+\lambda_j^2}}{\alpha+\lambda_j} < 1, \qquad \forall \alpha > 0,$$

where $\mathrm{sp}(\mathbf{W})$ and $\mathrm{sp}(\mathbf{T})$ denote the spectra of the matrices $\mathbf{W}$ and $\mathbf{T}$, respectively. Note that this bound only depends on the eigenvalues of the symmetric positive definite matrix $\mathbf{W}$ and/or the symmetric positive semidefinite matrix $\mathbf{T}$. In particular, for the choice $\alpha_\star = \sqrt{\gamma_{\min}\gamma_{\max}}$, with $\gamma_{\min}$ and $\gamma_{\max}$ being the smallest and the largest eigenvalues of the matrix $\mathbf{W}$, it holds that

$$\sigma(\alpha_\star) \leq \frac{\sqrt{\kappa_2(\mathbf{W})+1}}{\sqrt{\kappa_2(\mathbf{W})}+1}.$$

Here and in the sequel, $\kappa_2(\cdot)$ is used to represent the spectral condition number of the corresponding matrix.

To further generalize the MHSS iteration method and accelerate its convergence rate, in this paper we propose a *preconditioned MHSS* (**PMHSS**) iteration method for solving the complex symmetric linear system (1.1)-(1.2). This iteration scheme is, in spirit, analogous to the preconditioned HSS iteration methods discussed in [8, 5] for solving the non-Hermitian positive definite linear systems. We establish the convergence theory for the PMHSS iteration method under the condition that both $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite and, at least, one of them is positive definite. The PMHSS iteration method naturally leads to a preconditioning matrix for the complex symmetric matrix $\mathbf{A}$. For some special cases of the PMHSS iteration method, we prove their convergence theorems under the weaker condition 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. Moreover, for these special PMHSS preconditioners, we investigate the spectral properties of the preconditioned matrices in detail, which show that the eigenvalues of the preconditioned matrices are clustered within complex disks centered at 1 with radii $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$, and the condition numbers of the corresponding eigenvectors are equal to $\sqrt{\kappa_2(\alpha\mathbf{W}+\mathbf{T})}$, where $\alpha > 0$ is the iteration parameter [1]. Numerical results show that the PMHSS iteration methods, when used to precondition the Krylov subspace methods such as GMRES and its restarted variants, say, GMRES(#), can lead to satisfactory experimental results, and they have higher computing efficiency than the MHSS preconditioner proposed in [4]. Moreover, as both solver and preconditioner, the PMHSS iteration method shows mesh-independent and parameter-insensitive convergence behavior for the tested numerical examples.

The organization of the paper is as follows. In Section 2 we describe the PMHSS iteration method. In Section 3, we establish the convergence theory of the PMHSS iteration method and

[1] See Remark 3.1 for a discussion of the condition number of the eigenvector matrix.

discuss the spectral properties of the PMHSS preconditioning matrix under suitable conditions. Numerical results are given in Section 4 to show the effectiveness of this PMHSS iteration method as well as the corresponding PMHSS preconditioner. Finally, in Section 5 we put forth some conclusions and remarks to end the paper.

2 The PMHSS Iteration Method

In order to further accelerate the convergence rate of Method 1.1, we may precondition the complex symmetric linear system (1.1)-(1.2) by choosing a symmetric positive definite matrix, say, $\mathbf{V} \in \mathbb{R}^{n\times n}$. More concretely, with the notations

$$\widetilde{\mathbf{W}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{W}\mathbf{V}^{-\frac{1}{2}}, \quad \widetilde{\mathbf{T}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{T}\mathbf{V}^{-\frac{1}{2}}, \quad \widetilde{\mathbf{A}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{A}\mathbf{V}^{-\frac{1}{2}} \tag{2.1}$$

and

$$\widetilde{\mathbf{x}} = \mathbf{V}^{\frac{1}{2}}\mathbf{x}, \quad \widetilde{\mathbf{b}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{b}, \tag{2.2}$$

the system of linear equations (1.1)-(1.2) can be equivalently transformed into the preconditioned variant

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}, \tag{2.3}$$

where $\widetilde{\mathbf{A}} \in \mathbb{C}^{n\times n}$ is a complex symmetric matrix of the form

$$\widetilde{\mathbf{A}} = \widetilde{\mathbf{W}} + \imath\,\widetilde{\mathbf{T}}, \tag{2.4}$$

and $\widetilde{\mathbf{W}}, \widetilde{\mathbf{T}} \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semidefinite matrices, with $\widetilde{\mathbf{W}}$ being positive definite.

Now, we can first apply Method 1.1 directly to the preconditioned linear system (2.3)-(2.4) defined through (2.1)-(2.2), and then recover the resulting iteration scheme to the original variable, obtaining the *preconditioned MHSS* (**PMHSS**) iteration method described as follows:

Method 2.1. (The PMHSS Iteration Method)
Let $\mathbf{x}^{(0)} \in \mathbb{C}^n$ be an arbitrary initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges, compute the next iterate $\mathbf{x}^{(k+1)}$ according to the following procedure:

$$\begin{cases} (\alpha\mathbf{V} + \mathbf{W})\mathbf{x}^{(k+\frac{1}{2})} &= (\alpha\mathbf{V} - \imath\,\mathbf{T})\mathbf{x}^{(k)} + \mathbf{b}, \\ (\alpha\mathbf{V} + \mathbf{T})\mathbf{x}^{(k+1)} &= (\alpha\mathbf{V} + \imath\,\mathbf{W})\mathbf{x}^{(k+\frac{1}{2})} - \imath\,\mathbf{b}, \end{cases}$$

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

Note that Method 1.1 is a special case of Method 2.1 when $\mathbf{V} = \mathbf{I}$.

As $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n\times n}$ are symmetric positive definite, $\mathbf{T} \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite, and $\alpha \in \mathbb{R}$ is positive, we see that both matrices $\alpha\mathbf{V} + \mathbf{W}$ and $\alpha\mathbf{V} + \mathbf{T}$ are symmetric positive

definite. Hence, the two linear sub-systems involved in each step of the PMHSS iteration can also be solved effectively using mostly real arithmetic either exactly by a Cholesky factorization or inexactly by some conjugate gradient or multigrid scheme.

After straightforward derivations we can reformulate the PMHSS iteration scheme into the standard form

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

where

$$\mathbf{L}(\mathbf{V};\alpha) = (\alpha\mathbf{V} + \mathbf{T})^{-1}(\alpha\mathbf{V} + \imath\,\mathbf{W})(\alpha\mathbf{V} + \mathbf{W})^{-1}(\alpha\mathbf{V} - \imath\,\mathbf{T})$$

and

$$\mathbf{R}(\mathbf{V};\alpha) = (1-\imath)\alpha(\alpha\mathbf{V} + \mathbf{T})^{-1}\mathbf{V}(\alpha\mathbf{V} + \mathbf{W})^{-1}.$$

Note that $\mathbf{L}(\mathbf{V};\alpha)$ is the iteration matrix of the PMHSS iteration method.

