Received: 26 April 2017 | Revised: 4 February 2018 | Accepted: 13 February 2018

DOI: 10.1002/nla.2169

SPECIAL ISSUE PAPER

WILEY

Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners

Ai-Li Yang

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China

Correspondence
Ai-Li Yang, School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China.
Email: yangaili@lzu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 11401281; Fundamental Research Fund for the Central Universities, Grant/Award Number: lzujbky-2016-97

Summary

The performance of the Hermitian and skew-Hermitian splitting (HSS) preconditioner for the non-Hermitian positive definite system of linear equations is largely dependent on the choice of its parameter value. In this work, an efficient scaled norm minimization (SNM) method is proposed to compute the parameter value of the HSS preconditioner. In addition, by choosing different parameters for the Hermitian and the skew-Hermitian matrices in the HSS preconditioner, a two-parameter HSS preconditioner is proposed. Moreover, an efficient and practical formula for computing the parameter values of this new preconditioner is also derived by using the SNM method. Numerical examples are illustrated to verify the performances of the HSS and the two-parameter HSS preconditioners when their parameters are computed by the SNM method.

KEYWORDS

HSS preconditioner, matrix trace, parameter value, preconditioning effect, scaled norm minimization method

1 | INTRODUCTION

Many problems in scientific computing give rise to a system of linear equations

$$Ax = b, \quad A \in \mathbb{C}^{n\times n}, \quad x, b \in \mathbb{C}^n, \tag{1}$$

where A is a non-Hermitian positive definite matrix.

When the coefficient matrix A is large and sparse, iteration methods are more attractive than direct methods for solving the system of linear equations (1) because the direct methods are prohibitively expensive. Moreover, the accumulation of the rounding error may lead to inaccurate numerical solution.[1–3] As a class of important iteration methods, Krylov subspace methods are always employed to solve different kinds of systems of linear equations. However, with the increase of the problem size, not only the storage space needed by Krylov subspace methods always increases very fast, but also the convergence speed of the Krylov subspace methods will become slow. An efficient remedy to avoid these disadvantages is to design good preconditioners for the coefficient matrix A and then use these preconditioners to accelerate the convergence speed of the Krylov subspace methods.[4–7]

Let $A = H + S$ be the Hermitian and skew-Hermitian splitting (HSS) of the matrix A, where

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

Dedicated to Professor Yu-Jiang Wu on the occasion of his sixtieth birthday.

are the Hermitian and the skew-Hermitian parts of matrix A, respectively. Based on this matrix splitting, Bai et al.[8] introduced an HSS iteration method to approximate the solution of system of linear equations (1). The iteration scheme of the HSS iteration method has the form of

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} \ \ = (\alpha I - H)x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2}$$

where $k = 0, 1, 2, \ldots$, parameter α is a given positive constant, and $x^{(0)}$ is an arbitrary initial guess. Theoretical analysis shows that the iteration sequence generated by the HSS method converges unconditionally to the unique solution of the system of linear equations (1) for any $\alpha > 0$ and any initial guess $x^{(0)} \in \mathbb{C}^n$. Moreover, a quasioptimal parameter α was defined by minimizing an upper bound of the spectral radius of the HSS iteration matrix. Numerical results show that the HSS iteration method is very robust and efficient for solving the non-Hermitian positive definite systems of linear equations.[8]

Due to its promising performance and elegant mathematical properties, the HSS scheme immediately attracted wide attention and was used to solve different kinds of problems, such as saddle-point problems,[9–15] complex linear systems,[16–20] certain singular problems,[21,22] and nonlinear problems.[23,24] As it is used as a solver, the HSS method was also used as a preconditioner to accelerate the convergence speed of the Krylov subspace methods.[25–28] This preconditioner has the form of

$$P(\alpha) = \frac{1}{2\alpha}(\alpha I + H)(\alpha I + S), \quad \alpha > 0. \tag{3}$$

Because the performance of $P(\alpha)$ is different for different values of parameter α, it is very important to choose a suitable parameter value to optimize the performance of the preconditioner $P(\alpha)$. In 2003, together with the introduction of the HSS method, Bai et al.[8] defined a quasioptimal value for the parameter α by minimizing an upper bound of the spectral radius of iteration matrix. A few years later, by reducing the coefficient matrix A into a 2×2 matrix, Bai et al.[29] gave a more accurate formula to compute the optimal parameter value of the HSS method. In 2015, Chen[30] proposed another interesting method to compute the optimal parameter value by balancing the computing efficiencies of the two subsystems of linear equations in each step of the HSS scheme. Using the parameter values computed by the three methods, both the HSS iteration method and the HSS preconditioner are very efficient. However, the computation of the optimal values of parameter α with any one of the aforementioned three methods is a time-consuming work because it needs to compute the maximum and the minimum eigenvalues of the matrix H at first.

Denote by $N(\alpha)$ the difference of the coefficient matrix A and the preconditioner $P(\alpha)$, that is,

$$N(\alpha) = A - P(\alpha) = A - \frac{1}{2\alpha}(\alpha I + H)(\alpha I + S). \tag{4}$$

We know that if the matrix A and its preconditioner $P(\alpha)$ are closer, the preconditioner $P(\alpha)$ should be more efficient. Based on this idea, Huang[31] introduced an efficient method to compute the parameter α by minimizing $\|2\alpha N(\alpha)\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. In this way, the parameter α can be easily obtained by solving a simple cubic equation whose coefficients are represented by the traces of several matrices. This method avoids the computation of the eigenvalues of a matrix. However, we should note that the true difference of the coefficient matrix A and the preconditioner $P(\alpha)$ is $N(\alpha)$ rather than $2\alpha N(\alpha)$. Therefore, it is more reasonable to compute parameter α by minimizing $\|N(\alpha)\|_F$ rather than $\|2\alpha N(\alpha)\|_F$.

As it is well known, the constant $1/(2\alpha)$ of the preconditioner $P(\alpha)$ has almost no impact on its preconditioning effect. However, because the differences of matrix A and its preconditioners, that is,

$$A - (\alpha I + H)(\alpha I + S) \quad \text{and} \quad A - \frac{1}{2\alpha}(\alpha I + H)(\alpha I + S),$$

are different, we will obtain different values of parameter α by minimizing the Frobenius norms of the above two differences, respectively. Therefore, we can further find that the above norm minimization method will lead to different values of parameter α if we choose different coefficients for the preconditioner $(\alpha I + H)(\alpha I + S)$.

In view of these problems, in this work, we will propose a scaled norm minimization (SNM) method to compute the parameter values of the HSS preconditioner. The remainder of this work is organized as follows. In Section 2, using the SNM method, we derive a practical formula for computing the parameter values of the HSS preconditioner. By employing different parameters for the Hermitian and the skew-Hermitian matrices in the HSS preconditioner, in Section 3, we define a two-parameter HSS (TPHSS) preconditioner. An efficient formula for computing the parameter values of this new preconditioner is also derived by using the SNM method. In Section 4, three examples are employed to test the preconditioning effects of the HSS and the TPHSS preconditioners in which the parameters are computed by the SNM method. Finally, in Section 5, we give a brief conclusion for this paper.

2 | THE SNM METHOD FOR COMPUTING THE OPTIMAL PARAMETER VALUES OF THE HSS PRECONDITIONER

In this section, to optimize the performance of the HSS preconditioner, we introduce a practical formula to compute the optimal parameter value of the HSS preconditioner by using the SNM method.

Now, we introduce a scaled parameter ζ for the HSS preconditioner $(\alpha I + H)(\alpha I + S)$. This leads to

$$P(\alpha, \zeta) := \zeta(\alpha I + H)(\alpha I + S),$$

where α and ζ are two positive constants. Although the parameter ζ does not affect the performance of the HSS preconditioner, it may help us achieve better optimal value of the parameter α.

Let $\Omega^+ = \{(\alpha, \zeta) \in \mathbb{R}^2 \,|\, \alpha > 0,\ \zeta > 0\}$, and $N(\alpha, \zeta)$ be the difference of the HSS preconditioner $P(\alpha, \zeta)$ and coefficient matrix A, that is,

$$N(\alpha, \zeta) = P(\alpha, \zeta) - A = \alpha^2\zeta I + (\alpha\zeta - 1)S + (\alpha\zeta - 1)H + \zeta HS.$$

Then, the optimal values of α and ζ for the HSS preconditioner can be defined as

$$(\alpha_{\text{SNM}}, \zeta_{\text{SNM}}) = \arg\min_{(\alpha,\zeta)\in\Omega^+} \|N(\alpha, \zeta)\|_F. \tag{5}$$

Because the optimal values are defined by minimizing the norm of the difference of the HSS preconditioner $P(\alpha, \zeta)$ with a scaled parameter ζ and the coefficient matrix A, we call the method SNM method.

Remark 1. We should note that minimizing $\|N(\alpha, \zeta)\|_F$, or $\|P(\alpha, \zeta) - A\|_F$ equivalently, is just an auxiliary construction and not directly related to the performance of the HSS preconditioner, which is much more determined by the size of $\|I - P(\alpha, \zeta)^{-1}A\|_F$.

