Modified LPMHSS Method for a Class of Complex Symmetric Linear Systems

Shi-Liang Wu(✉) and Cui-Xia Li

School of Mathematics, Yunnan Normal University, Kunming 650500, Yunnan,
People's Republic of China

Abstract. In this paper, a modified LPMHSS (MLPMHSS) method is proposed to solve the problem of a class of complex symmetric linear systems with strong Hermitian parts. Theoretical analysis shows that the MLPMHSS method can converge to the unique solution of linear equations under appropriate conditions. Numerical experiments show that the method is effective.

Keywords: Complex symmetric linear system · Modified LPMHSS method · Convergence

1 Introduction

Consider the numerical solution of the linear system of the form

$$Ax = b, A \in \mathbb{C}^{n\times n}, x, b \in \mathbb{C}^n, \tag{1.1}$$

where

$$A = W + iT, \tag{1.2}$$

$i = \sqrt{-1}$ is the imaginary unit, and $W \in \mathbb{R}^{n\times n}$ is symmetric positive definite, and $T \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite. In the fields of structural dynamics, diffuse reflectance optical tomography, lattice quantum chromodynamics, eddy current problems, molecular dynamics, fluid dynamics and quantum chemistry, this form of complex symmetric linear system can be used to model these problems. You can view and reference [1–10] for more specific examples.

In order to solve the large-scale sparse complex symmetric linear system (1.1)–(1.2), based on the HSS method [11] and MHSS method [2], the preprocessing MHSS (PMHSS) method was ingeniously designed by Bai *et al.* [3] and the following work was done.

The PMHSS Method. Assum that $\alpha > 0$ and $x^{(0)} \in \mathbb{C}^n$ is a given arbitrary initial vector. For $k = 0, 1, 2, \ldots$ until the iterative sequences $\{x^{(k)}\}_{k=0}^{\infty}$ are convergent, calculate $x^{(k+1)}$ by

$$\begin{cases} (\alpha V + W)x^{(k+\frac{1}{2})} = (\alpha V - iT)x^{(k)} + b, \\ (\alpha V + T)x^{(k+1)} = (\alpha V + iW)x^{(k+\frac{1}{2})} - ib, \end{cases} \tag{1.3}$$

This research was supported by NSFC (No.11961082).

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 149–163, 2021.
https://doi.org/10.1007/978-3-030-72792-5_14

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

Under suitable conditions, the convergence of PMHSS method is given by Bai *et al.* [3]. Numerical experiments which is solving complex symmetric linear system (1.1)–(1.2) show that PMHSS method has better performance than MHSS method.

In order to further enhance the effect of this iterative method, a lopsided PMHSS (LPMHSS) method is introduced to solve the problem of complex symmetric linear system (1.1)–(1.2) in [12] and described as follows.

The LPMHSS Method. Assum that $\alpha > 0$ and $x^{(0)} \in \mathbb{C}^n$ is a given arbitrary initial vector. For $k = 0, 1, 2, \ldots$ until the iterative sequences $\{x^{(k)}\}_{k=0}^{\infty}$ are convergent, calculate $x^{(k+1)}$ by

$$\begin{cases} Wx^{(k+\frac{1}{2})} = -iTx^{(k)} + b, \\ (\alpha V + T)x^{(k+1)} = (\alpha V + iW)x^{(k+\frac{1}{2})} - ib, \end{cases} \tag{1.4}$$

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

The convergence theory of LPMHSS method is proposed in [12] by Li *et al.* and the theoretical optimal parameter with the minimum upper bound of spectral radius of LPMHSS iterative matrix is also derived. The numerical experiments in [12] show that the LPMHSS method is more efficient than the PMHSS method when it is used to solve complex symmetric linear systems with strong Hermitian parts (1.1)–(1.2).

In this paper, the iterative method for solving the complex symmetric linear system with strong Hermitian part (1.1)–(1.2) is further studied, in which the strong Hermitian part represents $\|W\| \gg \|T\|$ in (1.2) of a matrix norm. Based on the previous work in [13], a modified LPMHSS (MLPMHSS) method for a class of complex symmetric linear systems with strong Hermitian parts is proposed. Based on the upper bound of the spectral radius of MLPHSS iterative matrix, the convergence conditions of MLPHSS method are given, and the optimal parameters of MLPHSS method are given theoretically. The MLPHSS method is compared with LPMHSS method. Numerical experiments verify the effectiveness of the proposed method.

This paper is organized below. In Sect. 2, the MLPMHSS method is established and its convergence condition is discussed. Numerical examples are provided to verify the efficiency of the proposed method in Sect. 3. Finally, in Sect. 4, some conclusions are drawn to end this paper.