In addition, if we introduce matrices

$$\mathbf{F}(\mathbf{V};\alpha) = \frac{1+\imath}{2\alpha}(\alpha\mathbf{V} + \mathbf{W})\mathbf{V}^{-1}(\alpha\mathbf{V} + \mathbf{T}) \tag{2.5}$$

and

$$\mathbf{G}(\mathbf{V};\alpha) = \frac{1+\imath}{2\alpha}(\alpha\mathbf{V} + \imath\,\mathbf{W})\mathbf{V}^{-1}(\alpha\mathbf{V} - \imath\,\mathbf{T}),$$

then it holds that

$$\mathbf{A} = \mathbf{F}(\mathbf{V};\alpha) - \mathbf{G}(\mathbf{V};\alpha) \quad \text{and} \quad \mathbf{L}(\mathbf{V};\alpha) = \mathbf{F}(\mathbf{V};\alpha)^{-1}\mathbf{G}(\mathbf{V};\alpha). \tag{2.6}$$

Therefore, the PMHSS iteration scheme is induced by the matrix splitting $\mathbf{A} = \mathbf{F}(\mathbf{V};\alpha) - \mathbf{G}(\mathbf{V};\alpha)$ defined in (2.6). It follows that the splitting matrix $\mathbf{F}(\mathbf{V};\alpha)$ can be used as a preconditioner for the complex symmetric matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, which is referred as the PMHSS preconditioner.

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

$$\mathbf{L}(\alpha) := \mathbf{L}(\mathbf{W};\alpha) = \frac{\alpha+\imath}{\alpha+1}(\alpha\mathbf{W} + \mathbf{T})^{-1}(\alpha\mathbf{W} - \imath\,\mathbf{T})$$

and

$$\mathbf{R}(\alpha) := \mathbf{R}(\mathbf{W};\alpha) = \frac{\alpha(1-\imath)}{\alpha+1}(\alpha\mathbf{W} + \mathbf{T})^{-1};$$

and the PMHSS iteration scheme is now induced by the matrix splitting

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

with

$$\mathbf{F}(\alpha) := \mathbf{F}(\mathbf{W};\alpha) = \frac{(\alpha+1)(1+\imath)}{2\alpha}(\alpha\mathbf{W} + \mathbf{T}) \tag{2.7}$$

and

$$\mathbf{G}(\alpha) := \mathbf{G}(\mathbf{W};\alpha) = \frac{(\alpha+\imath)(1+\imath)}{2\alpha}(\alpha\mathbf{W} - \imath\,\mathbf{T}).$$

3 Theoretical Results

Because $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{T}}$ defined in (2.1) are similar to $\mathbf{V}^{-1}\mathbf{W}$ and $\mathbf{V}^{-1}\mathbf{T}$, respectively, analogously to Theorem 2.1 in [4] we can prove that the PMHSS iteration converges to the unique solution of the complex symmetric linear system (1.1)-(1.2) for any initial guess, and its convergence rate is bounded by

$$\begin{aligned}\sigma(\alpha) &\equiv \max_{\widetilde{\lambda}_j\in\mathrm{sp}(\mathbf{V}^{-1}\mathbf{W})}\frac{\sqrt{\alpha^2+\widetilde{\lambda}_j^2}}{\alpha+\widetilde{\lambda}_j}\cdot\max_{\widetilde{\mu}_j\in\mathrm{sp}(\mathbf{V}^{-1}\mathbf{T})}\frac{\sqrt{\alpha^2+\widetilde{\mu}_j^2}}{\alpha+\widetilde{\mu}_j}\\ &\leq \max_{\widetilde{\lambda}_j\in\mathrm{sp}(\mathbf{V}^{-1}\mathbf{W})}\frac{\sqrt{\alpha^2+\widetilde{\lambda}_j^2}}{\alpha+\widetilde{\lambda}_j}<1,\quad\forall\alpha>0.\end{aligned}\tag{3.1}$$

In particular, 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

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

Evidently, the smaller the condition number of the matrix $\mathbf{V}^{-1}\mathbf{W}$ is, the faster the asymptotic convergence rate of the PMHSS iteration will be.

Moreover, when $\mathbf{V}=\mathbf{W}$ it holds that

$$\rho(\mathbf{L}(\alpha))\leq\frac{\sqrt{\alpha^2+1}}{\alpha+1}<1,\qquad\forall\alpha>0.$$

Here and in the sequel, we use $\rho(\cdot)$ to denote the spectral radius of the corresponding matrix. Note that this upper bound is a constant independent of both data and size of the problem. It implies that when $\mathbf{F}(\alpha)$ defined in (2.7) is used to precondition the matrix $\mathbf{A}\in\mathbb{C}^{n\times n}$, the eigenvalues of the preconditioned matrix $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ are clustered within the complex disk centered at 1 with radius $\frac{\sqrt{\alpha^2+1}}{\alpha+1}$ due to $\mathbf{F}(\alpha)^{-1}\mathbf{A}=\mathbf{I}-\mathbf{L}(\alpha)$; see [2, 3]. When $\alpha=1$, this radius becomes $\frac{\sqrt{2}}{2}$.

For the above-mentioned special case, we can further prove the convergence of the 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 stated in the following theorem.

Theorem 3.1. *Let $\mathbf{A}=\mathbf{W}+\imath\,\mathbf{T}\in\mathbb{C}^{n\times n}$, with $\mathbf{W}\in\mathbb{R}^{n\times n}$ 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* $\mathrm{null}(\mathbf{W})\cap\mathrm{null}(\mathbf{T})=\{0\}$;

(ii) if $\mathrm{null}(\mathbf{W})\cap\mathrm{null}(\mathbf{T})=\{0\}$, *the spectral radius of the PMHSS iteration matrix* $\mathbf{L}(\alpha)$ *satisfies* $\rho(\mathbf{L}(\alpha))\leq\sigma(\alpha)$, *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*

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

i.e., the PMHSS iteration converges unconditionally to the unique solution of the complex symmetric linear system (1.1)-(1.2) for any initial guess.

Proof. Note that the matrix $\mathbf{A}$ is nonsingular if and only if the matrix $\widehat{\mathbf{A}} = (1-\imath)\mathbf{A}$ is nonsingular. Evidently, $\widehat{\mathbf{A}} = (\mathbf{W}+\mathbf{T}) - \imath(\mathbf{W}-\mathbf{T})$, with its Hermitian part being given by $\mathbf{W}+\mathbf{T}$. Hence, when both matrices $\mathbf{W}$ and $\mathbf{T}$ are symmetric positive semidefinite, we know that $\widehat{\mathbf{A}}$ is nonsingular if and only if $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$. This shows the validity of (i).

We now turn to the proof of (ii). For all $\alpha > 0$, $\mathbf{W}$ and $\mathbf{T}$ being symmetric positive semidefinite matrices and $\mathrm{null}(\mathbf{W}) \cap \mathrm{null}(\mathbf{T}) = \{0\}$ readily imply that the matrix $\alpha\mathbf{W}+\mathbf{T}$ is symmetric positive definite. Therefore, by straightforward computations we have

$$\begin{aligned}
\rho(\mathbf{L}(\alpha)) &= \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \rho((\alpha\mathbf{W}+\mathbf{T})^{-1}(\alpha\mathbf{W} - \imath\,\mathbf{T})) \\
&= \frac{\sqrt{\alpha^2+1}}{2(\alpha+1)} \cdot \rho((1-\imath)(\alpha\mathbf{W}+\mathbf{T})^{-1}(\alpha\mathbf{W} - \imath\,\mathbf{T})(1+\imath)) \\
&= \frac{\sqrt{\alpha^2+1}}{2(\alpha+1)} \cdot \rho((1-\imath)(\alpha\mathbf{W}+\mathbf{T})^{-1}[(\alpha\mathbf{W}+\mathbf{T}) + \imath(\alpha\mathbf{W}-\mathbf{T})]) \\
&= \frac{\sqrt{\alpha^2+1}}{2(\alpha+1)} \cdot \rho((1-\imath)[\mathbf{I} + \imath(\alpha\mathbf{W}+\mathbf{T})^{-1}(\alpha\mathbf{W}-\mathbf{T})]) \\
&= \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \max_{\mu^{(\alpha)} \in \mathrm{sp}(\widetilde{\mathbf{Z}}^{(\alpha)})} \left| \frac{1+\imath\mu^{(\alpha)}}{1+\imath} \right| \\
&= \frac{\sqrt{\alpha^2+1}}{\alpha+1} \cdot \max_{\mu^{(\alpha)} \in \mathrm{sp}(\widetilde{\mathbf{Z}}^{(\alpha)})} \sqrt{\frac{1+|\mu^{(\alpha)}|^2}{2}} \\
&= \sigma(\alpha).
\end{aligned}$$

It easily follows from $\mu^{(\alpha)} \in [-1,\, 1]$ that $\frac{1}{2}(1+|\mu^{(\alpha)}|^2) \leq 1$ and, therefore,

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

□

The spectral properties of the preconditioning matrix $\mathbf{F}(\alpha)$ are established in the following theorem.