In the following, we try to derive a practical formula for computing the optimal parameters α_{SNM} and ζ_{SNM}. Denote by $\text{tr}(\cdot)$ the trace of a matrix. It is easy to see that $\text{tr}(A^*B)$ is an inner product between square matrices A and B. Owing to $\|N(\alpha, \zeta)\|_F^2 = \text{tr}((N(\alpha, \zeta))^* N(\alpha, \zeta))$, we first give

$$\begin{aligned}(N(\alpha, \zeta))^* N(\alpha, \zeta) &= \alpha^4\zeta^2 I + 2\alpha^2\zeta(\alpha\zeta - 1)H + \left(2\alpha^2\zeta^2 - 2\alpha\zeta + 1\right)(HS + (HS)^*) \\ &\quad + (\alpha\zeta - 1)^2\left(S^*S + H^2\right) + 2\zeta(\alpha\zeta - 1)S^*HS \\ &\quad + \zeta(\alpha\zeta - 1)\left(H^2S + S^*H^2\right) + \zeta^2((HS)^*HS).\end{aligned}$$

Because $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$ and $\text{tr}(AB) = \text{tr}(BA)$ for any $A, B \in \mathbb{C}^{n\times n}$, it follows that

$$\text{tr}(HS + (HS)^*) = 0, \quad \text{tr}\left(H^2S + S^*H^2\right) = 0. \tag{6}$$

Denote $\Psi(\alpha, \zeta) := \text{tr}((N(\alpha, \zeta))^* N(\alpha, \zeta))$. Then, simple calculation gives

$$\Psi(\alpha, \zeta) = n\alpha^4\zeta^2 + c_1\alpha^2\zeta(\alpha\zeta - 1) + (c_2 + c_3)(\alpha\zeta - 1)^2 + c_4\zeta(\alpha\zeta - 1) + c_5\zeta^2, \tag{7}$$

where

$$c_1 = 2\text{tr}(H), \quad c_2 = \text{tr}(S^*S), \quad c_3 = \text{tr}(H^2), \quad c_4 = 2\text{tr}(S^*HS), \quad c_5 = \text{tr}((HS)^*HS). \tag{8}$$

Thus, the optimal parameters $(\alpha_{\text{SNM}}, \zeta_{\text{SNM}})$ defined by (5) yield

$$(\alpha_{\text{SNM}}, \zeta_{\text{SNM}}) = \arg\min_{(\alpha,\zeta)\in\Omega^+} \Psi(\alpha, \zeta). \tag{9}$$

To derive the computing formula for the optimal parameters α_{SNM} and ζ_{SNM}, we first give a useful lemma.

Lemma 1. *Let A be a non-Hermitian positive definite matrix. Matrices H and S are, respectively, the Hermitian and the skew-Hermitian parts of matrix A. Then, we have*

1. $c_i > 0$, *for* $i = 1, \ldots, 5$.
2. $c_1^2 - 4c_3n \le 0$; *the "=" holds if and only if the matrix H is a multiple of an identity matrix, that is,* $H = cI (c > 0)$.
3. $c_4^2 - 4c_2c_5 \le 0$; *the "=" holds if and only if HS is a multiple of S.*

Proof. For any square matrix, its trace equals to the sum of all its eigenvalues. Hence, it is obvious that $c_1 = 2\mathrm{tr}(H) > 0$ and $c_3 = \mathrm{tr}(H^2) > 0$ because matrix H is Hermitian positive definite. Inasmuch as matrix A is non-Hermitian, or $S \neq 0$ equivalently, it follows that S^*S, S^*HS and $(HS)^*HS$ are also nonzero matrices. Using the fact that the trace of a Hermitian positive semidefinite matrix P is zero if and only if $P = 0$, we can immediately obtain that $c_i > 0 (i = 2, 4, 5)$ according to matrices S^*S, S^*HS, and $(HS)^*HS$ are nonzero Hermitian positive semidefinite. Actually, this fact can also be obtained by noticing that $\mathrm{tr}(A^*B)$ is an inner product between square matrices A and B.

Let $\lambda_i (> 0, i = 1, \dots, n)$ be the eigenvalues of matrix H; then, using the Cauchy–Schwarz inequality gives

$$c_1^2 - 4c_3 n = 4\left(\mathrm{tr}(H)^2 - n \quad \mathrm{tr}\left(H^2\right)\right) = 4\left(\left(\sum_{i=1}^{n} \lambda_i\right)^2 - n\sum_{i=1}^{n} \lambda_i^2\right) \leq 0,$$

where the "=" holds if and only if $\lambda_1 = \lambda_2 = \cdots = \lambda_n$, or equivalently, matrix H is a multiple of the identity matrix, that is, $H = cI$, $c > 0$ is a real number.

For any matrices $M, N \in \mathbb{C}^{n\times n}$, it follows that $|\mathrm{tr}(M^*N)|^2 \leq \mathrm{tr}(M^*M)\mathrm{tr}(N^*N)$, where the "=" holds if and only if M is a multiple of N. Then, we have

$$c_4^2 - 4c_2c_5 = 4\left(\mathrm{tr}(S^*HS)^2 - \mathrm{tr}(S^*S) \cdot \mathrm{tr}\left((HS)^*HS\right)\right) \leq 0,$$

where the "=" holds if and only if HS is a multiple of S. □

The partial derivative of function $\Psi(\alpha, \zeta)$ in the variable ζ yields

$$\begin{aligned}\Psi'_\zeta(\alpha, \zeta) &= 2n\alpha^4\zeta + c_1\alpha^2(2\alpha\zeta - 1) + 2\alpha(c_2 + c_3)(\alpha\zeta - 1) + c_4(2\alpha\zeta - 1) + 2c_5\zeta \\ &= (2n\alpha^4 + 2c_1\alpha^3 + 2(c_2 + c_3)\alpha^2 + 2c_4\alpha + 2c_5)\zeta - c_1\alpha^2 - 2(c_2 + c_3)\alpha - c_4.\end{aligned} \tag{10}$$

Hence, for fixed $\alpha > 0$, the root of $\Psi'_\zeta(\alpha, \zeta) = 0$ has the form

$$\zeta = \zeta(\alpha) := \frac{c_1\alpha^2 + 2(c_2 + c_3)\alpha + c_4}{2n\alpha^4 + 2c_1\alpha^3 + 2(c_2 + c_3)\alpha^2 + 2c_4\alpha + 2c_5}. \tag{11}$$

Using Lemma 1, we have $\zeta(\alpha) > 0$. Moreover, it follows that $\Psi'_\zeta(\alpha, \zeta) > 0$ when $\zeta > \zeta(\alpha)$, and $\Psi'_\zeta(\alpha, \zeta) < 0$ when $\zeta < \zeta(\alpha)$. This means that the function $\Psi(\alpha, \zeta)$ achieves its minimum value on the curve $\zeta = \zeta(\alpha)$.

Now, substituting $\zeta = \zeta(\alpha)$ into the function $\Psi(\alpha, \zeta)$ gives

$$\tilde{\Psi}(\alpha) := \Psi(\alpha, \zeta(\alpha)) = \frac{\left(4n(c_2 + c_3) - c_1^2\right)\alpha^4 - 2c_1c_4\alpha^2 + 4(c_2 + c_3)c_5 - c_4^2}{4\left(n\alpha^4 + c_1\alpha^3 + (c_2 + c_3)\alpha^2 + c_4\alpha + c_5\right)}. \tag{12}$$

Thus, the optimal parameter α_{SNM} defined in (5), or equivalently in (9), is also the minimum point of the function $\tilde{\Psi}(\alpha)$, that is,

$$\arg\min_{\alpha\in(0,+\infty)} \tilde{\Psi}(\alpha) = \alpha_{\mathrm{SNM}}. \tag{13}$$

In the following, we only need to find the minimum point of the function $\tilde{\Psi}(\alpha)$ in the interval $(0, +\infty)$. Note that $\tilde{\Psi}(\alpha)$ is differentiable in the interval $(0, +\infty)$, and careful calculation gives

$$\tilde{\Psi}'(\alpha) = \frac{\left(c_1\alpha^2 + 2(c_2 + c_3)\alpha + c_4\right) \cdot \varphi(\alpha)}{4\left(n\alpha^4 + \alpha^3c_1 + \alpha^2(c_2 + c_3) + \alpha c_4 + c_5\right)^2}, \tag{14}$$

where

$$\varphi(\alpha) = \left(4n(c_2 + c_3) - c_1^2\right)\alpha^4 + 4nc_4\alpha^3 - 4c_1c_5\alpha + c_4^2 - 4(c_2 + c_3)c_5.$$

Using Lemma 1, we have $c_1\alpha^2 + 2(c_2 + c_3)\alpha + c_4 > 0$, $4n(c_2 + c_3) - c_1^2 > 0$, and $c_4^2 - 4(c_2 + c_3)c_5 < 0$. Thus, the function $\tilde{\Psi}'(\alpha)$ yields $\tilde{\Psi}'(0) < 0$, and there exists a positive constant τ such that $\tilde{\Psi}'(\alpha) > 0$ for $\alpha \geq \tau$. This means the function $\tilde{\Psi}(\alpha)$ achieves its minimum in the interval $(0, \tau)$, that is, $\alpha_{\mathrm{SNM}} \in (0, \tau)$. Moreover, from (14), the minimum point of $\tilde{\Psi}(\alpha)$, that is, the optimal parameter α_{SNM}, is a root of the quartic equation $\varphi(\alpha) = 0$.

The above analysis can be summarized as follows.

Theorem 1. *Let A be a non-Hermitian positive definite matrix. Matrices H and S are, respectively, the Hermitian and the skew-Hermitian parts of matrix A. Then, the optimal parameter α_{SNM} of the HSS preconditioner defined by* (5) *(or (9) equivalently) exists. Moreover, it is the one positive root of the quartic equation $\varphi(\alpha) = 0$ that minimizes the function $\tilde{\Psi}(\alpha)$. Substituting $\alpha = \alpha_{\mathrm{SNM}}$ into the function $\zeta = \zeta(\alpha)$ defined in (11), we also obtained the optimal parameter ζ_{SNM}.*

Remark 2. Because the choice of the parameter ζ does not affect the performance of the HSS preconditioner, we only need to calculate the optimal parameters α_{SNM} in the implementation of the HSS preconditioner.

3 | THE TPHSS PRECONDITIONER AND A PRACTICAL FORMULA FOR COMPUTING ITS OPTIMAL PARAMETERS

Note that the Hermitian and the skew-Hermitian matrices H and S in the HSS preconditioner have different properties, it should be better to use different shift parameters for the two matrices, respectively.[32] This leads to a TPHSS preconditioner of the form $(\alpha I + H)(\beta I + S)$, where $\alpha \geq 0$ and $\beta > 0$.