2 The MLPMHSS Method

To establish the MLPMHSS iteration method, the complex symmetric linear system (1.1)–(1.2) can be equivalently expressed as

$$Wx = -iTx + b \tag{2.1}$$

and

$$(\alpha V + W)x = (\alpha V - iT)x + b, \tag{2.2}$$

where $V \in \mathbb{R}^{n\times n}$ is a given symmetric positive definite matrix. Based on Eqs. (2.1) and (2.2), we can establish the following alternating splitting method for the complex symmetric linear system (1.1)–(1.2). Since the following alternating splitting method is similar to the LPMHSS method, which is called the MLPMHSS method.

The MLPMHSS Method. Assum that $\alpha > 0$ and $x^{(0)} \in \mathbb{C}^n$ is a given arbitrary initial vector. For $k = 0, 1, 2, \ldots$ until the iterative sequences $\{x^{(k)}\}_{k=0}^{\infty}$ are convergent, calculate $x^{(k+1)}$ by

$$\begin{cases} Wx^{(k+\frac{1}{2})} = -iTx^{(k)} + b, \\ (\alpha V + W)x^{(k+1)} = (\alpha V - iT)x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2.3}$$

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

To gain the convergence condition of the MLPMHSS method, we require Lemma 2.1.

Lemma 2.1 [11]**.** *Let $A = M_i - N_i$ $(i = 1, 2)$ be two splittings of $A \in \mathbb{C}^{n\times n}$, and let $x^{(0)} \in \mathbb{C}^n$ be a given initial vector. If the two-step iteration sequence $\{x^{(k)}\}$ is defined by*

$$\begin{cases} M_1x^{(k+\frac{1}{2})} = N_1x^{(k)} + b, \\ M_2x^{(k+1)} = N_2x^{(k+\frac{1}{2})} + b, \end{cases}$$

then

$$x^{(k+1)} = M_2^{-1}N_2M_1^{-1}N_1x^{(k)} + M_2^{-1}(I + N_2M_1^{-1})b,\ \ k = 0, 1, \ldots.$$

When $\rho(M_2^{-1}N_2M_1^{-1}N_1) < 1$, for all initial vectors $x^{(0)} \in \mathbb{C}^n$, the iteration sequence $\{x^{(k)}\}$ converges to the unique solution $x_* \in \mathbb{C}^n$ of the linear system (1.1).

Based on Lemma 2.1, from (2.3) we have

$$x^{(k+1)} = M_\alpha x^{(k)} + N_\alpha b, k = 0, 1, 2, \ldots, \tag{2.4}$$

where

$$M_\alpha = -i(\alpha V + W)^{-1}(\alpha V - iT)W^{-1}T$$

and

$$N_\alpha = (\alpha V + W)^{-1}(\alpha V + W - iT)W^{-1}b.$$

Theorem 2.1. *Let $A = W + iT \in \mathbb{C}^{n\times n}$ in (1.2) be normal, $V \in \mathbb{R}^{n\times n}$ be a symmetric positive definite and $\alpha > 0$. Then the iteration matrix M_α of MLPMHSS method is*

$$M_\alpha = -i(\alpha V + W)^{-1}(\alpha V - iT)W^{-1}T \tag{2.5}$$

and

$$\rho(M_\alpha) \le \delta(\alpha) = \frac{\mu_{\max}\sqrt{\alpha^2 + \mu_{\max}^2}}{\lambda_{\min}(\alpha + \lambda_{\min})}, \tag{2.6}$$

where $\mu_{\max}$ *is the largest eigenvalue of* $V^{-1}T$ *and* $\lambda_{\min}$ *is the smallest eigenvalue of* $V^{-1}W$*. Moreover, if*

$$\mu_{\max} \le \lambda_{\min},$$

then $\delta(\alpha) < 1$*, i.e., the MLPMHSS method* (2.3) *is convergent.*

Proof. Let

$$M_1 = W, N_1 = -iT, M_2 = \alpha V + W \text{ and } N_2 = \alpha V - iT.$$

Then the matrices W and $\alpha V + W$ are nonsingular on the base of the above assumptions, so (2.5) is valid.

Since matrix $A = W + iT$ is normal matrix, by simple computations, we obtain $WT = TW$. Further, $W^{-1}T = TW^{-1}$. Based on (2.5), the matrix M_α is equal to

$$M_\alpha = -i(\alpha V + W)^{-1}(\alpha V - iT)TW^{-1}.$$

Noting that V is symmetric positive definite, the matrices

$$\bar{W} = V^{-\frac{1}{2}}WV^{-\frac{1}{2}} \text{ and } \bar{T} = V^{-\frac{1}{2}}TV^{-\frac{1}{2}}$$

are well-defined. It follows that M_α is similar to

$$\bar{M}_\alpha = -i(\alpha I - i\bar{T})\bar{T}\bar{W}^{-1}(\alpha I + \bar{W})^{-1}.$$

Because $\bar{W}$ and $\bar{T}$, respectively, are symmetric positive definite and symmetric positive semidefinite, there exist unitary matrices V_1 and V_2 such that

$$\bar{W} = V_1\Lambda_1V_1^*,\ \bar{T} = V_2\Lambda_2V_2^*,$$

where $\Lambda_1 = \mathrm{diag}(\lambda_1, \lambda_1, \ldots, \lambda_n)$ and $\Lambda_2 = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)$ with λ_j and μ_j $(j = 1, 2, \ldots, n)$ being the eigenvalues of matrices $\bar{W}$ and $\bar{T}$, respectively.