Theorem 3.2. *Let* $\mathbf{A} = \mathbf{W} + \imath\,\mathbf{T} \in \mathbb{C}^{n\times n}$, *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 orthogonal eigenvectors. Then the eigenvalues of the matrix* $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ *are given by*

$$\lambda_j^{(\alpha)} = \frac{\alpha[(\alpha+1) - \imath(\alpha-1)](1-\imath\mu_j^{(\alpha)})}{(\alpha+1)(\alpha^2+1)}, \qquad j = 1, 2, \ldots, n,$$

and the corresponding eigenvectors are given by

$$\mathbf{x}_j^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{q}_j^{(\alpha)}, \qquad j = 1, 2, \ldots, n.$$

Therefore, it holds that $\mathbf{F}(\alpha)^{-1}\mathbf{A} = \mathbf{X}^{(\alpha)}\Lambda^{(\alpha)}\mathbf{X}^{(\alpha)^{-1}}$, *where* $\mathbf{X}^{(\alpha)} = (\mathbf{x}_1^{(\alpha)}, \mathbf{x}_2^{(\alpha)}, \ldots, \mathbf{x}_n^{(\alpha)}) \in \mathbb{R}^{n\times n}$ *and* $\Lambda^{(\alpha)} = \mathrm{diag}(\lambda_1^{(\alpha)}, \lambda_2^{(\alpha)}, \ldots, \lambda_n^{(\alpha)}) \in \mathbb{C}^{n\times n}$, *with* $\kappa_2(\mathbf{X}^{(\alpha)}) = \sqrt{\kappa_2(\alpha\mathbf{W} + \mathbf{T})}$.

Proof. Define matrices

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

and

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

Then it holds that

$$\mathbf{Z}^{(\alpha)} = \mathbf{Q}^{(\alpha)}\Xi^{(\alpha)}\mathbf{Q}^{(\alpha)^T}.$$

Here and in the sequel, $(\cdot)^T$ denotes the transpose of a real matrix. By straightforward computations we have

$$\begin{aligned}
\mathbf{F}(\alpha)^{-1}\mathbf{A} &= \frac{2\alpha}{(\alpha+1)(1+\imath)} \cdot (\alpha\mathbf{W} + \mathbf{T})^{-1}(\mathbf{W} + \imath\,\mathbf{T}) \\
&= \frac{2\alpha}{(\alpha+1)(1+\imath)(\alpha-\imath)} \cdot (\alpha\mathbf{W} + \mathbf{T})^{-1}[(\alpha\mathbf{W} + \mathbf{T}) - \imath(\mathbf{W} - \alpha\mathbf{T})] \\
&= \frac{2\alpha}{(\alpha+1)[(\alpha+1) + \imath(\alpha-1)]} \cdot [\mathbf{I} - \imath(\alpha\mathbf{W} + \mathbf{T})^{-1}(\mathbf{W} - \alpha\mathbf{T})] \\
&= \frac{2\alpha}{(\alpha+1)[(\alpha+1) + \imath(\alpha-1)]} \cdot (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}(\mathbf{I} - \imath\,\mathbf{Z}^{(\alpha)})(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}} \\
&= \frac{2\alpha}{(\alpha+1)[(\alpha+1) + \imath(\alpha-1)]} \cdot (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{Q}^{(\alpha)}(\mathbf{I} - \imath\,\Xi^{(\alpha)})\mathbf{Q}^{(\alpha)^T}(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}} \\
&= \mathbf{X}^{(\alpha)}\Lambda^{(\alpha)}\mathbf{X}^{(\alpha)^{-1}}.
\end{aligned}$$

Hence, the eigenvalues of the matrix $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ are given by

$$\lambda_j^{(\alpha)} = \frac{\alpha[(\alpha+1) - \imath(\alpha-1)](1 - \imath\mu_j^{(\alpha)})}{(\alpha+1)(\alpha^2+1)}, \qquad j = 1, 2, \ldots, n,$$

and the corresponding eigenvectors are given by $\mathbf{x}_j^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{q}_j^{(\alpha)}$, $j = 1, 2, \ldots, n$.

Besides, as $\mathbf{X}^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{Q}^{(\alpha)}$ and $\mathbf{Q}^{(\alpha)} \in \mathbb{R}^{n\times n}$ is orthogonal, we can obtain

$$\|\mathbf{X}^{(\alpha)}\|_2 = \|(\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\mathbf{Q}^{(\alpha)}\|_2 = \|(\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}\|_2 = \|(\alpha\mathbf{W} + \mathbf{T})^{-1}\|_2^{\frac{1}{2}}$$

and

$$\|\mathbf{X}^{(\alpha)^{-1}}\|_2 = \|\mathbf{Q}^{(\alpha)^T}(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}}\|_2 = \|(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}}\|_2 = \|\alpha\mathbf{W} + \mathbf{T}\|_2^{\frac{1}{2}}.$$

It then follows that

$$\kappa_2(\mathbf{X}^{(\alpha)}) = \|\mathbf{X}^{(\alpha)}\|_2\|\mathbf{X}^{(\alpha)^{-1}}\|_2 = \|(\alpha\mathbf{W} + \mathbf{T})^{-1}\|_2^{\frac{1}{2}}\|\alpha\mathbf{W} + \mathbf{T}\|_2^{\frac{1}{2}} = \sqrt{\kappa_2(\alpha\mathbf{W} + \mathbf{T})}.$$

□

Remark 3.1. *The previous result requires some comments. Because of the non-uniqueness of the eigenvectors, the condition number $\kappa_2(\mathbf{X}^{(\alpha)})$ of the eigenvector matrix is also not uniquely defined. One possibility is to replace it with the infimum over all possible choices of the eigenvector matrix $\mathbf{X}^{(\alpha)}$. However, this quantity is not easily computable. As an approximation, we will use instead the condition number of the matrix formed with the normalized eigenvectors returned by the* `eig` *function in Matlab. When the eigenvectors are normalized in the 2-norm, $\mathbf{X}^{(\alpha)}$ is replaced by*

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

with

$$\mathbf{D}^{(\alpha)} = \operatorname{diag}\left(\|\mathbf{x}_1^{(\alpha)}\|_2, \|\mathbf{x}_2^{(\alpha)}\|_2, \ldots, \|\mathbf{x}_n^{(\alpha)}\|_2\right),$$

leading to

$$\mathbf{D}^{(\alpha)} = \left(\operatorname{diag}(\mathbf{q}_1^{(\alpha)^T}(\alpha\mathbf{W}+\mathbf{T})^{-1}\mathbf{q}_1^{(\alpha)}, \mathbf{q}_2^{(\alpha)^T}(\alpha\mathbf{W}+\mathbf{T})^{-1}\mathbf{q}_2^{(\alpha)}, \ldots, \mathbf{q}_n^{(\alpha)^T}(\alpha\mathbf{W}+\mathbf{T})^{-1}\mathbf{q}_n^{(\alpha)})\right)^{\frac{1}{2}}$$

and

$$\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = \kappa_2(\mathbf{D}^{(\alpha)}\mathbf{Q}^{(\alpha)^T}(\alpha\mathbf{W}+\mathbf{T})^{1/2}).$$

In the special case when the coefficient matrix $\mathbf{A} = \mathbf{W} + \imath\,\mathbf{T} \in \mathbb{C}^{n\times n}$ is normal, we can easily see that the PMHSS-preconditioned matrix $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ is also normal. In this case the condition number of the normalized eigenvector matrix $\widetilde{\mathbf{X}}^{(\alpha)}$ is of course exactly equal to one. This property is formally stated in the following theorem.