To derive a practical formula for computing the optimal parameters of the TPHSS preconditioner, we first introduce a positive scaled parameter ζ for the TPHSS preconditioner, which leads to

$$P(\alpha, \beta, \zeta) = \zeta(\alpha I + H)(\beta I + S). \tag{15}$$

Denote by $N(\alpha, \beta, \zeta)$ the difference of the TPHSS preconditioner $P(\alpha, \beta, \zeta)$ and the coefficient matrix A, that is,

$$N(\alpha, \beta, \zeta) = P(\alpha, \beta, \zeta) - A = \alpha\beta\zeta I + (\alpha\zeta - 1)S + (\beta\zeta - 1)H + \zeta HS.$$

The optimal parameters of the preconditioner (15) can be similarly defined as

$$\left(\bar{\alpha}_{\mathrm{SNM}}, \bar{\beta}_{\mathrm{SNM}}, \bar{\zeta}_{\mathrm{SNM}}\right) = \underset{(\alpha,\beta,\zeta)\in\Omega_1^+}{\arg\min} \|N(\alpha, \beta, \zeta)\|_F, \tag{16}$$

where $\Omega_1^+ = \{(\alpha, \beta, \zeta) \in \mathbb{R}^3 \,|\, \alpha \geq 0,\, \beta > 0,\, \zeta > 0\}$.

Owing to $\|N(\alpha, \beta, \zeta)\|_F^2 = \mathrm{tr}((N(\alpha, \beta, \zeta))^* N(\alpha, \beta, \zeta))$, we can derive that

$$\begin{aligned}
\|N(\alpha, \beta, \zeta)\|_F^2 = {} & \alpha^2\beta^2\zeta^2 I + 2\alpha\beta\zeta(\beta\zeta - 1)H + (2\alpha\beta\zeta^2 - (\alpha + \beta)\zeta + 1)(HS + (HS)^*) \\
& + (\alpha\zeta - 1)^2 S^* S + 2(\alpha\zeta - 1)\zeta S^* HS + (\beta\zeta - 1)^2 H^2 \\
& + (\beta\zeta - 1)\zeta(H^2 S + S^* H^2) + \zeta^2((HS)^* HS).
\end{aligned}$$

Denote $\Phi(\alpha, \beta, \zeta) := \|N(\alpha, \beta, \zeta)\|_F^2$. Using (6) and (8), the above equality can be simplified as

$$\Phi(\alpha, \beta, \zeta) = n\alpha^2\beta^2\zeta^2 + c_1\alpha\beta\zeta(\beta\zeta - 1) + c_2(\alpha\zeta - 1)^2 + c_3(\beta\zeta - 1)^2 + c_4(\alpha\zeta - 1)\zeta + c_5\zeta^2. \tag{17}$$

Then, the optimal parameters defined in (16) yield

$$\left(\bar{\alpha}_{\mathrm{SNM}}, \bar{\beta}_{\mathrm{SNM}}, \bar{\zeta}_{\mathrm{SNM}}\right) = \underset{(\alpha,\beta,\zeta)\in\Omega_1^+}{\arg\min} \Phi(\alpha, \beta, \zeta). \tag{18}$$

To derive the practical formula for computing the optimal parameter values, we first give

$$\begin{aligned}
\Phi'_\zeta(\alpha, \beta, \zeta) &= 2n\alpha^2\beta^2\zeta + c_1\alpha\beta(2\beta\zeta - 1) + 2\alpha c_2(\alpha\zeta - 1) + 2\beta c_3(\beta\zeta - 1) + c_4(2\alpha\zeta - 1) + 2c_5\zeta \\
&= \left(2n\alpha^2\beta^2 + 2c_2\alpha^2 + 2c_1\alpha\beta^2 + 2c_4\alpha + 2c_3\beta^2 + 2c_5\right)\zeta - c_4 - 2\alpha c_2 - 2\beta c_3 - \alpha\beta c_1.
\end{aligned}$$

Then, for fixed $\alpha \geq 0$ and $\beta > 0$, the root of $\Phi'_\zeta(\alpha, \beta, \zeta) = 0$ has the form

$$\zeta = \zeta(\alpha, \beta) := \frac{c_4 + 2\alpha c_2 + 2\beta c_3 + \alpha\beta c_1}{2n\alpha^2\beta^2 + 2c_2\alpha^2 + 2c_1\alpha\beta^2 + 2c_4\alpha + 2c_3\beta^2 + 2c_5}. \tag{19}$$

Using Lemma 1, we have $\zeta(\alpha, \beta) > 0$. Moreover, it follows that $\Phi'_\zeta(\alpha, \beta, \zeta) > 0$ when $\zeta > \zeta(\alpha, \beta)$, and $\Phi'_\zeta(\alpha, \beta, \zeta) < 0$ when $\zeta < \zeta(\alpha, \beta)$. This means that the function $\Phi(\alpha, \beta, \zeta)$ achieves its minimum value on the surface $\zeta = \zeta(\alpha, \beta)$.

Substituting $\zeta = \zeta(\alpha, \beta)$ into the function $\Phi(\alpha, \beta, \zeta)$ gives

$$\bar{\Phi}(\alpha, \beta) := \Phi(\alpha, \beta, \zeta(\alpha, \beta)) = \frac{k_2(\alpha)\beta^2 + k_1(\alpha)\beta + k_0(\alpha)}{4(l_2(\alpha)\beta^2 + l_0(\alpha))},$$

where $l_0(\alpha) = c_2\alpha^2 + c_4\alpha + c_5$, $l_2(\alpha) = n\alpha^2 + c_1\alpha + c_3$, and

$$\begin{aligned}
k_2(\alpha) &= 4c_2c_3 - \alpha^2c_1^2 + 4\alpha c_1c_2 + 4\alpha^2c_2n + 4\alpha^2c_3n,\\
k_1(\alpha) &= -4c_3c_4 - 2\alpha c_1c_4 - 8\alpha c_2c_3 - 4\alpha^2c_1c_2,\\
k_0(\alpha) &= 4c_2c_3\alpha^2 + 4c_3\alpha c_4 - c_4^2 + 4c_2c_5 + 4c_3c_5.
\end{aligned}$$

In the following, we only need to find the minimum point of function $\bar{\Phi}(\alpha,\beta)$ in the domain $\Omega_2^+ = \{(\alpha,\beta)\,|\,\alpha \geq 0,\ \beta > 0\}$. This is because the optimal parameters $\bar{\alpha}_{\text{SNM}}$ and $\bar{\beta}_{\text{SNM}}$ defined in (16) yield

$$\underset{(\alpha,\beta)\in\Omega_2^+}{\arg\min}\ \bar{\Phi}(\alpha,\beta) = (\bar{\alpha}_{\text{SNM}}, \bar{\beta}_{\text{SNM}}).$$

To calculate the optimal parameters $\bar{\alpha}_{\text{SNM}}$ and $\bar{\beta}_{\text{SNM}}$, we further consider the first-order partial derivative of $\bar{\Phi}(\alpha,\beta)$ in the variable β, that is,

$$\frac{\partial\bar{\Phi}(\alpha,\beta)}{\partial\beta} = \frac{-k_1(\alpha)l_2(\alpha)\beta^2 + 2\left(k_2(\alpha)l_0(\alpha) - l_2(\alpha)k_0(\alpha)\right)\beta + k_1(\alpha)l_0(\alpha)}{4\left(l_2(\alpha)\beta^2 + l_0(\alpha)\right)^2}. \tag{20}$$

Obviously, $\partial\bar{\Phi}(\alpha,\beta)/(\partial\beta) = 0$ is equivalent to

$$-k_1(\alpha)l_2(\alpha)\beta^2 + 2(k_2(\alpha)l_0(\alpha) - l_2(\alpha)k_0(\alpha))\beta + k_1(\alpha)l_0(\alpha) = 0. \tag{21}$$

Using the definitions of functions $l_i(\alpha)$ and $k_j(\alpha)$ with $i = 0,2$ and $j = 0,1,2$, the Equation (21) can be equivalently rewritten as

$$\begin{aligned}
2(c_4 + 2c_2\alpha + (2c_3 + c_1\alpha)\beta)\Big(&\left(c_3c_4 + (c_1c_4 + 2c_2c_3)\alpha + (2c_1c_2 + c_4n)\alpha^2 + 2c_2n\alpha^3\right)\beta\\
&- \left(2c_3c_5 + (c_1c_5 + 2c_3c_4)\alpha + (c_1c_4 + 2c_2c_3)\alpha^2 + c_1c_2\alpha^3\right)\Big) = 0.
\end{aligned} \tag{22}$$

Because $\alpha \geq 0$, $\beta > 0$, and $c_i > 0$ for $i = 1, \ldots, 5$, the first factor of (22) is positive, that is, $c_4 + 2c_2\alpha + (2c_3 + c_1\alpha)\beta > 0$. Hence, for fixed $\alpha \geq 0$, the root of Equation (22) is

$$\beta = \beta(\alpha) := \frac{2c_3c_5 + (c_1c_5 + 2c_3c_4)\alpha + (c_1c_4 + 2c_2c_3)\alpha^2 + c_1c_2\alpha^3}{c_3c_4 + (c_1c_4 + 2c_2c_3)\alpha + (2c_1c_2 + c_4n)\alpha^2 + 2c_2n\alpha^3}. \tag{23}$$

Using Lemma 1 gives $\beta(\alpha) > 0$. From (20)–(22), we know that $\partial\bar{\Phi}(\alpha,\beta)/\partial\beta > 0$ for $\beta > \beta(\alpha)$, and $\partial\bar{\Phi}(\alpha,\beta)/\partial\beta < 0$ for $\beta < \beta(\alpha)$. Therefore, function $\bar{\Phi}(\alpha,\beta)$ achieves its minimum value on the curve $\beta = \beta(\alpha)$ defined in (23).