Further, we obtain

$$\begin{aligned}
\rho(M_\alpha) =&\rho(\bar{M}_\alpha)\\
=&\rho(-i(\alpha I - i\bar{T})\bar{T}\bar{W}^{-1}(\alpha I + \bar{W})^{-1})\\
=&\rho((\alpha I - i\bar{T})\bar{T}\bar{W}^{-1}(\alpha I + \bar{W})^{-1})\\
\le&\|(\alpha I - i\bar{T})\bar{T}\bar{W}^{-1}(\alpha I + \bar{W})^{-1}\|_2\\
=&\|V_2(\alpha I - i\Lambda_2)\Lambda_2V_2^*V_1\Lambda_1^{-1}(\alpha I + \Lambda_1)^{-1}V_1^*\|_2\\
\le&\|(\alpha I - i\Lambda_2)\Lambda_2\|_2\|(\alpha I + \Lambda_1)^{-1}\Lambda_1^{-1}\|_2.
\end{aligned}$$

Since $\bar{W}$ is similar to $V^{-1}W$ and $\bar{T}$ is similar to $V^{-1}T$,

$$\|(\alpha I + \Lambda_1)^{-1}\Lambda_1^{-1}\|_2 = \max_{\lambda_j \in \lambda(V^{-1}W)} \frac{1}{\lambda_j(\alpha + \lambda_j)} = \frac{1}{\lambda_{\min}(\alpha + \lambda_{\min})} \tag{2.7}$$

and

$$\|(\alpha I - i\Lambda_2)\Lambda_2\|_2 = \max_{\mu_j \in \mu(V^{-1}T)} \mu_j\sqrt{\alpha^2 + \mu_j^2} = \mu_{\max}\sqrt{\alpha^2 + \mu_{\max}^2}, \tag{2.8}$$

where $\lambda(V^{-1}W)$ and $\mu(V^{-1}T)$ the spectrum of matrices $V^{-1}W$ and $V^{-1}T$, respectively.

Combining (2.7) with (2.8), the bound $\delta(\alpha)$ in (2.6) for $\rho(M_\alpha)$ can be obtained. If $\mu_{\max} \leq \lambda_{\min}$, then $\rho(M_\alpha) \leq \delta(\alpha) < 1$. ☐

Theorem 2.1 tells us that the rate of convergence of the MLPMHSS method is limited by $\delta(\alpha)$, which depends on the largest eigenvalue of $V^{-1}T$ and the smallest eigenvalue of $V^{-1}W$. If the largest eigenvalue of $V^{-1}T$ and the smallest eigenvalue of $V^{-1}W$ are obtained, then the specifical upper bound of $\rho(M_\alpha)$ can be provided. In the meantime, we gain the theoretical optimal parameter to minimize this upper bound, see Corollary 2.1.

Corollary 2.1. *Let the conditions of Theorem 2.1 be satisfied. Then the optimal value of the parameter* α *is* $\alpha^* = \frac{\mu_{\max}^2}{\lambda_{\min}}$ *and the minimum bound for* $\rho(M_\alpha)$ *is*

$$\delta(\alpha^*) = \frac{\mu_{\max}^2}{\lambda_{\min}\sqrt{\mu_{\max}^2 + \lambda_{\min}^2}}. \tag{2.9}$$

Proof. Differentiating the bound $\delta(\alpha)$ leads to

$$\delta'(\alpha) = \frac{\mu_{\max}}{\lambda_{\min}} \cdot \frac{\lambda_{\min}\alpha - \mu_{\max}^2}{\sqrt{\alpha^2 + \mu_{\max}^2}(\alpha + \lambda_{\min})^2}.$$

Setting $\delta'(\alpha) = 0$, we have

$$\alpha^* = \frac{\mu_{\max}^2}{\lambda_{\min}}, \tag{2.10}$$

Substituting (2.10) into (2.6) leads to (2.9). ☐

It is emphasized that in Corollary 2.1, the upper bound $\delta(\alpha)$ of $\rho(M_\alpha)$ achieves the minimum under the optimal parameter α^*, but this optimal parameter α^* does not minimize $\rho(M_\alpha)$ of M_α. Even so, based on Corollary 2.1, we can choose an effective parameter α for the MLPMHSS method. If we can choose the precondition matrix V such that $\mu_{\max} = \lambda_{\min}$, then $\rho(M_{\alpha^*}) \leq \delta(\alpha^*) = \frac{\sqrt{2}}{2}$.