Theorem 3.3. *Let the conditions of Theorem 3.2 be satisfied, and the eigenvector matrix $\mathbf{X}^{(\alpha)}$ be normalized as in Remark 3.1 with $\widetilde{\mathbf{X}}^{(\alpha)}$ being the normalized matrix. Assume that the matrix $\mathbf{A} = \mathbf{W} + \imath\,\mathbf{T} \in \mathbb{C}^{n\times n}$ is normal. Then it holds that $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = 1$. Moreover, the orthogonal eigenvectors $\mathbf{q}_1^{(\alpha)}, \mathbf{q}_2^{(\alpha)}, \ldots, \mathbf{q}_n^{(\alpha)}$ of the matrix $\mathbf{Z}^{(\alpha)} \in \mathbb{R}^{n\times n}$ are independent of the positive parameter α.*

Proof. Because $\mathbf{A} = \mathbf{W} + \imath\,\mathbf{T} \in \mathbb{C}^{n\times n}$ is normal, the matrices $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n\times n}$ commute, i.e., it holds that $\mathbf{WT} = \mathbf{TW}$. Hence, there exists an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{W} = \mathbf{Q}\Omega\mathbf{Q}^T \quad \text{and} \quad \mathbf{T} = \mathbf{Q}\Gamma\mathbf{Q}^T,$$

where

$$\Omega = \operatorname{diag}(\omega_1, \omega_2, \ldots, \omega_n) \quad \text{and} \quad \Gamma = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n)$$

are diagonal matrices with $\omega_j, \gamma_j \geq 0$, $j = 1, 2, \ldots, n$. It follows that

$$\alpha\mathbf{W} + \mathbf{T} = \mathbf{Q}(\alpha\Omega + \Gamma)\mathbf{Q}^T \quad \text{and} \quad \mathbf{W} - \alpha\mathbf{T} = \mathbf{Q}(\Omega - \alpha\Gamma)\mathbf{Q}^T.$$

As

$$(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}} = \mathbf{Q}(\alpha\Omega + \Gamma)^{\frac{1}{2}}\mathbf{Q}^T,$$

we obtain

$$\mathbf{Z}^{(\alpha)} = (\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}}(\mathbf{W} - \alpha\mathbf{T})(\alpha\mathbf{W} + \mathbf{T})^{-\frac{1}{2}} = \mathbf{Q}\Xi^{(\alpha)}\mathbf{Q}^T,$$

with

$$\Xi^{(\alpha)} = (\alpha\Omega + \Gamma)^{-1}(\Omega - \alpha\Gamma).$$

Therefore, the eigenvectors of the matrix $\mathbf{Z}^{(\alpha)}$ are given by the columns of the orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$, say, $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$, which are independent of the positive parameter α.

In addition, by straightforward computations we find

$$\mathbf{q}_j^T(\alpha\mathbf{W} + \mathbf{T})^{-1}\mathbf{q}_j = \mathbf{q}_j^T\mathbf{Q}(\alpha\Omega + \Gamma)^{-1}\mathbf{Q}^T\mathbf{q}_j = \mathbf{e}_j^T(\alpha\Omega + \Gamma)^{-1}\mathbf{e}_j = (\alpha\omega_j + \gamma_j)^{-1},$$

where $\mathbf{e}_j$ denotes the j-th unit vector in $\mathbb{R}^n$. Therefore, it holds that $\mathbf{D}^{(\alpha)} = (\alpha\Omega + \Gamma)^{-\frac{1}{2}}$ and

$$(\mathbf{D}^{(\alpha)}\mathbf{Q}^T(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}})(\mathbf{D}^{(\alpha)}\mathbf{Q}^T(\alpha\mathbf{W} + \mathbf{T})^{\frac{1}{2}})^T = \mathbf{D}^{(\alpha)}\mathbf{Q}^T(\alpha\mathbf{W} + \mathbf{T})\mathbf{Q}\mathbf{D}^{(\alpha)} = \mathbf{I},$$

which immediately results in $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = 1$. □

Remark 3.2. *If $\alpha = 1$, then Theorem 3.1(ii) leads to $\sigma(1) \leq \frac{\sqrt{2}}{2}$. This shows that when*

$$\mathbf{F} := (1 + \imath)(\mathbf{W} + \mathbf{T})$$

is used to precondition the matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, the eigenvalues of the preconditioned matrix $\mathbf{F}^{-1}\mathbf{A}$ are clustered within the complex disk centered at 1 with radius $\frac{\sqrt{2}}{2}$. Moreover, Theorem 3.2 indicates that the matrix $\mathbf{F}^{-1}\mathbf{A}$ is diagonalizable, with the matrix $\mathbf{X}^{(1)}$, formed by its eigenvectors, satisfying $\kappa_2(\mathbf{X}^{(1)}) = \sqrt{\kappa_2(\mathbf{W} + \mathbf{T})}$. Hence, the preconditioned Krylov subspace iteration methods, when employed to solve the complex symmetric linear system (1.1)-(1.2), can be expected to converge rapidly, at least when $\sqrt{\kappa_2(\mathbf{W} + \mathbf{T})}$ is not too large. As the previous theorem shows, this is guaranteed in the normal case.

4 Numerical Results

In this section we use three test problems from [1, 9, 4] to assess the feasibility and effectiveness of the PMHSS iteration method in terms of both iteration count (denoted as **IT**) and computing time (in seconds, denoted as **CPU**), when it is employed either as a solver or as a preconditioner for solving the system of linear equations (1.1)-(1.2). Besides comparing the efficiency of the PMHSS and the MHSS iteration methods, we also examine their numerical behavior as preconditioners for the (full) GMRES method and its restarted variants, say, GMRES(#); see [11].

In our implementations, the initial guess is chosen to be $\mathbf{x}^{(0)} = 0$ and the iteration is terminated once the current iterate $\mathbf{x}^{(k)}$ satisfies

$$\frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}\|_2}{\|\mathbf{b}\|_2} \leq 10^{-6}.$$

To accelerate the convergence rates of GMRES(#) and GMRES, we adopt the MHSS preconditioner defined by

$$\mathbf{F}(\mathbf{I}; \alpha) = \frac{1 + \imath}{2\alpha}(\alpha\mathbf{I} + \mathbf{W})(\alpha\mathbf{I} + \mathbf{T})$$

and the PMHSS preconditioner defined by

$$\mathbf{F}(\alpha) = \frac{(\alpha+1)(1+\imath)}{2\alpha}(\alpha\mathbf{W}+\mathbf{T}),$$

respectively; see (2.5) and (2.7).

In both MHSS and PMHSS iteration methods, the two half-steps comprising each iteration are computed exactly by the sparse Cholesky factorization incorporated with the `symamd.m` ordering algorithm. This technique is equally applied to the actions of the MHSS and the PMHSS preconditioners $\mathbf{F}(\mathbf{I};\alpha)$ and $\mathbf{F}(\alpha)$, respectively.