Furthermore, substituting $\beta = \beta(\alpha)$ into $\bar{\Phi}(\alpha,\beta)$, we obtain a single-variable function of the form

$$\tilde{\Phi}(\alpha) := \bar{\Phi}(\alpha,\beta(\alpha)) = \frac{k_2(\alpha)\beta^2(\alpha) + k_1(\alpha)\beta(\alpha) + k_0(\alpha)}{4\left(l_2(\alpha)\beta^2(\alpha) + l_0(\alpha)\right)}. \tag{24}$$

From the above analysis, we know that the optimal parameter $\bar{\alpha}_{\text{SNM}}$ defined in (16) is also the minimum point of function $\tilde{\Phi}(\alpha)$, that is,

$$\underset{\alpha\in(0,+\infty)}{\arg\min}\ \tilde{\Phi}(\alpha) = \bar{\alpha}_{\text{SNM}}. \tag{25}$$

From (19) and (23), the other two optimal parameters $\bar{\beta}_{\text{SNM}}$ and $\bar{\zeta}_{\text{SNM}}$ defined in (16) can be written as

$$\bar{\beta}_{\text{SNM}} = \frac{2c_3c_5 + (c_1c_5 + 2c_3c_4)\bar{\alpha}_{\text{SNM}} + (c_1c_4 + 2c_2c_3)\bar{\alpha}_{\text{SNM}}^2 + c_1c_2\bar{\alpha}_{\text{SNM}}^3}{c_3c_4 + (c_1c_4 + 2c_2c_3)\bar{\alpha}_{\text{SNM}} + (2c_1c_2 + c_4n)\bar{\alpha}_{\text{SNM}}^2 + 2c_2n\bar{\alpha}_{\text{SNM}}^3} \tag{26}$$

and

$$\bar{\zeta}_{\text{SNM}} = \frac{c_4 + 2\bar{\alpha}_{\text{SNM}}c_2 + 2\bar{\beta}_{\text{SNM}}c_3 + \bar{\alpha}_{\text{SNM}}\bar{\beta}_{\text{SNM}}c_1}{2n\bar{\alpha}_{\text{SNM}}^2\bar{\beta}_{\text{SNM}}^2 + 2c_2\bar{\alpha}_{\text{SNM}}^2 + 2c_1\bar{\alpha}_{\text{SNM}}\bar{\beta}_{\text{SNM}}^2 + 2c_4\bar{\alpha}_{\text{SNM}} + 2c_3\bar{\beta}_{\text{SNM}}^2 + 2c_5}. \tag{27}$$

Hence, the calculation of the optimal parameters $\bar{\alpha}_{\text{SNM}}, \bar{\beta}_{\text{SNM}}, \bar{\zeta}_{\text{SNM}}$ is reduced to finding the minimum point of function $\tilde{\Phi}(\alpha)$.

In the following, we first show the existence of the minimum point of $\tilde{\Phi}(\alpha)$. Note that $l_0(\alpha) > 0$ and $l_2(\alpha) > 0$; the denominator of $\tilde{\Phi}(\alpha)$ defined in (24) is positive for any $\alpha \geq 0$. Hence, $\tilde{\Phi}(\alpha)$ is derivable on the interval $[0,+\infty)$. Careful calculation gives

$$\tilde{\Phi}'(\alpha) = \frac{d_6\alpha^6 + d_5\alpha^5 + d_4\alpha^4 + d_3\alpha^3 + d_2\alpha^2 + d_1\alpha + d_0}{4\left(c_2\alpha^2 + c_4\alpha + c_5\right)^2\left(n\alpha^2 + c_1\alpha + c_3\right)^2}, \tag{28}$$

where

$$\begin{aligned}
d_0 &= c_3^2 c_4 \left(c_4^2 - 4c_2c_5\right), \\
d_1 &= 2c_1c_3c_4^3 + 2c_2c_3^2c_4^2 + 8nc_3^2c_5^2 - 2c_1^2c_3c_5^2 - 8c_1c_2c_3c_4c_5 - 8c_2^2c_3^2c_5, \\
d_2 &= c_1^2c_4^3 + 4c_1c_2c_3c_4^2 + 4nc_1c_3c_5^2 + 16nc_3^2c_4c_5 + 2nc_3c_4^3 - c_1^3c_5^2 - 4c_1^2c_2c_4c_5 \\
&\quad - 4c_1^2c_3c_4c_5 - 16c_1c_2^2c_3c_5 - 8nc_2c_3c_4c_5, \\
d_3 &= 2c_1^2c_2c_4^2 + 8c_5nc_1c_3c_4 + 2nc_1c_4^3 + 16c_5nc_2c_3^2 + 4nc_2c_3c_4^2 + 8nc_3^2c_4^2 - 2c_5c_1^3c_4 \\
&\quad - 8c_5c_1^2c_2^2 - 4c_5c_1^2c_2c_3 - 2c_1^2c_3c_4^2 - 8c_5nc_1c_2c_4 - 16c_5nc_2^2c_3, \\
d_4 &= 8c_5c_1c_2c_3n + 4c_1c_2c_4^2n + 4c_1c_3c_4^2n + 16c_2c_3^2c_4n + c_4^3n^2 - 4c_5c_2c_4n^2 - 2c_5c_1^3c_2 \\
&\quad - c_1^3c_4^2 - 4c_1^2c_2c_3c_4 - 16c_5c_1c_2^2n, \\
d_5 &= 2c_2\left(4c_1c_3c_4n + c_4^2n^2 + 4c_2c_3^2n - c_1^3c_4 - c_1^2c_2c_3 - 4c_5c_2n^2\right), \\
d_6 &= c_1c_2^2\left(4c_3n - c_1^2\right).
\end{aligned} \tag{29}$$

From Lemma 1, if HS is not a multiple of S, we have $d_0 < 0$ and $d_6 > 0$. Thus, the function $\tilde{\Phi}'(\alpha)$ yields $\tilde{\Phi}'(0) < 0$, and there exists a positive constant $\bar{\tau}$ such that $\tilde{\Phi}'(\alpha) > 0$ for $\alpha \geq \bar{\tau}$. This means the function $\tilde{\Phi}(\alpha)$ achieves its minimum in the interval $(0, \bar{\tau})$. From (25), we know that the optimal parameter $\bar{\alpha}_{\text{SNM}}$ is exactly the minimum point of $\tilde{\Phi}(\alpha)$. Moreover, from (28), the minimum point of $\tilde{\Phi}(\alpha)$ (i.e., the optimal parameter $\bar{\alpha}_{\text{SNM}}$) can be obtained by solving the sextic equation

$$d_6\alpha^6 + d_5\alpha^5 + d_4\alpha^4 + d_3\alpha^3 + d_2\alpha^2 + d_1\alpha + d_0 = 0. \tag{30}$$

When HS is a multiple of S, that is, $HS = \delta S(\delta > 0)$, and H is not a multiple of an identity matrix, we have $c_4 = 2\delta c_2$, $c_5 = \delta^2 c_2$, and $c_4^2 - 4c_2c_5 = 0$. Furthermore, we can derive that $d_0 = 0$ and

$$\begin{aligned}
d_1 &= 2\delta^4c_2^2c_3\left(4nc_3 - c_1^2\right) > 0, & d_2 &= \delta^3c_2^2(\delta c_1 + 8c_3)\left(4nc_3 - c_1^2\right) > 0, \\
d_3 &= 4c_2^2\delta^2(3c_3 + c_1\delta)\left(4c_3n - c_1^2\right) > 0, & d_4 &= 2c_2^2\delta(4c_3 + 3c_1\delta)\left(4c_3n - c_1^2\right) > 0, \\
d_5 &= 2c_2^2(2\delta c_1 + c_3)\left(4c_3n - c_1^2\right) > 0, & d_6 &= c_1c_2^2\left(4c_3n - c_1^2\right) > 0.
\end{aligned}$$

Thus, we have $\tilde{\Phi}'(0) = 0$ and $\tilde{\Phi}'(\alpha) > 0$ for any $\alpha > 0$, which means that the function $\tilde{\Phi}(\alpha)$ achieves its minimum value at $\alpha = 0$. Therefore, the optimal parameter $\bar{\alpha}_{\text{SNM}} = 0$. Substituting it into (26) and (27), we obtained the optimal parameters $\bar{\beta}_{\text{SNM}}$ and $\bar{\zeta}_{\text{SNM}}$, which are

$$\bar{\beta}_{\text{SNM}} = \frac{2c_5}{c_4} = \delta \qquad \text{and} \qquad \bar{\zeta}_{\text{SNM}} = \frac{c_4 + 2kc_3}{2c_5 + 2k^2c_3} = \frac{1}{\delta}.$$

In particular, if matrix H is a multiple of an identity matrix, that is, $H = cI(c > 0)$, we have $c_1 = 2cn$, $c_3 = c^2n$, $c_4 = 2cc_2$, $c_5 = c^2c_2$, $c_4^2 - 4c_2c_5 = 0$, and $4c_3n - c_1^2 = 0$. Substituting these relations into (26), (27), and (29) gives

$$\bar{\beta}_{\text{SNM}} = c, \quad \bar{\zeta}_{\text{SNM}} = \frac{1}{\bar{\alpha}_{\text{SNM}} + c},$$

and $d_i = 0$ for $i = 0, 1, \ldots, 6$. Using (28) gives $\tilde{\Phi}'(\alpha) \equiv 0$, or equivalently, $\tilde{\Phi}(\alpha)$ is a constant. This means that the optimal parameter $\bar{\alpha}_{\text{SNM}}$ can be chosen arbitrarily on the interval $[0, +\infty)$.

The above analysis can be summarized as the following theorem.

Theorem 2. *Let A be a non-Hermitian positive definite matrix. Matrices H and S are, respectively, the Hermitian and the skew-Hermitian parts of matrix A. Then, the optimal parameters $\bar{\alpha}_{\text{SNM}}$, $\bar{\beta}_{\text{SNM}}$, and $\bar{\zeta}_{\text{SNM}}$ of the preconditioner $P(\alpha, \beta, \zeta)$ exist. Moreover,*

1. *If HS is not a multiple of S, the optimal parameter $\bar{\alpha}_{\text{SNM}}$ is the one positive root of* (30)*, which minimizes the function $\tilde{\Phi}(\alpha)$. Using the value of $\bar{\alpha}_{\text{SNM}}$, the optimal parameters $\bar{\beta}_{\text{SNM}}$ and $\bar{\zeta}_{\text{SNM}}$ can be easily obtained according to the formulas* (26) *and* (27)*, respectively.*
2. *If HS is a multiple of S, that is, $HS = \delta S(\delta > 0)$, but H is not a multiple of an identity matrix, the optimal parameters yield $\bar{\alpha}_{\text{SNM}} = 0$, $\bar{\beta}_{\text{SNM}} = \delta$, and $\bar{\zeta}_{\text{SNM}} = 1/\delta$.*
3. *If H is a multiple of an identity matrix, that is, $H = cI(c > 0)$, we have $\bar{\beta}_{\text{SNM}} = c$ and $\bar{\zeta}_{\text{SNM}} = 1/(\bar{\alpha}_{\text{SNM}} + c)$, where $\bar{\alpha}_{\text{SNM}}$ is not unique; it can be chosen arbitrarily on the interval* $[0, +\infty)$.