For the convergence condition of the LPMHSS method, the following result was obtained in [12].

Theorem 2.2. *Let* $A = W + iT \in \mathbb{C}^{n\times n}$ *be defined in* (1.2)*,* $V \in \mathbb{R}^{n\times n}$ *be symmetric positive definite and* $\alpha > 0$*. Let* $\mu_{\max}$ *and* $\lambda_{\min}$ *be defined in Theorem 2.1. Then the iteration matrix* L_α *of the LPMHSS method is*

$$L_\alpha = -i(\alpha V + T)^{-1}(\alpha V + iW)W^{-1}T$$

and $\rho(L_\alpha) \leq \sigma(\alpha)$*, where*

$$\sigma(\alpha) = \frac{\mu_{\max}\sqrt{\alpha^2 + \lambda_{\max}^2}}{\lambda_{\min}(\alpha + \mu_{\max})}.$$

Moreover, it holds that

(i) If $\lambda_{\min} \geq \mu_{\max}$*, then* $\sigma(\alpha) < 1$ *for any* $\alpha > 0$*;*
(ii) If $\lambda_{\min} < \mu_{\max}$*, then* $\sigma(\alpha) < 1$ *if and only if*

$$\alpha < \frac{2\mu_{\max}^2 \lambda_{\min}}{\mu_{\max}^2 - \lambda_{\min}^2}.$$

Further, the optimal parameter α *is* $\alpha^\star = \frac{\lambda_{\min}^2}{\mu_{\max}}$ *and*

$$\sigma(\alpha^\star) = \frac{\mu_{\max}}{\sqrt{\mu_{\max}^2 + \lambda_{\min}^2}}.$$

Based on Corollary 2.1 and Theorem 2.2, Theorem 2.3 gives a comparison between the MLPMHSS method and the LPMHSS method.

Theorem 2.3. *Let* $A = W + iT \in \mathbb{C}^{n\times n}$ *in* (1.2) *be normal,* $V \in \mathbb{R}^{n\times n}$ *be symmetric positive definite and* $\alpha > 0$*, and let* $\mu_{\max}$ *and* $\lambda_{\min}$ *be defined in Theorem 2.1. Then the optimal upper bound* $\delta(\alpha^*)$ *of the spectral radius of the MLPMHSS iteration matrix and the optimal upper bound* $\sigma(\alpha^\star)$ *of the spectral radius of the LPMHSS iteration matrix satisfy*

$$\delta(\alpha^*) \leq \sigma(\alpha^\star) \text{ for } \mu_{\max} \leq \lambda_{\min}.$$

Proof. By calculation, we have

$$\frac{\mu_{\max}}{\sqrt{\mu_{\max}^2 + \lambda_{\min}^2}} \geq \frac{\mu_{\max}^2}{\lambda_{\min}\sqrt{\mu_{\max}^2 + \lambda_{\min}^2}}.$$

This completes the proof. □

Theorem 2.3 tells us that when both of them are optimal parameters, the optimal upper bound of MLPMHSS is less than or equal to the optimal upper bound of MLPMHSS, and the maximum eigenvalue of $V^{-1}T$ is less than or equal to the minimum eigenvalue $V^{-1}W$. It is worth noting that the result of Theorem 2.3 gives the comparison of the upper bounds of spectral radius of MLPMHSS and MLPMHSS iterative matrices, but does not give the comparison of spectral radii of MLPMHSS and MLPMHSS iterative matrices. Even so, Theorem 2.3 may imply that the convergence rate of LPMHSS is less than that of MLPMHSS when the complex symmetric linear system (1.1)–(1.2) takes their respective optimal parameters. The numerical results are confirmed in the next section.

3 Numerical Experiments

In this section, we test three problems to demonstrate the effectiveness of the MLPMHSS method for solving the complex symmetric linear system (1.1)–(1.2). In the implementations, $x^{(0)} = 0$ is chosen as the initial guess and

$$\frac{\|b - Ax^{(k)}\|_2}{\|b\|_2} \leq 10^{-6}$$

is chosen as the stopping criteria. All tests are performed in MATLAB 7.0 with machine precision 10^{-16}.

In order to verify the effectiveness of MLPMHSS, LPMHSS in [12] is better than PMHSS and MHSS, while PMHSS in [2,3] is better than MHSS and HSS. Based on the above characteristics, we compare MLPMHSS with LPMHSS.

In our calculation, based on the choice of V in LPMHSS [12], the precondition V used in LMPHSS and MLPMHSS methods is set to $V = W$. Since the coefficient matrices of all linear subsystems are symmetric and positive definite in each iteration step of LMPHSS and MLPMHSS methods, we use sparse Cholesky factorization to obtain the inverse of the corresponding matrix. Based on two aspects, we compare MLPMHSS method with LPMHSS method: one is iteration step (expressed by “IT”) and the other is CPU running time in seconds (represented by “CPU”).