The iteration parameters used in both MHSS and PMHSS iteration methods as well as the corresponding MHSS and PMHSS preconditioners are the experimentally found ones, which minimize the numbers of iteration steps; see Tables 7 and 8. Moreover, if these optimal iteration parameters form intervals, then they are further optimized according to the lest computing times; see Tables 1-6. We remark that when the right endpoints of the optimal parameter intervals obtained from minimizing the iteration steps are larger than 1000.0, we just cut off and set them as 1000.0 and, by noticing that all left endpoints of such intervals are less than 3.65, we then search the optimal iteration parameters by minimizing the computing times from the left endpoints of the intervals to 10.0. The optimal iteration parameters determined in such a manner are denoted as $\alpha_{\exp}$.

In addition, all codes were run in MATLAB (version R2009a) in double precision and all experiments were performed on a personal computer with 2.96GHz central processing unit (Intel(R) Core(TM)2 Duo CPU L9400), 1.86G memory and Windows operating system.

Example 4.1. *(See [1, 4]) The system of linear equations (1.1)-(1.2) is of the form*

$$\left[\left(\mathbf{K}+\frac{3-\sqrt{3}}{\tau}\mathbf{I}\right)+\imath\left(\mathbf{K}+\frac{3+\sqrt{3}}{\tau}\mathbf{I}\right)\right]\mathbf{x}=\mathbf{b}, \tag{4.1}$$

where τ is the time step-size and $\mathbf{K}$ is the five-point centered difference matrix approximating the negative Laplacian operator $L=-\Delta$ with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0,1]\times[0,1]$ with the mesh-size $h=\frac{1}{m+1}$. The matrix $\mathbf{K}\in\mathbb{R}^{n\times n}$ possesses the tensor-product form $\mathbf{K}=\mathbf{I}\otimes\mathbf{B}_m+\mathbf{B}_m\otimes\mathbf{I}$, with $\mathbf{B}_m=h^{-2}\cdot\mathrm{tridiag}(-1,2,-1)\in\mathbb{R}^{m\times m}$. Hence, $\mathbf{K}$ is an $n\times n$ block-tridiagonal matrix, with $n=m^2$. We take

$$\mathbf{W}=\mathbf{K}+\frac{3-\sqrt{3}}{\tau}\mathbf{I}\quad \textit{and}\quad \mathbf{T}=\mathbf{K}+\frac{3+\sqrt{3}}{\tau}\mathbf{I},$$

and the right-hand side vector $\mathbf{b}$ with its jth entry $[\mathbf{b}]_j$ being given by

$$[\mathbf{b}]_j=\frac{(1-\imath)j}{\tau(j+1)^2},\qquad j=1,2,\ldots,n.$$

Furthermore, we normalize coefficient matrix and right-hand side by multiplying both by h^2.

In our tests we take $\tau=h$. Numerical results for Example 4.1 are listed in Tables 1 and 2. In Table 1 we show IT and CPU for MHSS, PMHSS, GMRES and GMRES(#) methods, while in Table 2 we show results for MHSS- and PMHSS-preconditioned GMRES and GMRES(#) methods, respectively.

From Table 1 we see that the iteration counts with the MHSS, GMRES and GMRES(#) methods grow rapidly with problem size, while that of PMHSS method remains constant. In other words the PMHSS iteration method shows h-independent convergence, unlike the other schemes. Moreover, PMHSS considerably outperforms MHSS, GMRES and GMRES(#), both in terms of iteration counts and in terms of CPU time.

In Table 2 we report results for GMRES and GMRES(#) preconditioned with MHSS and PMHSS. From these results we observe that when used as a preconditioner, PMHSS performs much better than MHSS in both iteration steps and CPU times, especially when the mesh-size h becomes small. While the number of iterations with the MHSS preconditioner increases with problem size, those for the PMHSS preconditioner is almost constant. Thus, the PMHSS-preconditioned GMRES and GMRES(#) methods show h-independent convergence property, whereas the MHSS-preconditioned GMRES and GMRES(#) methods do not; see Figure 1 (left). In particular, when we set the iteration parameter α to be 1, we see that the iteration counts for the PMHSS-preconditioned GMRES and GMRES(#) methods are almost identical to those obtained with the experimentally found optimal parameters $\alpha_{\exp}$. This shows 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.

Table 1: IT and CPU for HSS, MHSS, GMRES and GMRES(20) Methods for Example 4.1

Method	$m \times m$	16× 16	32× 32	64× 64	128×128	256×256
MHSS	$\alpha_{\exp}$	1.16	0.78	0.55	0.40	0.30
	IT	39	53	72	98	133
	CPU	0.012	0.067	0.577	4.327	34.022
PMHSS	$\alpha_{\exp}$	1.09	1.36	1.35	1.05	1.44
	IT	21	21	21	21	21
	CPU	0.008	0.035	0.240	1.099	6.188
GMRES	IT	34	53	81	112	155
	CPU	0.027	0.148	1.397	10.625	94.164
GMRES(20)	IT	39	62	91	136	214
	CPU	0.022	0.083	0.505	2.723	22.469

Example 4.2. *(See [9, 4]) The system of linear equations (1.1)-(1.2) is of the form*

$$[(-\omega^2\mathbf{M} + \mathbf{K}) + \imath\,(\omega\mathbf{C}_V + \mathbf{C}_H)]\mathbf{x} = \mathbf{b}, \tag{4.2}$$

where $\mathbf{M}$ *and* $\mathbf{K}$ *are the inertia and the stiffness matrices,* $\mathbf{C}_V$ *and* $\mathbf{C}_H$ *are the viscous and the hysteretic damping matrices, respectively, and* ω *is the driving circular frequency. We take* $\mathbf{C}_H = \mu\mathbf{K}$ *with* μ *a damping coefficient,* $\mathbf{M} = \mathbf{I}$, $\mathbf{C}_V = 10\mathbf{I}$, *and* $\mathbf{K}$ *the five-point centered difference matrix approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square* $[0,1] \times [0,1]$ *with the mesh-size* $h = \frac{1}{m+1}$. *The matrix* $\mathbf{K} \in \mathbb{R}^{n\times n}$ *possesses the tensor-product form* $\mathbf{K} = \mathbf{I} \otimes \mathbf{B}_m + \mathbf{B}_m \otimes \mathbf{I}$, *with* $\mathbf{B}_m = h^{-2} \cdot \mathrm{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}$. *Hence,* $\mathbf{K}$ *is an* $n \times n$ *block-tridiagonal matrix, with* $n = m^2$. *In addition, we set* $\omega = \pi$, $\mu = 0.02$, *and the right-hand side vector* $\mathbf{b}$ *to be*

Table 2: IT and CPU for Preconditioned GMRES and GMRES(10) for Example 4.1

Method	Prec	$m \times m$	16× 16	32× 32	64× 64	128×128	256×256
GMRES	MHSS	$\alpha_{\exp}$	1.65	1.06	0.74	0.57	0.40
		IT	9	12	15	19	22
		CPU	0.006	0.029	0.206	1.586	10.446
	PMHSS	$\alpha_{\exp}$	0.52	1.82	1.48	1.20	1.60
		IT	6	7	8	8	8
		CPU	0.003	0.010	0.062	0.354	2.045
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	6	7	8	8	8
		CPU	0.003	0.013	0.075	0.372	2.114
GMRES(10)	MHSS	$\alpha_{\exp}$	2.03	0.94	0.79	0.49	0.36
		IT	9	12	15	19	22
		CPU	0.006	0.029	0.212	1.537	10.148
	PMHSS	$\alpha_{\exp}$	0.56	1.64	3.67	0.86	1.19
		IT	6	7	8	8	8
		CPU	0.003	0.010	0.060	0.347	2.019
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	6	7	8	8	8
		CPU	0.003	0.012	0.074	0.377	2.149

$\mathbf{b} = (1 + \imath)\mathbf{A1}$, *with* $\mathbf{1}$ *being the vector of all entries equal to* 1. *As before, we normalize the system by multiplying both sides through by* h^2.