Remark 3. Because the choice of the parameter ζ does not affect the performance of the TPHSS preconditioner (15), we only need to calculate the optimal parameters $\bar{\alpha}_{\text{SNM}}$ and $\bar{\beta}_{\text{SNM}}$ in the implementation of the TPHSS preconditioner.

Remark 4. When HS is a multiple of S, that is, $HS = \delta S$, but H is *not* a multiple of an identity matrix, we have

$$N(\alpha, \beta, \zeta) = P(\alpha, \beta, \zeta) - A = \alpha\beta\zeta I + (\alpha\zeta - 1)S + (\beta\zeta - 1)H + \zeta\delta S.$$

Substituting the optimal parameters $\bar{\alpha}_{\mathrm{SNM}} = 0$, $\bar{\beta}_{\mathrm{SNM}} = \delta$, and $\bar{\zeta}_{\mathrm{SNM}} = 1/\delta$ into the above equality, the matrix $N(\alpha, \beta, \zeta)$ becomes zero. Similarly, if H is a multiple of an identity matrix, that is, $H = cI(c > 0)$, matrix $N(\alpha, \beta, \zeta)$ becomes

$$N(\alpha, \beta, \zeta) = P(\alpha, \beta, \zeta) - A = ((\alpha + c)\beta\zeta - c)I + ((\alpha + c)\zeta - 1)S.$$

Hence, for any $\bar{\alpha}_{\mathrm{SNM}} \geq 0$, $N(\alpha, \beta, \zeta)$ reduces to zero matrix when $\bar{\beta}_{\mathrm{SNM}} = c$ and $\bar{\zeta}_{\mathrm{SNM}} = 1/(\bar{\alpha}_{\mathrm{SNM}} + c)$. Therefore, the above results verify the correctness of the last two conclusions of Theorem 2.

4 | NUMERICAL EXAMPLES

In this section, we employ three examples to test the performances of the HSS and the TPHSS preconditioners in which the parameters are computed by the SNM method. For comparison, the preconditioning effects of the HSS preconditioners whose parameters are computed by other methods will also be presented.

In the implementations, we use the HSS and the TPHSS preconditioners to accelerate the convergence of the generalized minimum residual (GMRES) method. The initial guesses $x^{(0)}$ for the GMRES and the preconditioned GMRES (PGMRES) methods are chosen as zero vector, and the iterations are terminated once the current iterate $x^{(k)}$ satisfies

$$\frac{\|b - Ax^{(k)}\|_2}{\|b - Ax^{(0)}\|_2} \leq 10^{-6}. \tag{31}$$

In addition, for Examples 1 and 2, the systems of linear equations with matrices $\alpha I + H$ and $\gamma I + S\,(\gamma = \alpha \text{ or } \beta)$ involved in the PGMRES method are solved by direct methods, that is, the Cholesky factorization in combination with the symmetric approximate minimum degree reordering and the LU factorization in combination with the column approximate minimum degree reordering, respectively. All the computations are implemented in MATLAB [version 7.11.0.584 (R2010b)] in double precision on a personal computer with 2.60GHZ central processing unit [Intel(R) Core(TM) i7-4510U] and 8.00GB memory. All the tested preconditioners and their detailed descriptions are listed in Table 1.

When we compute the parameter values of the HSS and the TPHSS preconditioners by the SNM method or the method proposed by Huang,[31] the traces of matrices S^*S, H^2, S^*HS, and $(HS)^*HS$ should be computed. To reduce workload, the traces of S^*S and H^2 are calculated according to the formula

$$\mathrm{tr}(AB) = \sum_{i,j=1}^{n} \left(A \circ B^T\right)_{ij}, \quad A, B \in \mathbb{C}^{n\times n},$$

where $\circ$ denotes the Hadamard product. In this way, the calculation of the matrix product AB is avoided. However, the matrix product HS cannot be avoided when we compute the traces of S^*HS and $(HS)^*HS$.

TABLE 1 The tested preconditioners

Preconditioner	Description
HSS–Opt	HSS preconditioner whose parameter α is chosen as the experimentally found optimal one that leads to the least number of iteration steps of the preconditioned generalized minimum residual method
HSS–BGN	HSS preconditioner whose parameter is defined by Bai et al.,[8] that is, $\alpha_{\mathrm{BGN}} = \sqrt{\lambda_{\max}(H)\lambda_{\min}(H)}$, where $\lambda_{\max}(H)$ and $\lambda_{\min}(H)$ denote the maximum and the minimum eigenvalues of matrix H, respectively
HSS–Huang	HSS preconditioner whose parameter is defined by Huang,[31] that is, $\alpha_{\mathrm{Huang}} = \min_{\alpha>0}\|2\alpha N(\alpha)\|_F$, where $N(\alpha)$ is defined in (4)
HSS–SNM	HSS preconditioner whose parameters $(\alpha_{\mathrm{SNM}}, \zeta_{\mathrm{SNM}})$ are computed by the SNM method
TPHSS–SNM	TPHSS preconditioner whose parameters $(\bar{\alpha}_{\mathrm{SNM}}, \ \bar{\beta}_{\mathrm{SNM}}, \ \bar{\zeta}_{\mathrm{SNM}})$ are computed by the SNM method

Note. HSS = Hermitian and skew-Hermitian splitting; TPHSS = two-parameter HSS; SNM = scaled norm minimization

TABLE 2 The parameter values of the Hermitian and skew-Hermitian splitting and the two-parameter HSS preconditioners for Example 1

		$\alpha_{\mathbf{Opt}}$	$\alpha_{\mathbf{BGN}}$	$\alpha_{\mathbf{Huang}}$	$(\alpha_{\mathbf{SNM}}, \zeta_{\mathbf{SNM}})$	$(\bar{\alpha}_{\mathbf{SNM}}, \bar{\beta}_{\mathbf{SNM}}, \bar{\zeta}_{\mathbf{SNM}})$
$d = 2$	$q = 0.01$	[0.0008, 0.0025]	0.1570	3.09e−9	(0.0350, 28.378)	(2.575e−9, 4.7437, 0.2108)
	$q = 0.1$	[0.0045, 0.0106]	0.1570	3.09e−7	(0.1115, 8.7717)	(2.575e−7, 4.7437, 0.2108)
	$q = 1$	[0.0229, 0.0329]	0.1570	3.09e−5	(0.3606, 2.5805)	(2.575e−5, 4.7437, 0.2108)
	$q = 10$	[0.1063, 0.1959]	0.1570	3.10e−3	(1.2083, 0.6550)	(2.575e−3, 4.7433, 0.2107)
	$q = 100$	[0.6150, 0.9378]	0.1570	0.3524	(3.5483, 0.1545)	(0.2581, 4.7100, 0.2017)
	$q = 1000$	[2.5521, 3.4614]	0.1570	3.9088	(4.9530, 0.1060)	(28.2392, 4.1187, 0.0309)
$d = 3$	$q = 0.01$	[0.0047, 0.0158]	0.7475	3.31e−8	(0.0915, 10.791)	(2.905e−8, 6.8056, 0.1469)
	$q = 0.1$	[0.0210, 0.0748]	0.7475	3.31e−6	(0.2932, 3.2708)	(2.905e−6, 6.8056, 0.1469)
	$q = 1$	[0.1601, 0.1955]	0.7475	3.31e−4	(0.9648, 0.9063)	(2.905e−4, 6.8055, 0.1469)
	$q = 10$	[0.5550, 1.2879]	0.7475	3.33e−2	(3.2459, 0.2045)	(2.905e−2, 6.8023, 0.1464)
	$q = 100$	[2.9121, 4.1946]	0.7475	5.3621	(6.2693, 0.0803)	(2.9742, 6.5702, 0.1051)
	$q = 1000$	[8.4139, 9.5386]	0.7475	5.9853	(9.3386, 0.0631)	(321.287, 6.0175, 0.0031)

Note. SNM = scaled norm minimization

Example 1. Consider the following convection–diffusion equation:

$$\begin{aligned} -\Delta u + q \operatorname{div} u &= f, \text{in} \quad \Omega, \\ u &= 0, \text{on} \quad \partial\Omega, \end{aligned} \tag{32}$$

where $q > 0$ is a constant and $\Omega = [0, 1]^d$ with $d = 2$ and 3. We discretize this problem using the central difference scheme with mesh size $h = 1/l$, where the mesh number is chosen, as $l = 80$ for the 2D case and $l = 25$ for the 3D case.

In Table 2, for both of the 2D and the 3D cases, we list the computed parameter values of the HSS and the TPHSS preconditioners computed by the methods listed in Table 1. When we compute the parameter $\alpha_{\text{BGN}} = \sqrt{\lambda_{\max}(H)\lambda_{\min}(H)}$, we use the power and the inverse power methods, respectively, to compute the maximum and the minimum eigenvalues of the matrix H. Note that the matrix H is independent of the coefficient q, so the parameter α_{BGN} should be also q independent; see Table 2.

For different values of the coefficient q, in Tables 3 and 4, we list the numerical results, including the number of iteration steps (abbreviated as IT) and the elapsed CPU time (in seconds, abbreviated as CPU, and which includes the CPU time for computing the parameter values of the preconditioner), of the PGMRES methods for the 2D and the 3D cases, respectively. Note that the parameter values of the HSS–Opt preconditioner are the experimentally found optimal ones; it is very time consuming to find these parameter values. Therefore, for comparison, we only list the IT of the HSS-Opt PGMRES method.