3.1 Results for MLPMHSS Iteration

Example 3.1 ([7,12,14]). Let σ_1, σ_2 be two real coefficient functions. Then we consider the Helmholtz equation

$$-\Delta u + \sigma_1 u + i\sigma_2 u = f,$$

where u satisfies Dirichlet boundary conditions in $D = [0,1] \times [0,1]$. Using the five-point centered difference technique for the negative Laplacian operator on an uniform mesh with mesh-size $h = \frac{1}{m+1}$, we can obtain the complex symmetric linear system (1.1)–(1.2) of the form

$$[(H + \sigma_1 I) + i\sigma_2 I]x = b,$$

where $H = B_m \otimes I + I \otimes B_m$ with $B_m = h^{-2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$. Clearly, H is an $n \times n$ block-tridiagonal matrix with $n = m^2$. In addition, we set $\sigma_1 = 100$ and $b = (1 + i)Ae$, with $e = (1, 1, \ldots, 1)^T$. Further, by multiplying both sides by h^2, we can normalize the corresponding system.

To compare the MLPMHSS method with the LPMHSS method under the same conditions, based on Theorem 2.3, some values of the real coefficient function σ_2 are necessary to be selected. In this case, Table 1 lists some values of the real coefficient function σ_2.

Next, the spectral radius of MLPMHSS and LPMHSS iterative matrix is considered, because the spectral radius of iterative matrix largely determines

Table 1. The optimal parameters and the least upper bounds of MLPMHSS and LPMHSS for Example 3.1.

		σ_2	20	40	60	80	100
$n = 16384$	MLPMHSS	α^*	0.0279	0.1116	0.2511	0.4464	0.6975
		$\sigma(\alpha^*)$	0.0275	0.1059	0.2245	0.3712	0.5353
	LPMHSS	$\alpha^\star$	5.9869	2.9935	1.9956	1.4967	1.1974
		$\sigma(\alpha^\star)$	0.1647	0.3168	0.4480	0.5555	0.6410
$n = 65536$	MLPMHSS	α^*	0.0279	0.1116	0.2511	0.4464	0.6975
		$\sigma(\alpha^*)$	0.0275	0.1059	0.2245	0.3712	0.5353
	LPMHSS	$\alpha^\star$	5.9869	2.9935	1.9956	1.4967	1.1974
		$\sigma(\alpha^\star)$	0.1647	0.3168	0.4480	0.5555	0.6410

the convergence speed of iterative method. The comparison of spectral radii of two different iterative matrices obtained by MLPMHSS and LPMHSS is shown in Table 1. When the optimal parameter α^* of MLPMHSS method is selected by inference 2.1, and the optimal parameter $\alpha^\star$ of LPMHSS method is selected by Theorem 2.2. See Table 1 for details. The numerical results in Table 1 show that the theoretical results in Theorem 2.3 are valid.

From the numerical results in Table 1, fixing the mesh size m with σ_2 increasing, it is easy to see that the optimal parameter α^* and the upper bound $\sigma(\alpha^*)$ of the MLPMHSS method are increased, the optimal parameter $\alpha^\star$ of the LPMHSS method are decreased but the upper bound $\sigma(\alpha^\star)$ are increased. Fixing σ_2 with the mesh size m increasing, we find that the optimal parameter α^* and the upper bound $\sigma(\alpha^*)$ of the MLPMHSS method are decreased, the optimal parameter $\alpha^\star$ of the LPMHSS method are increased but the upper bound $\sigma(\alpha^\star)$ are decreased.

The numerical results of MLPMHSS and LPMHSS are given in Table 2 using the best parameters in Table 1. It can be seen from the numerical results in Table 2 that the iteration steps and CPU time of MLPMHSS and LPMHSS methods increase with the increase of σ_2. With the increase of mesh size m,

Table 2. IT and CPU of MLPMHSS and LPMHSS for Example 3.1.

		σ_2	20	40	60	80	100
$n = 16384$	MLPMHSS	IT	3	5	8	11	18
		CPU	0.609	0.984	1.578	2.187	3.547
	LPMHSS	IT	6	10	14	19	24
		CPU	1.235	2.063	2.875	3.828	4.875
$n = 65536$	MLPMHSS	IT	3	5	7	10	16
		CPU	4.109	6.765	9.454	13.609	21.719
	LPMHSS	IT	6	9	12	17	22
		CPU	8.203	12.516	16.656	23.516	29.672

the iteration steps of MLPMHSS and LPMHSS methods decrease after fixed σ_2. Not surprisingly, the CPU time of all methods increases with the grid size of m. It can be seen from the numerical results in Table 2 that the MLPMHSS method is superior to the LPMHSS method in iterative step size and CPU time in solving complex symmetric linear systems (1.1)–(1.2). Under certain conditions, MLPMHSS is more effective than LPMHSS for solving complex symmetric linear systems (1.1)–(1.2).