Numerical results for Example 4.2 are listed in Tables 3 and 4. In Table 3 we show IT and CPU for MHSS, PMHSS, GMRES and GMRES(#) methods, while in Table 4 we show results for MHSS- and PMHSS-preconditioned GMRES and GMRES(#) methods, respectively.

From Table 3 we see that the iteration counts for MHSS, GMRES and GMRES(#) increase rapidly with problem size, while those for PMHSS method are essentially constant after a slight increases when going from $m = 16$ to $m = 64$. Therefore, the PMHSS iteration method shows h-independent convergence, whereas the other iteration methods do not. Moreover, PMHSS considerably outperforms MHSS, GMRES and GMRES(#) both in terms of iteration counts and in terms of CPU times.

In Table 4 we report results for GMRES and GMRES(#) preconditioned with MHSS and PMHSS. From these results we observe that when used as a preconditioner, PMHSS performs much better than MHSS in both iteration counts and CPU times, especially when the mesh-size h becomes small. While the iteration counts for the MHSS preconditioner grow with problem size, those for the PMHSS preconditioner remain nearly constants. Again, the PMHSS-preconditioned GMRES and GMRES(#) methods show h-independent convergence property, whereas the convergence rates for MHSS-preconditioned GMRES and GMRES(#) are h-dependent; see Figure 1 (middle). As before, using $\alpha = 1$ gives nearly optimal results and this value should be used in practice.

Table 3: IT and CPU for HSS, MHSS, GMRES and GMRES(20) Methods for Example 4.2

Method	$m \times m$	16× 16	32× 32	64× 64	128× 128	256× 256
MHSS	$\alpha_{\exp}$	0.21	0.09	0.04	0.02	0.01
	IT	34	37	50	81	139
	CPU	0.011	0.050	0.434	3.903	35.421
PMHSS	$\alpha_{\exp}$	0.68	0.98	0.93	1.10	0.97
	IT	34	37	38	38	38
	CPU	0.013	0.052	0.449	1.755	10.451
GMRES	IT	26	52	102	196	379
	CPU	0.018	0.144	2.290	31.617	932.419
GMRES(20)	IT	39	128	412	1297	4369
	CPU	0.032	0.149	1.950	27.477	467.754

Table 4: IT and CPU for Preconditioned GMRES and GMRES(10) for Example 4.2

Method	Prec	$m \times m$	16× 16	32× 32	64× 64	128×128	256×256
GMRES	MHSS	$\alpha_{\exp}$	0.28	0.17	0.05	0.03	0.03
		IT	8	10	13	18	25
		CPU	0.006	0.028	0.217	1.553	14.831
	PMHSS	$\alpha_{\exp}$	12.09	8.90	1.46	6.95	7.23
		IT	6	7	7	7	7
		CPU	0.003	0.011	0.057	0.322	1.818
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	7	7	7	7	7
		CPU	0.004	0.013	0.059	0.336	1.796
GMRES(10)	MHSS	$\alpha_{\exp}$	0.29	0.19	0.08	0.02	0.01
		IT	8	10	13	21	35
		CPU	0.006	0.028	0.219	1.845	19.147
	PMHSS	$\alpha_{\exp}$	8.90	4.65	2.45	7.15	6.23
		IT	6	7	7	7	7
		CPU	0.003	0.011	0.058	0.323	1.771
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	7	7	7	7	7
		CPU	0.004	0.012	0.067	0.355	1.813

Example 4.3. *(See [4]) The system of linear equations (1.1)-(1.2) is of the form* $(\mathbf{W}+\imath\,\mathbf{T})\mathbf{x}=\mathbf{b}$*, with*

$$\mathbf{T}=\mathbf{I}\otimes\mathbf{B}+\mathbf{B}\otimes\mathbf{I}\quad \textit{and}\quad \mathbf{W}=10(\mathbf{I}\otimes\mathbf{B}_c+\mathbf{B}_c\otimes\mathbf{I})+9(\mathbf{e}_1\mathbf{e}_m^T+\mathbf{e}_m\mathbf{e}_1^T)\otimes\mathbf{I},$$

where $\mathbf{B}=\text{tridiag}(-1,2,-1)\in\mathbb{R}^{m\times m}$*,* $\mathbf{B}_c=\mathbf{B}-\mathbf{e}_1\mathbf{e}_m^T-\mathbf{e}_m\mathbf{e}_1^T\in\mathbb{R}^{m\times m}$*, and* $\mathbf{e}_1$ *and* $\mathbf{e}_m$ *are the first and the* m*th unit basis vectors in* $\mathbb{R}^m$*, respectively. We take the right-hand side vector* $\mathbf{b}$ *to be of the form* $\mathbf{b}=(1+\imath)\mathbf{A1}$*, with* $\mathbf{1}$ *being the vector of all entries equal to* 1*.*

Numerical results for Example 4.3 are listed in Tables 5 and 6. In Table 5 we show IT and CPU for MHSS, PMHSS, GMRES and GMRES(#) methods, while in Table 6 we show results for MHSS- and PMHSS-preconditioned GMRES and GMRES(#) methods, respectively.

From Table 5 we see that the iteration counts for MHSS, GMRES and GMRES(#) increase rapidly with problem size, while the rate of convergence with PMHSS is essentially constant. PMHSS vastly outperforms MHSS, GMRES and GMRES(#) in terms of both iteration counts and CPU times.

In Table 6 we report results for GMRES and GMRES(#) preconditioned with MHSS and PMHSS. From these results we observe that when used as a preconditioner, PMHSS performs considerably better than MHSS in both iteration counts and CPU times, especially when the mesh-size h becomes small. Similar observations to the ones made for the other two examples apply.

Table 5: IT and CPU for HSS, MHSS, GMRES and GMRES(20) Methods for Example 4.3

Method	$m\times m$	16× 16	32× 32	64× 64	128× 128	256× 256
MHSS	α_{exp}	1.79	1.05	0.55	0.27	0.14
	IT	51	75	128	241	458
	CPU	0.026	0.182	1.965	22.959	221.55
PMHSS	α_{exp}	0.61	0.42	0.57	0.78	0.73
	IT	30	30	30	30	30
	CPU	0.018	0.095	0.623	3.519	19.568
GMRES	IT	35	70	138	263	–
	CPU	0.039	0.272	4.308	59.031	–
GMRES(20)	IT	65	184	414	1295	2840
	CPU	0.045	0.239	2.094	27.761	305.224

In Table 7 we list the experimentally found optimal parameters α_{exp} for the MHSS and PMHSS iterations, and in Table 8 we list those for the MHSS- and PMHSS-preconditioned GMRES and GMRES(#) methods. These optimal parameters are obtained by minimizing the numbers of iterations with respect to each test example and each spatial mesh-size.