TABLE 3 Numerical results of the preconditioned generalized minimum residual methods for the 2D case of Example 1

		HSS–Opt	**HSS–BGN**	**HSS–Huang**	**HSS–SNM**	**TPHSS–SNM**
$q = 0.01$	IT	6	27	240	14	2
	CPU	–	2.6520	50.3883	0.8112	0.2652
$q = 0.1$	IT	10	29	351	25	3
	CPU	–	2.3712	81.0737	1.0452	0.2964
$q = 1$	IT	21	32	459	44	5
	CPU	–	2.4024	42.2295	1.7784	0.3900
$q = 10$	IT	46	46	443	69	14
	CPU	–	3.1512	40.4823	2.9016	0.6864
$q = 100$	IT	39	97	52	68	42
	CPU	–	5.6628	2.1840	2.7924	1.7004
$q = 1000$	IT	23	141	24	26	29
	CPU	–	8.6113	1.0608	1.0920	1.1232

Note. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS; IT = iteration steps; CPU = elapsed CPU time

TABLE 4 Numerical results of the preconditioned generalized minimum residual methods for the 3D case of Example 1

		HSS–Opt	HSS–BGN	HSS–Huang	HSS–SNM	TPHSS–SNM
$q = 0.01$	IT	5	18	610	8	2
	CPU	–	12.0589	6,890.3614	5.3664	2.4804
$q = 0.1$	IT	8	18	614	13	3
	CPU	–	10.2649	361.0799	7.1136	2.2776
$q = 1$	IT	14	21	612	23	5
	CPU	–	14.4457	368.4120	11.5441	4.2588
$q = 10$	IT	28	28	210	35	15
	CPU	–	16.2053	108.5767	17.3161	8.4241
$q = 100$	IT	19	45	21	23	23
	CPU	–	24.9602	11.1853	12.5737	12.4645
$q = 1000$	IT	13	50	15	13	11
	CPU	–	27.6122	8.1433	7.8157	6.5988

Note. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS; IT = iteration steps; CPU = elapsed CPU time

From the numerical results listed in Tables 3 and 4, we see that all the iteration sequences generated by the PGMRES methods are convergent to the unique solution of the system of linear equations. The HSS–SNM preconditioner generally is more efficient than the HSS–BGN, especially for large coefficient q. Compared with the HSS–Huang, the HSS–SNM is much more robust and efficient. This is because, for $q = 100, 1000$, the performances of the two preconditioners are comparable; for $q = 0.01, 0.1, 1$, and 10, the performance of the HSS–SNM is much better than that of the HSS–Huang preconditioner. For example, when $q = 1$, the IT and the CPU cost by the HSS–SNM PGMRES method, for the 2D case, are only about one-tenth and a few percent of those of the HSS–Huang PGMRES method, respectively. However, there is still a little gap between the preconditioning effects of the HSS–SNM and the HSS–Opt preconditioners, because the iteration numbers of the HSS–SNM PGMRES method are nearly two times of those of the HSS–Opt PGMRES method for the 2D case and are less than two times for the 3D case. Among all of the tested preconditioners, the TPHSS–SNM is always the most efficient one either in IT or in CPU.

In Figures 1 and 2, the spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices of 2D case are presented for $q = 1$ and 1000, respectively. We can see from the figures that the eigenvalues of the HSS–SNM preconditioned coefficient matrices are more clustered than those of the HSS–BGN and the HSS–Huang preconditioned coefficient matrices. Among the four preconditioners, that is, HSS–BGN, HSS–Huang, HSS–SNM, and

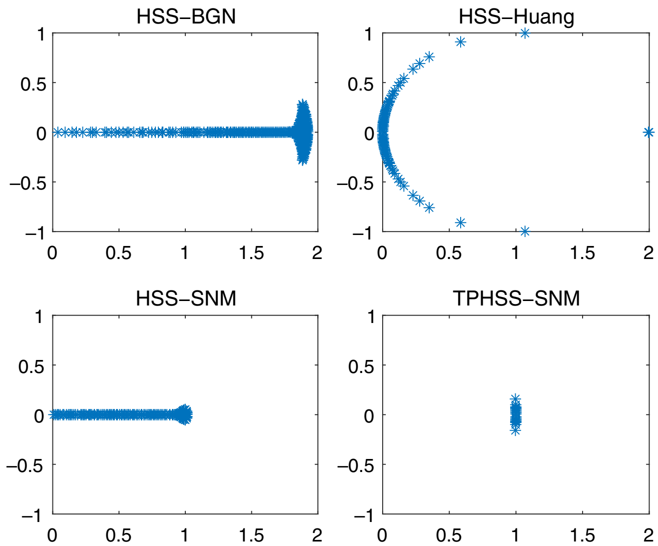

FIGURE 1 Spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the 2D case of Example 1 with $q = 1$ and $l = 80$. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

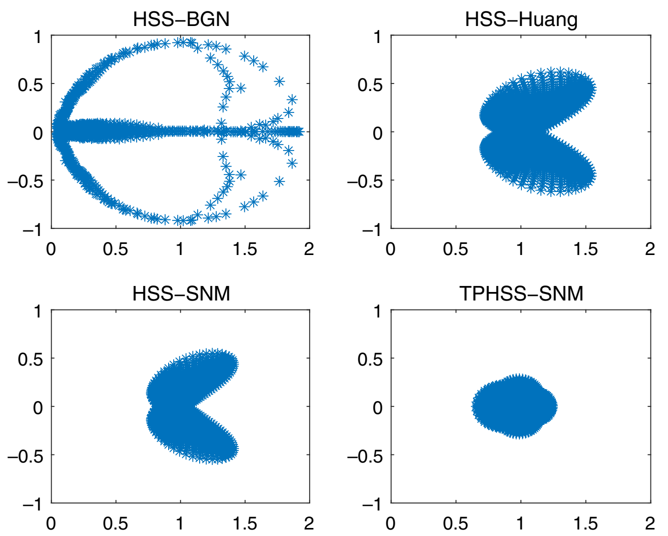

FIGURE 2 Spectral distributions of the Hermitian and skew-Hermitian splitting and the two-parameter HSS preconditioned coefficient matrices for the 2D case of Example 1 with $q = 1000$ and $l = 80$. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

TPHSS–SNM, the eigenvalues of the TPHSS–SNM preconditioned coefficient matrices are the most clustered ones for both of the two cases $q = 1$ and $q = 1000$. This explains that why in Table 3, the TPHSS–SNM PGMRES has the best convergence property.

Example 2. The complex linear system

$$\left(I + \left(1 + \frac{1}{\sqrt{3}}i\right)\frac{\tau}{4}L\right)x = b \tag{33}$$

arises in Padé approximation-type integration schemes for parabolic problems.[33] Here, L is the matrix of a standard five-point discrete operator approximating the 2D negative Laplacian operator

$$Lu(x) = -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2}, \quad x = (x_1, x_2) \in \Omega$$

or of a standard seven-point discrete operator approximating the 3D negative Laplacian operator

$$Lu(x) = -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} - \frac{\partial^2 u}{\partial x_3^2}, \quad x = (x_1, x_2, x_3) \in \Omega$$

with homogeneous Dirichlet boundary conditions on a uniform mesh in the unit square or the unit cube. Parameter $h = 1/l$ is a step size, where l is the mesh number in each direction. In the implementation, we choose $\tau = h$.

In Table 5, we list the parameter values of the HSS and the TPHSS preconditioners computed by different methods for both of the 2D and the 3D problems with different mesh numbers. Based on these parameter values, the HSS and the TPHSS preconditioners in Table 6 are used to accelerate the convergence of the GMRES method. The numerical results show that the HSS–SNM and the HSS–Huang PGMRES methods cost greater iteration numbers

TABLE 5 The parameter values of the Hermitian and skew-Hermitian splitting and the two-parameter HSS preconditioners for Example 2

		α_{Opt}	α_{BGN}	α_{Huang}	$(\alpha_{\text{SNM}}, \zeta_{\text{SNM}})$	$(\tilde{\alpha}_{\text{SNM}}, \tilde{\beta}_{\text{SNM}}, \tilde{\zeta}_{\text{SNM}})$
2D	$l = 32$	[3.0517, 6.8845]	8.6509	31.179	(38.507, 0.0124)	(3.3815, 47.912, 0.0192)
	$l = 64$	[4.9320, 6.3619]	11.784	61.404	(76.245, 0.0062)	(6.7241, 95.270, 0.0097)
	$l = 128$	[6.0868, 8.3842]	16.336	121.862	(151.720, 0.0031)	(13.411, 189.98, 0.0049)
3D	$l = 12$	[3.0615, 6.4789]	7.6618	18.307	(21.197, 0.0227)	(2.6410, 24.693, 0.0360)
	$l = 24$	[3.9368, 6.2203]	9.7509	35.605	(41.648, 0.0115)	(5.2021, 48.932, 0.0182)

Note. SNM = scaled norm minimization

TABLE 6 Numerical results of the preconditioned generalized minimum residual methods for Example 2

			HSS–Opt	HSS–BGN	HSS–Huang	HSS–SNM	TPHSS–SNM
2D	$l = 32$	IT	18	19	29	31	14
		CPU	–	0.1872	0.1872	0.2340	0.0936
	$l = 64$	IT	22	26	44	47	21
		CPU	–	2.1216	1.5912	1.8096	0.7332
	$l = 128$	IT	27	32	64	68	30
		CPU	–	13.2445	11.1853	12.3865	6.1776
3D	$l = 12$	IT	12	13	16	17	10
		CPU	–	0.4524	0.3276	0.3588	0.2184
	$l = 24$	IT	16	18	28	29	15
		CPU	–	21.1693	22.9165	23.2441	13.2913

Note. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS; IT = iteration steps; CPU = elapsed CPU time

than the HSS–BGN and the HSS–Opt PGMRES methods. However, the computing times cost by the HSS–SNM, the HSS–Huang, and the HSS–BGN PGMRES methods are almost the same because the parameter computation of the HSS–BGN preconditioner is a little time consuming. The TPHSS–SNM preconditioner, for this example, is also the most efficient one. It even costs smaller iteration numbers than the optimal case of the HSS (i.e., HSS-Opt) preconditioner.