Example 3.2 ([2,3,9,15,16])**.** Consider the complex symmetric linear system

$$[(-\omega^2 M + K) + i(\omega C_V + C_H)]x = b,$$

where ω is the driving circular frequency, K and M are the stiffness and inertia matrices, C_H and C_V are the hysteretic damping and viscous matrices, respectively. In our numerical computations, we take $\omega = 1$, $M = I$, $C_V = 10I$, $C_H = \mu K$ with μ being a damping coefficient, where $K = I \otimes V_m + V_m \otimes I$ with $V_m = h^{-2}\text{tridiag}(-1, 2, -1)$ and the mesh-size $h = \frac{1}{m+1}$. In addition, we take $b = (1 + i)Ae$ with $e = (1, 1, \ldots, 1)^T$.

Table 3. The optimal parameters and the least upper bounds of MLPMHSS and LPMHSS for Example 3.2.

		μ	0.1	0.01	0.001
$n = 16384$	MLPMHSS	α^*	0.4083	0.2962	0.2859
		$\sigma(\alpha^*)$	0.3441	0.2602	0.2521
	LPMHSS	$\alpha^\star$	1.5649	1.8376	1.8701
		$\sigma(\alpha^\star)$	0.5385	0.4780	0.4715
$n = 65536$	MLPMHSS	α^*	0.4083	0.2961	0.2859
		$\sigma(\alpha^*)$	0.3441	0.2601	0.2521
	LPMHSS	$\alpha^\star$	1.5650	1.8376	1.8702
		$\sigma(\alpha^\star)$	0.5384	0.4780	0.4715

Analogously to Example 3.1, we select some values of μ to satisfy Theorem 2.3 such that we can compare the MLPMHSS method with the LPMHSS method under the same conditions. Specifically, see Table 3. Numerical results in Table 3 further confirm that the theoretical results in Theorem 2.3 are right.

From these numerical results in Table 3, fixing the mesh size m with μ decreasing, we find that the optimal parameter α^* and the upper bound $\sigma(\alpha^*)$ of the MLPMHSS method are decreased, the optimal parameter $\alpha^\star$ of the MLPMHSS method are increased but the upper bound $\sigma(\alpha^\star)$ are decreased. When $n = 256, 1024$ and 4096, fixing μ with the mesh size m increasing, the optimal parameter α^* and the upper bound $\sigma(\alpha^*)$ of the MLPMHSS method are decreased, and the optimal parameter $\alpha^\star$ of the LPMHSS method are increased but the upper bound $\sigma(\alpha^\star)$ are decreased. When $n = 16384$ and 65536, the optimal parameters and the upper bounds of both are almost unchanged, see Table 3.

Table 4. IT and CPU of MLPMHSS and LPMHSS for Example 3.2.

		μ	0.1	0.01	0.001
$n = 16384$	MLPMHSS	IT	9	7	7
		CPU	1.75	1.36	1.359
	LPMHSS	IT	15	12	12
		CPU	2.969	2.391	2.375
$n = 65536$	MLPMHSS	IT	8	6	6
		CPU	10.469	7.828	7.828
	LPMHSS	IT	13	11	11
		CPU	17.172	14.547	14.453

Table 4 lists the numerical results of the MLPMHSS and LPMHSS methods for Example 3.2 when the optimal parameters in Table 3 are employed. From Table 4, fixing the mesh size m with μ decreasing, the number of iteration steps and CPU times of the MLPMHSS and LPMHSS methods decrease. Fixing μ with the mesh size m increasing, the number of iteration steps of the MLPMHSS and LPMHSS methods are decreased. It is no surprise that CPU times for all methods grow with the mesh size m. Based on the numerical results in Table 4, from the view of iteration step and CPU time, the MLPMHSS method is superior to the LPMHSS method. That is to say, the MLPMHSS method is more efficient than the LPMHSS method when both are used to solve the complex symmetric linear system (1.1)–(1.2).

In all, by our numerical experiments, the MLPMHSS method is superior to the LPMHSS method under certain conditions when both are employed to solve the complex symmetric linear system (1.1)–(1.2).

3.2 Results of the Related Preconditioner

In this subsection, we consider the related preconditioner for the complex symmetric linear system (1.1)–(1.2). When $V = W$, the preconditioner $P_2 = \alpha W + T$ was considered in [12]. In the same way, we take the preconditioner $P_1 = \alpha W + W = (1 + \alpha)W$. It is noted that the multiplicative factor $1 + \alpha$ in the preconditioner P_1 has no influence on the preconditioned system and thus it can be dropped. Whereas, for convenient comparison, in our numerical experiment, the multiplicative factor $1 + \alpha$ in the preconditioner P_1 is not deleted.