From Table 7 we observe that for both Examples 4.1 and 4.2 the optimal parameters of the PMHSS iteration method forms intervals including 1.0 as an interior point, and for Example 4.3 they form intervals of large widths. This implies that the PMHSS iteration method is insensitive to the iteration parameter α and, in actual implementations, for Examples 4.1 and 4.2 we can always take $\alpha=1.0$ to obtain essentially optimal convergence rates. The situation for the MHSS

Table 6: IT and CPU for Preconditioned GMRES and GMRES(10) for Example 4.3

Method	Prec	$m \times m$	16× 16	32× 32	64× 64	128×128	256×256
GMRES	MHSS	$\alpha_{\exp}$	4.16	2.52	1.25	0.61	0.34
		IT	10	14	19	27	38
		CPU	0.008	0.045	0.361	3.304	28.250
	PMHSS	$\alpha_{\exp}$	4.37	7.06	2.71	4.84	7.22
		IT	5	6	7	9	11
		CPU	0.003	0.012	0.078	0.657	4.024
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	5	6	8	9	11
		CPU	0.003	0.018	0.097	0.738	4.151
GMRES(10)	MHSS	$\alpha_{\exp}$	6.13	1.93	1.05	0.79	0.28
		IT	10	14	24	39	51
		CPU	0.008	0.045	0.483	4.780	33.806
	PMHSS	$\alpha_{\exp}$	6.62	6.82	8.67	1.34	2.06
		IT	5	6	7	9	11
		CPU	0.003	0.012	0.079	0.638	4.241
	PMHSS	α	1.00	1.00	1.00	1.00	1.00
		IT	5	6	8	9	11
		CPU	0.003	0.016	0.100	0.759	4.300

iteration method is quite different. Its optimal parameter forms very narrow intervals with respect to different spatial mesh-sizes for Examples 4.1 and 4.3 with the exception $m = 256$ for Examples 4.3, and it is a single point with respect to almost all spatial mesh-sizes for Example 4.2 except for $m = 16$. This shows that the MHSS iteration method is quite sensitive to the iteration parameter α. Note that the optimal parameters of the MHSS iteration method are always either less than or larger than 1.0, and just setting $\alpha = 1.0$ in the MHSS iteration method will not produce optimal results; see Figure 2.

From Table 8 we see that similar conclusions hold relative to the optimal parameters for the PMHSS-preconditioned GMRES and GMRES(#) iteration methods. Note that for Examples 4.2 and 4.3 the intervals containing optimal values of α are very wide. This implies that the PMHSS preconditioner is not sensitive to the iteration parameter α and PMHSS-preconditioned GMRES and GMRES(#) are, roughly speaking, α-independent iteration methods for almost all cases of the spatial mesh-sizes and, in actual implementations, we can always take $\alpha = 1.0$ to obtain essentially optimal results. On the other hand, the MHSS preconditioner is relatively sensitive to the choice of α, and just setting $\alpha = 1$ will not perform well in practice.

Note that in the three examples the matrices $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n\times n}$ are symmetric positive definite. Moreover, for Examples 4.1 and 4.2 they are simultaneously orthogonally similar to diagonal matrices and therefore the coefficient matrices $\mathbf{A} \in \mathbb{C}^{n\times n}$ are normal. From Theorem 3.3 we see that the Euclidean condition numbers of the normalized matrices $\widetilde{\mathbf{X}}^{(\alpha)}$ from the eigenvector matrices $\mathbf{X}^{(\alpha)}$ of the PMHSS-preconditioned matrices $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ are equal to 1, i.e., $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)}) = 1$; see Remark 3.1 and Theorem 3.2. Recall that in this case $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ in both Examples 4.1 and 4.2

Table 7: The Experimental Optimal Parameters $\alpha_{\exp}$ for MHSS and PMHSS Iteration Methods by Minimizing Iteration Steps

Example	Method	Grid				
		16×16	32×32	64×64	128×128	256×256
No. 4.1	MHSS	[1.11, 1.16]	[0.78, 0.81]	[0.55, 0.57]	[0.40, 0.41]	[0.29, 0.30]
	PMHSS	[0.97, 1.55]	[0.94, 1.51]	[0.87, 1.49]	[0.80, 1.48]	[0.75, 1.47]
No. 4.2	MHSS	[0.17, 0.27]	0.09	0.04	0.02	0.01
	PMHSS	[0.55, 1.04]	[0.64, 1.16]	[0.68, 1.18]	[0.74, 1.12]	[0.77, 1.10]
No. 4.3	MHSS	[1.69, 1.88]	[1.03, 1.05]	[0.55, 0.57]	[0.27, 0.28]	0.14
	PMHSS	[0.24, 0.78]	[0.31, 0.78]	[0.40, 0.79]	[0.52, 0.80]	[0.73, 0.75]

Table 8: The Experimental Optimal Parameters $\alpha_{\exp}$ for Preconditioned GMRES and GMRES(10) by Minimizing Iteration Steps

Example	Method	Prec	Grid				
			16×16	32×32	64×64	128×128	256×256
No. 4.1	GMRES	MHSS	[1.40, 2.18]	[0.87, 1.49]	[0.69, 0.88]	[0.40, 0.69]	[0.35, 0.40]
		PMHSS	[0.13, 2.67]	[0.01, 3.13]	[0.01, 4.02]	[0.01, 2.10]	[0.01, 1.81]
	GMRES(10)	MHSS	[1.40, 2.18]	[0.90, 1.46]	[0.72, 0.80]	[0.43, 0.64]	[0.35, 0.40]
		PMHSS	[0.13, 2.67]	[0.01, 3.13]	[0.01, 4.02]	[0.01, 2.10]	[0.01, 1.81]
No. 4.2	GMRES	MHSS	[0.09, 0.48]	[0.08, 0.24]	[0.05, 0.11]	[0.03, 0.06]	[0.02, 0.03]
		PMHSS	[3.65, 1000]	[0.68, 1000]	[0.72, 1000]	[0.73, 1000]	[0.73, 1000]
	GMRES(10)	MHSS	[0.09, 0.48]	[0.08, 0.24]	[0.06, 0.10]	0.02	0.01
		PMHSS	[3.65 1000]	[0.68, 1000]	[0.72, 1000]	[0.73 1000]	[0.73, 1000]
No. 4.3	GMRES	MHSS	[3.28, 6.71]	[1.77, 3.95]	[1.20, 1.79]	[0.60, 0.94]	[0.31, 0.42]
		PMHSS	[0.90, 1000]	[0.92, 1000]	[1.40, 1000]	[0.77, 1000]	[0.69, 1000]
	GMRES(10)	MHSS	[3.28, 6.71]	[1.88, 1.97]	[1.05, 1.12]	[0.79, 0.83]	[0.28, 0.30]
		PMHSS	[0.90, 1000]	[0.92, 1000]	[1.40, 1000]	[0.77, 1000]	[0.75, 30.75]

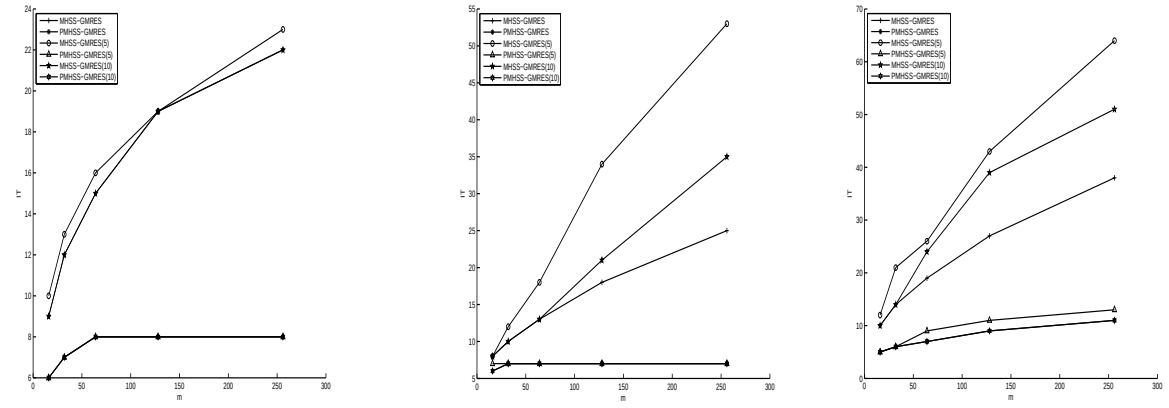

Figure 1: Pictures of IT versus m for MHSS- and PMHSS-preconditioned GMRES(5), GMRES(10) and GMRES methods with $\alpha = \alpha_{\exp}$; left: Example 4.1, middle: Example 4.2, and right: Example 4.3.