In Figures 3 and 4, we present the spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the 2D and the 3D problems, respectively. We see from the figures that the HSS–SNM preconditioned coefficient matrices have worse eigenvalue distributions than the HSS–BGN and the HSS–Huang preconditioned coefficient matrices for both of the 2D and the 3D problems. This explains why the HSS–SNM PGMRES method, in Table 6, costs larger number of ITs than the HSS–BGN and the HSS–Huang PGMRES methods. The eigenvalues of the TPHSS–SNM preconditioned coefficient matrices compared with those of the other three preconditioned matrices are the most clustered ones. Therefore, in Table 6, the TPHSS–SNM PGMRES costs the least iteration number and computing time to achieve the same stop criterion.

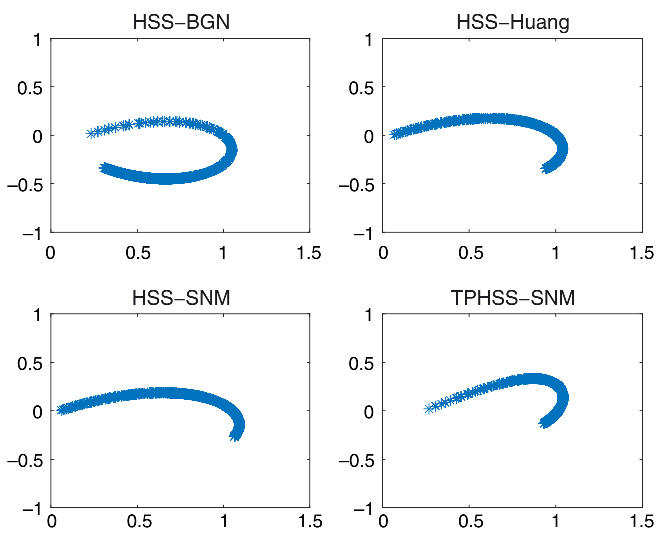

FIGURE 3 Spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the 2D case of Example 2 with $l = 32$. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

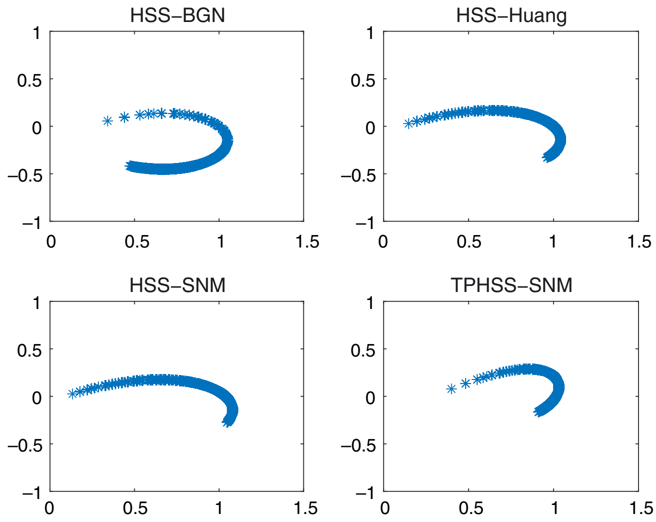

FIGURE 4 Spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the 3D case of Example 2 with $l = 12$. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

Example 3. Consider the system of linear equations of the form

$$\begin{pmatrix} B & E \\ -E^T & \mu I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix},$$

where

$$B = I_3 \otimes \left(I_p \otimes I_p \otimes T + I_p \otimes T \otimes I_p + T \otimes I_p \otimes I_p\right) \in \mathbb{R}^{3p^3 \times 3p^3}$$

and

$$E = \begin{pmatrix} I_p \otimes I_p \otimes F \\ I_p \otimes F \otimes I_p \\ F \otimes I_p \otimes I_p \end{pmatrix} \in \mathbb{R}^{3p^3 \times p^3}.$$

Here, $\otimes$ and I_n denote the Kronecker product symbol and the identity matrix of order n, respectively, and

$$T = \nu \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{p \times p}, \quad F = h \cdot \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{p \times p},$$

where $h = 1/(p + 1)$ denotes the discretization mesh size.

This kind of systems of linear equations with two-by-two block structure arises in many applications, such as discrete Stokes equations,[29] shift-splitting iteration scheme for solving nonsymmetric saddle-point problems,[34] and image reconstruction and restoration.[35] Moreover, this example can be considered as a generalization of example 4.1 in the work of Bai et al.[10] In the computation, we choose $\mu = 1/2$. For this example, simple calculation gives

$$\alpha I + H = \begin{pmatrix} \alpha I + B & 0 \\ 0 & (\alpha + \mu)I \end{pmatrix}$$

and

$$\gamma I + S = \begin{pmatrix} \gamma I & E \\ -E^T & \gamma I \end{pmatrix} = \begin{pmatrix} I & 0 \\ -\frac{1}{\gamma}E^T & I \end{pmatrix} \begin{pmatrix} \gamma I & E \\ 0 & \gamma I + \frac{1}{\gamma}E^T E \end{pmatrix},$$

where $\gamma = \alpha$ for the HSS preconditioner and $\gamma = \beta$ for the TPHSS preconditioner. Thus, in the implementation of the PGMRES method, the solutions of the two systems of linear equations with matrices $\alpha I + H$ and $\gamma I + S$ can be reduced to solving two subsystems with symmetric positive definite coefficient matrices $\alpha I + B$ and $\gamma I + \frac{1}{\gamma}E^T E$. We choose conjugate gradient method to approximate the solutions of the two subsystems.

To test the performances of the HSS–SNM and the TPHSS–SNM preconditioners, we list the optimal parameter values of the HSS and the TPHSS preconditioners computed by different methods in Table 7.

TABLE 7 The parameter values of the HSS and the TPHSS preconditioners for Example 3

		$\alpha_{\mathbf{Opt}}$	$\alpha_{\mathbf{BGN}}$	$\alpha_{\mathbf{Huang}}$	$(\alpha_{\mathbf{SNM}}, \zeta_{\mathbf{SNM}})$	$(\bar{\alpha}_{\mathbf{SNM}}, \bar{\beta}_{\mathbf{SNM}}, \bar{\zeta}_{\mathbf{SNM}})$
$\nu = 1$	$p = 8$	[0.4694, 0.5358]	1.9581	4.17e−3	(1.4246, 0.5648)	(7.53e−3, 7.0891, 0.1409)
	$p = 16$	[0.3203, 0.3824]	1.0884	1.20e−3	(1.0240, 0.8357)	(2.13e−3, 7.1642, 0.1395)
	$p = 32$	[0.1103, 0.2899]	0.5684	3.23e−4	(0.7254, 1.2353)	(5.67e−4, 7.1996, 0.1389)
$\nu = 0.01$	$p = 8$	[0.0268, 0.0514]	4.25e−2	0.1445	(0.2682, 1.7089)	(5.86e−2, 0.4068, 2.0877)
	$p = 16$	[0.0269, 0.0313]	2.26e−2	5.27e−2	(0.2285, 2.2103)	(1.52e−2, 0.4371, 2.1973)
	$p = 32$	[0.0088, 0.0178]	1.17e−2	1.24e−2	(0.1731, 3.3243)	(3.98e−3, 0.4452, 2.2229)

Note. SNM = scaled norm minimization

TABLE 8 Numerical results of the backslash operator and the preconditioned generalized minimum residual methods for Example 3

			\	HSS–Opt	HSS–BGN	HSS–Huang	HSS–SNM	TPHSS–SNM
$\nu = 1$	$p = 8$	IT	–	8	14	73	12	6
		CPU	0.0624	–	0.2028	0.7800	0.0780	0.0780
	$p = 16$	IT	–	10	16	123	15	5
		CPU	1.7004	–	5.0856	23.0569	2.1216	0.9516
	$p = 32$	IT	–	15	20	207	22	4
		CPU	85.6445	–	70.2005	457.0985	32.0426	9.2665
$\nu = 0.01$	$p = 8$	IT	–	21	21	26	30	26
		CPU	0.1092	–	0.2340	0.3744	0.1716	0.1248
	$p = 16$	IT	–	28	29	34	41	21
		CPU	1.6224	–	5.3196	3.3228	3.9468	2.1996
	$p = 32$	IT	–	45	45	45	67	18
		CPU	83.429	–	101.1510	62.2912	84.7709	28.2050

Note. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS; IT = iteration steps; CPU = elapsed CPU time

Based on these optimal parameter values, the HSS and the TPHSS preconditioners are used to accelerate the convergence of the GMRES method; the iteration numbers and the computing times (include the CPU times for computing the parameter values of the preconditioners) of the PGMRES methods are listed in Table 8. For comparison, the computing times of the MATLAB's backslash operator "\" are also listed. The numerical results show that the iteration numbers of the HSS–SNM PGMRES method are nearly 1.5 times of those of the optimal case (i.e., the HSS-Opt) for both of the two cases $\nu = 1$ and $\nu = 0.01$. In addition, the HSS–SNM preconditioner performs better than the HSS–BGN and the HSS–Huang preconditioners when $\nu = 1$. However, its performance becomes a little worse than the HSS–Huang preconditioner for the case $\nu = 0.01$. Satisfyingly, the TPHSS–SNM preconditioner always performs better than all of the tested HSS preconditioners for both of the two cases $\nu = 1$ and 0.01. Furthermore, different from the HSS preconditioners, the iteration numbers cost by the TPHSS-SNM PGMRES method, for this example, is decreasing with the increase of the mesh number (or, equivalently, the problem size). At last, we note that both the HSS–SNM and the TPHSS–SNM PGMRES methods outperform MATLAB's backslash operator, especially for large-scale problems.