For Example 3.1, in Tables 5, 6, 7, 8 and 9, we present some numerical results of GMRES(20) with P_1 and P_2 for solving the complex symmetric linear system (1.1)–(1.2).

Table 5. IT and CPU of P_1 and P_2 for $\sigma_2 = 20$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	2	2	2	2
		CPU	0.437	0.453	0.453	0.453
	P_2	IT	16	8	6	3
		CPU	1.9061	1.047	0.844	0.531
$n = 65536$	P_1	IT	2	2	2	2
		CPU	2.765	2.719	2.765	2.75
	P_2	IT	15	7	5	3
		CPU	11.89	6.093	4.719	3.344

Table 6. IT and CPU of P_1 and P_2 for $\sigma_2 = 40$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	3	3	3	3
		CPU	0.547	0.547	0.547	0.563
	P_2	IT	21	10	7	4
		CPU	2.594	1.25	0.937	0.625
$n = 65536$	P_1	IT	2	2	2	2
		CPU	2.719	2.719	2.719	2.672
	P_2	IT	19	9	6	3
		CPU	14.969	7.578	5.406	3.297

Table 7. IT and CPU of P_1 and P_2 for $\sigma_2 = 60$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	3	3	3	3
		CPU	0.531	0.531	0.531	0.531
	P_2	IT	26	11	8	4
		CPU	3	1.328	1.016	0.594
$n = 65536$	P_1	IT	3	3	3	3
		CPU	3.344	3.328	3.328	3.28
	P_2	IT	24	10	7	4
		CPU	18.75	8.157	6	3.985

For Example 3.2, Tables 10, 11 and 12 present some numerical results of GMRES(20) with P_1 and P_2 for solving the complex symmetric linear system (1.1)–(1.2).

To easily compare the precondtioner P_1 with the precondtioner P_2, the same iteration parameter α is employed. With respect to the choice of α, one can

Table 8. IT and CPU of P_1 and P_2 for $\sigma_2 = 80$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	4	4	3	3
		CPU	0.656	0.641	0.531	0.531
	P_2	IT	29	12	9	5
		CPU	3.36	1.437	1.109	0.703
$n = 65536$	P_1	IT	3	3	3	3
		CPU	3.328	3.328	3.328	3.328
	P_2	IT	26	11	8	4
		CPU	20.015	8.781	6.75	4.016

Table 9. IT and CPU of P_1 and P_2 for $\sigma_2 = 100$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	4	4	4	4
		CPU	0.625	0.641	0.641	0.625
	P_2	IT	32	13	10	5
		CPU	3.625	1.531	1.219	0.687
$n = 65536$	P_1	IT	3	3	3	3
		CPU	3.391	3.328	3.343	3.344
	P_2	IT	28	12	9	4
		CPU	21.391	9.5	7.437	3.969

Table 10. IT and CPU of P_1 and P_2 for $\mu = 0.1$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	3	3	3	3
		CPU	0.532	0.531	0.515	0.516
	P_2	IT	6	5	5	3
		CPU	0.812	0.703	0.703	0.531
$n = 65536$	P_1	IT	3	3	3	2
		CPU	3.359	3.359	3.344	2.672
	P_2	IT	5	5	4	3
		CPU	4.719	4.625	3.953	3.281

see [17] for more details. In this case, some numerical results are presented in Tables 5, 6, 7, 8, 9, 10, 11 and 12 to illustrate the convergence behaviors of two preconditioners P_1 and P_2.

From Tables 5, 6, 7, 8, 9, 10, 11 and 12, it is easy to see that the iteration steps and CPU times of the preconditioner P_1 are less than the preconditioner

Table 11. IT and CPU of P_1 and P_2 for $\mu = 0.01$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	3	3	3	3
		CPU	0.531	0.531	0.531	0.531
	P_2	IT	12	8	6	4
		CPU	1.422	1.047	0.797	0.61
$n = 65536$	P_1	IT	3	3	3	2
		CPU	3.391	3.344	3.344	2.703
	P_2	IT	11	7	5	3
		CPU	8.828	6.063	4.64	3.13

Table 12. IT and CPU of P_1 and P_2 for $\mu = 0.001$.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	3	3	3	3
		CPU	0.531	0.531	0.531	0.531
	P_2	IT	16	8	6	4
		CPU	1.875	1.016	0.812	0.609
$n = 65536$	P_1	IT	3	3	3	2
		CPU	3.344	3.36	3.359	2.688
	P_2	IT	15	7	6	3
		CPU	11.703	6.031	5.328	3.296

P_2 in [12]. When used as a preconditioner, the computational efficiency of P_1 performs much better than P_2. Based on these numerical results, we can draw a conclusion that the preconditioner P_1 has considerable competition.