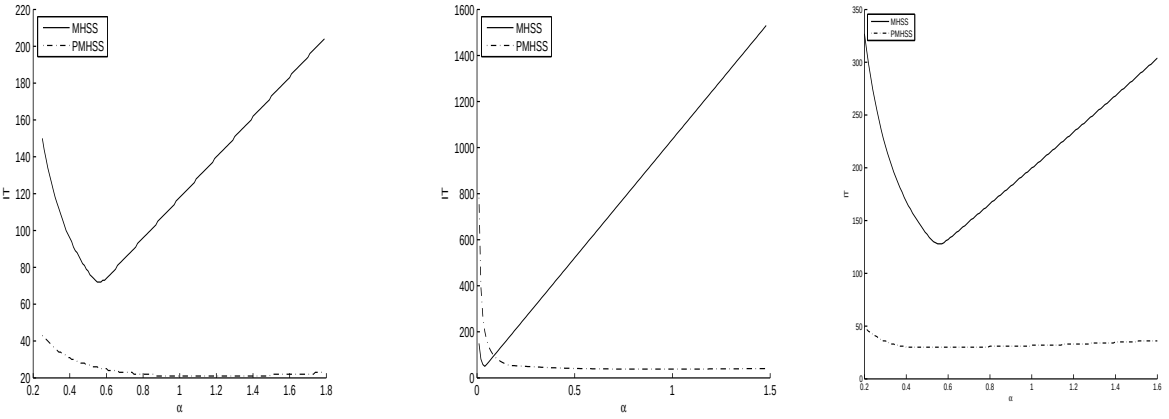

Figure 2: Pictures of IT versus α for MHSS and PMHSS iteration methods with $m = 64$; left: Example 4.1, middle: Example 4.2, and right: Example 4.3.

are normal matrices. The situation for Example 4.3 is different, since the matrices $\mathbf{W}, \mathbf{T} \in \mathbb{R}^{n\times n}$ do not commute, so that $\mathbf{F}(\alpha)^{-1}\mathbf{A}$ is not a normal matrix. As a result, we cannot expect $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha)})$ to be 1, or even to remain bounded as the problem size increases. We found, however, that for mesh sizes $m = 16$, $m = 32$ and $m = 64$, the condition numbers for the experimentally optimal values of α are $\kappa_2(\widetilde{\mathbf{X}}^{(\alpha_{\exp})}) = 11.86$, 20.41 and 20.58, respectively. This suggests that for the optimal value of α the eigenvector condition numbers remain bounded as the mesh size increases.

5 Concluding Remarks

In this paper we have studied a preconditioned variant of the modified Hermitian and skew-Hermitian iteration for a class of complex symmetric systems. The PMHSS iteration method not

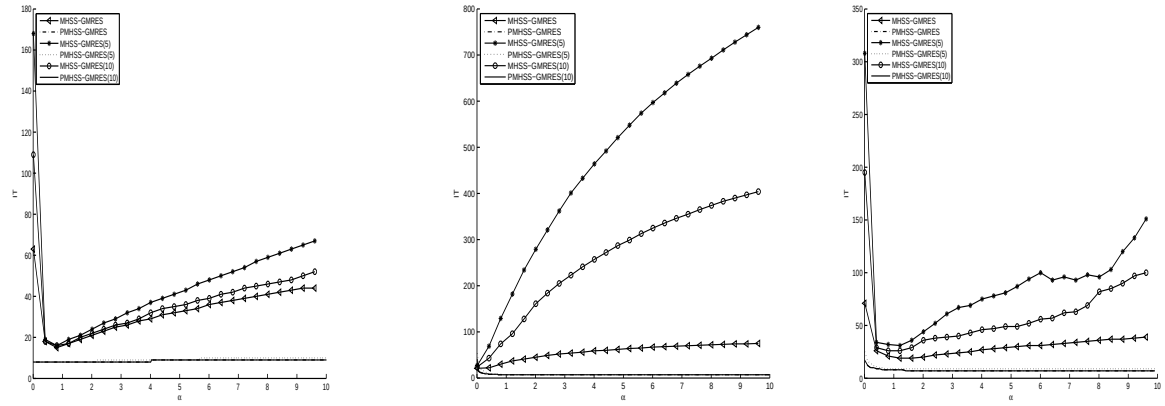

Figure 3: Pictures of IT versus α for MHSS- and PMHSS-preconditioned GMRES(5), GMRES(10) and GMRES methods with $\alpha = \alpha_{\exp}$; left: Example 4.1, middle: Example 4.2, and right: Example 4.3.

only presents a more general framework, but also yields much better theoretical and numerical properties than the MHSS iteration method. In particular, the PMHSS iteration results in asymptotically h-independent convergence rates when it is employed either as a solver or as a preconditioner.

Using PMHSS as a preconditioner for GMRES always results in faster solution times than using PMHSS as a stationary (fixed point) iteration. However, GMRES acceleration requires additional operations, such as inner products and orthogonalization steps, which may be difficult to implement efficiently on parallel architectures. Hence, it may be better to use PMHSS alone in some cases.

Our analysis shows that when $\mathbf{V} = \mathbf{W}$ we obtain PMHSS preconditioners for which the eigenvalues of the preconditioned matrices are clustered within complex disks centered at 1 with radii $\delta(\alpha) := \frac{\sqrt{\alpha^2+1}}{\alpha+1}$, and the condition numbers of the corresponding eigenvector matrices are equal to $\gamma(\alpha) := \sqrt{\kappa_2(\alpha\mathbf{W} + \mathbf{T})}$, where $\alpha > 0$ is the iteration parameter. Note that when $\alpha = 1$, it holds that $\delta(1) = \frac{\sqrt{2}}{2}$ and $\gamma(1) = \sqrt{\kappa_2(\mathbf{W} + \mathbf{T})}$. In actual implementations of the PMHSS preconditioner we can simply take the iteration parameter α to be 1, resulting in a parameter-free method.

In this paper we have limited ourselves to "exact" variants of the PMHSS iteration. In practice, using $\mathbf{V} = \mathbf{W}$ is likely to be too costly, especially when solving problems arising from the discretization of 3D partial differential equations. In this case, inexact solve should be used instead, that is, one should use $\mathbf{V} \approx \mathbf{W}$. For the problems considered in this paper, exact solves can be replaced with a single multigrid V-cycle or any other spectrally equivalent preconditioner. We leave the investigation of inexact variants of PMHSS for future work.

References

[1] O. Axelsson and A. Kucherov, Real valued iterative methods for solving complex symmetric linear systems, *Numer. Linear Algebra Appl.*, 7(2000), 197-218.

[2] Z.-Z. Bai, Construction and analysis of structured preconditioners for block two-by-two matrices, *J. Shanghai Univ. (English Edition)*, 8(2004), 397-405.

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

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

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

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

[9] M. Benzi and D. Bertaccini, Block preconditioning of real-valued iterative algorithms for complex linear systems, *IMA J. Numer. Anal.*, 28(2008), 598-618.

[10] D. Bertaccini, Efficient solvers for sequences of complex symmetric linear systems, *Electr. Trans. Numer. Anal.*, 18(2004), 49-64.

[11] K. Chen, Matrix Preconditioning Techniques and Applications, *Cambridge University Press*, Cambridge and New York, 2005.