When $p = 16$, the spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices are also provided for the two cases $\nu = 1$ and $\nu = 0.01$ in Figures 5 and 6, respectively. From the two figures, we see that the HSS–Huang preconditioned matrix is not well conditioned, because its eigenvalues are loose and some of them are very close to the point $(0, 0)$. Although the eigenvalues of the HSS–SNM preconditioned matrix look more clustered than those of the HSS–BGN, the HSS–SNM preconditioner may perform worse than the HSS–BGN preconditioner. This is also because that the eigenvalues of the HSS–SNM preconditioned matrix are too close to the point $(0, 0)$. At last, we see that the TPHSS–SNM preconditioned matrix is well conditioned because its eigenvalues are tightly clustered.

Finally, owing to the high performances of the HSS–SNM and the TPHSS–SNM preconditioners, we can conclude that the SNM method proposed in this work is an efficient and robust method for computing the parameter values of the HSS and the TPHSS preconditioners.

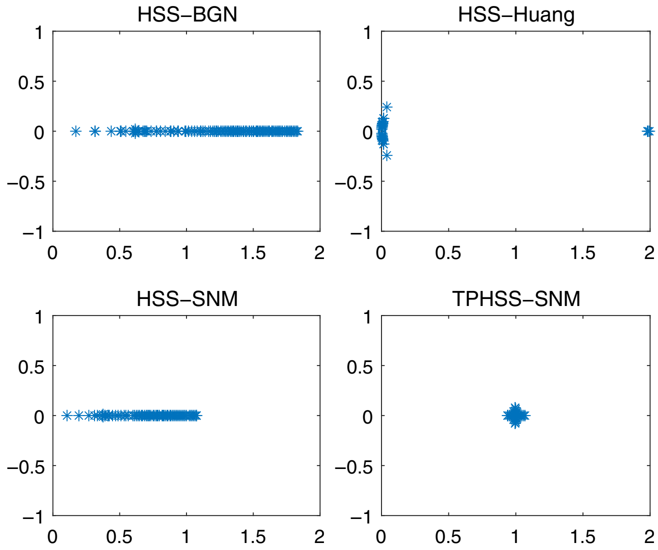

FIGURE 5 Spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the Example 3 with $\nu = 1$ and $p = 16$. HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

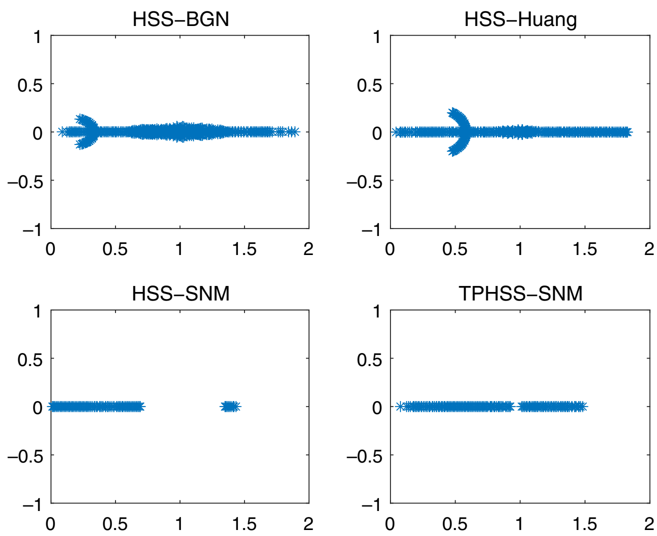

FIGURE 6 Spectral distributions of the HSS and the TPHSS preconditioned coefficient matrices for the Example 3 with $\nu = 0.01$ and $p = 16$.HSS = Hermitian and skew-Hermitian splitting; SNM = scaled norm minimization; TPHSS = two-parameter HSS

5 | CONCLUSIONS

For the non-Hermitian positive definite system of linear equations, in this work, we introduced the SNM method to determine the parameter value of the HSS preconditioner. In addition, by introducing one more parameter for the HSS preconditioner, a TPHSS preconditioner is proposed. The parameter values of this new preconditioner are also determined by the SNM method. Numerical results show that the HSS and the TPHSS preconditioners, whose parameter values are computed by the SNM method, are very efficient. Moreover, the performance of the TPHSS preconditioner is much better than those of the HSS preconditioners whose parameter values are computed by the methods proposed by Huang[31] and Bai et al.[8]

In this work, we only use the SNM method to compute the parameter values of the HSS and the TPHSS preconditioners used for preconditioning the non-Hermitian positive definite system of linear equations. Whether the SNM method still works for the two preconditioners used for preconditioning the non-Hermitian positive semidefinite system of linear equations, such as the saddle-point problem, is also an interesting topic.

ACKNOWLEDGEMENTS

This work is finished when the author's visiting The Academy of Mathematics and System Sciences, Chinese Academy of Sciences. The author is very much indebted to the referees for their constructive and valuable comments and suggestions that greatly improved the original manuscript of this paper.

CONFLICT OF INTEREST

The author declares no potential conflict of interests.

ORCID

Ai-Li Yang http://orcid.org/0000-0002-0185-6818

REFERENCES

1. Bai Z-Z, Rozložník M. On the numerical behavior of matrix splitting iteration methods for solving linear systems. SIAM J Numer Anal. 2015;53(4):1716–1737.
2. Golub GH, Van Loan CF. Matrix computations. 3rd ed. Baltimore, MD: Johns Hopkins University Press; 1996.
3. Wilkinson JH. Rounding errors in algebraic processes. New York, NY: Dover Publications, Incorporated; 1994.
4. Saad Y. Iterative methods for sparse linear systems. 2nd ed. Philadelphia, PA: SIAM; 2003.
5. Bai Z-Z. Motivations and realizations of Krylov subspace methods for large sparse linear systems. J Comput Appl Math. 2015;283:71–78.
6. Erlangga YA, Nabben R. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. SIAM J Matrix Anal Appl. 2008;30(2):684–699.
7. Bai Z-Z, Yin J-F, Su Y-F. A shift-splitting preconditioner for non-Hermitian positive definite matrices. J Comput Math. 2006;24(4):539–552.
8. Bai Z-Z, Golub GH, Ng MK. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J Matrix Anal Appl. 2003;24(3):603–626.
9. Bai Z-Z, Golub GH. Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J Numer Anal. 2007;27(1):1–23.
10. Bai Z-Z, Golub GH, Pan J-Y. Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer Math. 2004;98(1):1–32.
11. Bai Z-Z. Optimal parameters in the HSS-like methods for saddle-point problems. Numer Linear Algebra Appl. 2009;16(6):447–479.
12. Benzi M, Gander M, Golub GH. Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems. BIT Numer Math. 2003;43(5):881–900.
13. Bai Z-Z, Golub GH, Li C-K. Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. Math Comput. 2007;76(257):287–298.
14. Simoncini V, Benzi M. Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems. SIAM J Matrix Anal Appl. 2004;26(2):377–389.
15. Yang A-L, Wu Y-J. The Uzawa-HSS method for saddle-point problems. Appl Math Lett. 2014;38:38–42.
16. Bai Z-Z, Benzi M, Chen F. Modified HSS iteration methods for a class of complex symmetric linear systems. Computing. 2010;87(3–4):93–111.
17. Bai Z-Z, Benzi M, Chen F. On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer Algoritm. 2011;56(2):297–317.
18. Bai Z-Z, Benzi M, Chen F, Wang Z-Q. Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA J Numer Anal. 2013;33(1):343–369.
19. Gu G-D. HSS method with a complex parameter for the solution of complex linear system. J Comput Math. 2011;29(4):441–457.
20. Guo X-X, Wang S. Modified HSS iteration methods for a class of non-Hermitian positive-definite linear systems. Appl Math Comput. 2012;218(20):10122–10128.
21. Bai Z-Z. On semi-convergence of Hermitian and skew-Hermitian splitting methods for singular linear systems. Computing. 2010;89(3–4):171–197.
22. Li W, Liu Y-P, Peng X-F. The generalized HSS method for solving singular linear systems. J Comput Appl Math. 2012;236(9):2338–2353.
23. Guo X-P, Duff IS. Semilocal and global convergence of the Newton-HSS method for systems of nonlinear equations. Numer Linear Algebra Appl. 2011;18(3):299–315.
24. Salkuyeh DK. The Picard–HSS iteration method for absolute value equations. Optim Lett. 2014;8(8):2191–2202.
25. Benzi M, Golub GH. A preconditioner for generalized saddle point problems. SIAM J Matrix Anal Appl. 2005;26(1):20–41.
26. Cao Y, Yao L-Q, Jiang M-Q, Niu Q. A relaxed HSS preconditioner for saddle point problems from meshfree discretization. J Comput Math. 2013;31(4):398–421.
27. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. Acta Numer. 2005;14:1–137.

28. Shen S-Q, Huang T-Z, Yu J. Eigenvalue estimates for preconditioned nonsymmetric saddle point matrices. SIAM J Matrix Anal Appl. 2010;31(5):2453–2476.
29. Bai Z-Z, Golub GH, Li C-K. Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. SIAM J Sci Comput. 2006;28(2):583–603.
30. Chen F. On choices of iteration parameter in HSS method. Appl Math Comput. 2015;271:832–837.
31. Huang Y-M. A practical formula for computing optimal parameters in the HSS iteration methods. J Comput Appl Math. 2014;255:142–149.
32. Yang A-L, An J, Wu Y-J. A generalized preconditioned HSS method for non-Hermitian positive definite linear systems. Appl Math Comput. 2010;216(6):1715–1722.
33. Axelsson O, Kucherov A. Real valued iterative methods for solving complex symmetric linear systems. Numer Linear Algebra Appl. 2000;7(4):197–218.
34. Cao Y, Li S, Yao L-Q. A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems. Appl Math Lett. 2015;49:20–27.
35. Chan LC, Ng MK, Tsing NK. Spectral analysis for HSS preconditioners. Numer Math Theory Methods Appl. 2008;1(1):57–77.

How to cite this article: Yang A-L. Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners. *Numer Linear Algebra Appl*. 2018;e2169. https://doi.org/10.1002/nla.2169