Finally, the following complex symmetric linear system is considered.

Example 3.3 [2]**.** Consider the complex symmetric linear system of the form

$$(W + iT)x = b,$$

with

$$K = I \otimes V + V \otimes I \text{ and } W = 10(I \otimes V_c + V_c \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I,$$

where $V =$tridiag$(-1, 2, -1) \in \mathbb{R}^{m \times m}$, $V_c = V - e_1 e_m^T - e_m e_1^T \in \mathbb{R}^{m \times m}$, and e_1 and e_m are the first and the last unit vectors in $\mathbb{R}^m$, respectively. We take $b = (1 + i)Ae$ with $e = (1, 1, \ldots, 1)^T$. Specifically, see Example 4.3 in [2].

From the numerical results in Tables 2, 4, 5, 6, 7, 8, 9, 10, 11 and 12, we find that the MLPMHSS method outperforms the LPMHSS method, the preconditioner P_1 also outperforms the preconditioner P_2. Further, we can find that the related preconditioner $P_1(P_2)$ with GMRES(20) is more efficiency than the MLPMHSS(LPMHSS) method. Based on this case, we only consider the efficiency of the preconditioners P_1 and P_2.

Table 13 reports the numerical results for GMRES(20) with P_1 and P_2 for Example 3.3.

Table 13. IT and CPU of P_1 and P_2 for Example 3.3.

		α	0.01	0.05	0.1	0.5
$n = 16384$	P_1	IT	8	8	8	8
		CPU	1.609	1.578	1.578	1.547
	P_2	IT	19	16	14	9
		CPU	3.437	2.86	2.562	1.719
$n = 65536$	P_1	IT	10	10	10	10
		CPU	15.171	15.094	15.125	15.188
	P_2	IT	25	19	17	11
		CPU	36.359	27.296	24.672	16.641

Compared with the preconditioner P_2, Table 13 further confirms that the preconditioner P_1 has considerable competition.

4 Conclusion

In this paper, a modified LPMHSS (MLPMHSS) method is proposed for a class of complex symmetric linear systems with strong Hermitian parts, and its convergence conditions are given. It is proved that the MLPMHSS method converges unconditionally under suitable conditions. The effectiveness of MLPMHSS is verified by three examples.

References

1. Arridge, S.R.: Optical tomography in medical imaging. Inverse Prob. **15**, R41–R93 (1999)
2. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
3. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algor. **56**, 297–317 (2011)
4. Bai, Z.-Z.: Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models. Numer. Linear Algebra Appl. **19**, 914–936 (2012)
5. Bai, Z.-Z.: Structured preconditioners for nonsingular matrices of block two-by-two structures. Math. Comput. **75**, 791–815 (2006)

6. Benzi, M., Bertaccini, D.: Block preconditioning of real-valued iterative algorithms for complex linear systems. IMA J. Numer. Anal. **28**, 598–618 (2008)
7. Bertaccini, D.: Efficient solvers for sequences of complex symmetric linear systems. Electron. Trans. Numer. Anal. **18**, 49–64 (2004)
8. Day, D.D., Heroux, M.A.: Solving complex-valued linear systems via equivalent real formulations. SIAM J. Sci. Comput. **23**, 480–498 (2001)
9. Feriani, A., Perotti, F., Simoncini, V.: Iterative system solvers for the frequency analysis of linear mechanical systems. Comput. Methods Appl. Mech. Eng. **190**, 1719–1739 (2000)
10. Bai, Z.-Z.: On preconditioned iteration methods for complex linear systems. J. Eng. Math. **93**(1), 41–60 (2014). https://doi.org/10.1007/s10665-013-9670-5
11. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)
12. Li, X., Yang, A.-L., Wu, Y.-J.: Lopsided PMHSS iteration method for a class of complex symmetric linear systems. Numer. Algorithms **66**(3), 555–568 (2013). https://doi.org/10.1007/s11075-013-9748-1
13. Noormohammadi Pour, H., Sadeghi Goughery, H.: New Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems. Numer. Algorithms **69**(1), 207–225 (2014). https://doi.org/10.1007/s11075-014-9890-4
14. Wu, S.-L., Huang, T.-Z., Li, L., Xiong, L.-L.: Positive stable preconditioners for symmetric indefinite linear systems arising from Helmholtz equations. Phys. Lett. A **373**, 2401–2407 (2009)
15. Wu, S.-L.: Several variants of the HSS method for a class of complex symmetric linear systems. Numer. Linear Algebra Appl. **22**, 338–356 (2015)
16. Wu, S.-L., Li, C.-X.: A splitting iterative method for the discrete dynamic linear systems. J. Comput. Appl. Math. **267**, 49–60 (2014)
17. Benzi, M.: A generalization of the Hermitian and skew-Hermitian splitting iteration. SIAM J. Matrix Anal. Appl. **31**, 360–374 (2009)