Accepted Manuscript

The generalized double steps scale-SOR iteration method for solving complex symmetric linear systems

Zheng-Ge Huang, Li-Gong Wang, Zhong Xu, Jing-Jing Cui

PII: S0377-0427(18)30442-4
DOI: https://doi.org/10.1016/j.cam.2018.07.022
Reference: CAM 11808

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 5 January 2018
Revised date : 25 April 2018

Please cite this article as: Z.-G. Huang, L.-G. Wang, Z. Xu, J.-J. Cui, The generalized double steps scale-SOR iteration method for solving complex symmetric linear systems, *Journal of Computational and Applied Mathematics* (2018), https://doi.org/10.1016/j.cam.2018.07.022

The generalized double steps scale-SOR iteration method for solving complex symmetric linear systems☆

Zheng-Ge Huang, Li-Gong Wang*, Zhong Xu, Jing-Jing Cui

Department of Applied Mathematics, School of Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China

Abstract

By utilizing the successive-overrelaxation (SOR) acceleration technique for the generalized version of the double-step scale (DSS) iteration method, we construct the generalized DSS-SOR (GDSSOR) iteration method for solving a class of complex symmetric linear systems. The convergence theory of the GDSSOR iteration method is established and its optimal parameters are investigated. Meanwhile, a practical way to choose iteration parameters for the GDSSOR iteration method is developed. Inexact version of the GDSSOR iteration (IGDSSOR) method and its convergence properties are also presented. Numerical experiments illustrate that both GDSSOR and IGDSSOR iteration methods are feasible and effective for solving the complex symmetric linear systems, and perform better than some other commonly used iteration methods.

Keywords: Complex symmetric linear systems, SOR acceleration technique, Convergence properties, Optimal parameters, Inexact implementation
2010 MSC: 65F10, 65F50

1. Introduction

We consider the iterative solution of systems of linear equations of the form:

$$Ax \equiv (W + iT)x = b, \tag{1}$$

where $W, T \in \mathbb{R}^{n\times n}$ are symmetric matrices, with W being positive definite and T positive semi-definite. Throughout this paper, the right-hand side vector $b \in \mathbb{R}^n$ is given and $i = \sqrt{-1}$ denotes the imaginary unit. Here we assume $T \neq 0$, which implies that the matrix A in (1) is non-Hermitian.

The complex symmetric linear systems of this kind are important and widely arise in a variety of scientific computing and engineering applications such as eddy current problem [4], diffuse optical tomography [1], FFT-based solution of certain time-dependent PDEs [14], molecular scattering [25], and electrical power modeling [19]. For more examples and additional references, we refer to [7, 8, 31].

The Hermitian and skew-Hermitian parts of the complex symmetric matrix A can be shown by

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

respectively, hence, the matrix $A \in \mathbb{C}^{n\times n}$ in (1) is a non-Hermitian, but positive definite matrix. Here, A^* denotes the conjugate transpose of the matrix A. To compute the approximate solution of (1) effectively, based on the Hermitian and skew-Hermitian splitting (HSS) of the matrix A in (1): $A = H + S$, Bai et al. [10] first constructed the HSS iteration method with the scheme:

The HSS iteration method: Let $\alpha > 0$ be a positive constant. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

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

However, it follows from (2) that at each iteration step of the HSS iteration method, a shift skew-Hermitian linear system needs to be solved, which necessitates the use of complex arithmetic. To avoid this problem, Bai

☆This research was supported by the National Natural Science Foundation of China (No. 11171273) and Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (No. CX201628).
*Corresponding author.
Email addresses: ZhenggeHuang@mail.nwpu.edu.cn (Zheng-Ge Huang), lgwang@nwpu.edu.cn (Li-Gong Wang), zhongxu@nwpu.edu.cn (Zhong Xu), JingjingCui@mail.nwpu.edu.cn (Jing-Jing Cui)

et al. [7] skillfully designed a modified HSS (MHSS) method which is much more efficient than the HSS one for solving the complex symmetric linear system (1). To further generalize the MHSS iteration method and accelerate its convergence rate, a preconditioned variant of the MHSS method named as the PMHSS iteration method was proposed by Bai et al. [8]. The form of the PMHSS iteration method is as follows.

The PMHSS iteration method: Let $\alpha > 0$ be a positive constant and $V \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha V + W)x^{(k+\frac{1}{2})} = (\alpha V - iT)x^{(k)} + b, \\ (\alpha V + T)x^{(k+1)} = (\alpha V - iW)x^{(k+\frac{1}{2})} - ib. \end{cases} \tag{3}$$

The MHSS and the PMHSS iteration methods have attracted many researchers' attentions due to their good properties, for example, both of them are unconditionally convergent. In order to improve the efficiency of the PMHSS iteration method, many kinds of iteration methods for the complex symmetric linear system (1) have been derived recently. By introducing two different parameters α and β in the PMHSS scheme, Dehghan et al. [16] presented the generalized PMHSS (GPMHSS) method, which has the PMHSS one as its special case. Based on the HSS iteration method, Li et al. [23] established the lopsided HSS (LHSS) iteration method for (1). By combining the PMHSS iteration method with the LHSS iteration method, Li et al. [24] put forward the lopsided PMHSS (LPMHSS) iteration method which outperforms the PMHSS one when the real part of A is dominant. Further, to avoid the complex arithmetic in the LHSS iteration method, Pour and Goughery [26] proposed the new HSS (NHSS) iteration method for solving the non-Hermitian positive definite linear systems. Recently, Hezari et al. [18] designed a scale-splitting (SCSP) iteration method by multiplying a complex number $(\alpha - i)$ through both sides of the complex system (1) and proved that it is convergent to the unique solution of the linear system (1) for a loose restriction on the iteration parameter α. Inspired by the ideas of the NHSS and the SCSP iteration methods, a parameterized variant of NHSS (PNHSS) iteration method was developed by Xiao and Yin [36], and they newly further generalized the PNHSS iteration method and designed the preconditioned PNHSS (PPNHSS) iteration method [35] as follows:

The PPNHSS iteration method: Let $\alpha, \omega > 0$ be positive constants and $V \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\omega W + T)x^{(k+\frac{1}{2})} = -i(\omega T - W)x^{(k)} + (\omega - i)b, \\ (\alpha V + \omega W + T)x^{(k+1)} = [\alpha V - i(\omega T - W)]x^{(k+\frac{1}{2})} + (\omega - i)b. \end{cases} \tag{4}$$

Alternatively, by combining real and imaginary parts of A in (1), Wang et al. [30] derived the combination method of real part and imaginary part which is simply called the CRI iteration method, and proved that the upper bound of the spectral radius of the CRI iteration matrix is smaller than that of the PMHSS one. Subsequently, motivated by the ideas of symmetry of the PMHSS method and the technique of scaling to reconstruct complex linear system (1), Zheng et al. [40] presented a double-step scale splitting (DSS) iteration method, and deduced its convergence theory and optimal parameter. The iterative scheme of the DSS iteration method is as follows.

The DSS method: Let $\alpha > 0$ be a positive constant. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha W + T)x^{(k+\frac{1}{2})} = i(W - \alpha T)x^{(k)} + (\alpha - i)b, \\ (\alpha T + W)x^{(k+1)} = i(\alpha W - T)x^{(k+\frac{1}{2})} + (1 - \alpha i)b. \end{cases} \tag{5}$$

In [3], Bai designed the skew-normal splitting (SNS) to solve the non-Hermitian positive definite systems. The SNS method for solving (1) can be described as following.

The SNS method: Let $\alpha > 0$ be a positive constant. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha I - iT)x^{(k+\frac{1}{2})} = (\alpha W - T^2)x^{(k)} - iTb, \\ (\alpha W + T^2)x^{(k+1)} = i(\alpha I + iT)x^{(k+\frac{1}{2})} - iTb. \end{cases} \tag{6}$$

After that, Wu [31] multiplied W on (1) from the left and considered the Hermitian normal splitting (HNS) method. Similar to the SNS iteration method, Pourbagher and Salkuyeh [27] multiplied both sides of (1) by iT to obtain a modification of the SNS (MSNS) iteration method:

The MSNS method: Let $\alpha > 0$ be a positive constant. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha I + T)x^{(k+\frac{1}{2})} = (i\alpha W + T^2)x^{(k)} + iTb, \\ (i\alpha W - T^2)x^{(k+1)} = (\alpha I - T)x^{(k+\frac{1}{2})} + iTb. \end{cases} \tag{7}$$

The aforementioned iteration methods are two-step methods except for the SCSP one. Now we review some single-step methods for the complex symmetric linear systems. Recently, by applying the scale technique to the

single-step HSS (SHSS) iteration method put forward by Li and Wu [22], Zeng and Ma [39] derived the parameterized SHSS (PSHSS) iteration method. Then Xiao et al. [34] established a generalized version of the PSHSS iteration method, called as the parameterized single-step preconditioned variant of HSS (PSPHSS) iteration method. Very recently, Xiao and Wang [33] introduced a new single-step iteration method referred to as the parameterized variant of the fixed-point iteration adding the asymmetric error (PFPAE) iteration method for solving (1). They theoretically studied the convergence properties of the PFPAE iteration method and derived its quasi-optimal parameters.

The PFPAE iteration method: Let $\alpha > 0$ and $\omega > 0$ be two positive constants. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$(\omega W + T)x^{(k+\frac{1}{2})} = [(1-\alpha)(\omega W + T) - i\alpha(\omega T - W)]x^{(k)} + \alpha(\omega - i)b. \tag{8}$$

In order to construct a fast and stably convergent iteration method for solving the linear system (1), in this work, we first combine the two-parameter acceleration technique used in [9, 13] with the general two-step strategy and theory applied in [2, 5], and establish the generalized DSS (GDSS) iteration method which reduces to the DSS one as $\alpha = \beta$. On the basis of the GDSS iteration method and motivated by the idea of successive-overrelaxation (SOR) acceleration proposed in [11], we then design the generalized double steps scale-SOR (GDSSOR) iteration method by adopting the SOR acceleration technique for the GDSS one, and its inexact version is also discussed. It is proved that the GDSSOR and the inexact GDSSOR (IGDSSOR) iteration methods are convergent to the unique solution of the linear system (1) under proper conditions. In addition, we give a practical way for the choice of parameters of the GDSSOR iteration method by making use of the approximate solution strategy.

The outline of this paper is organized as follows. In Section 2, the GDSSOR iteration method is introduced and established. In Section 3, the convergence properties of the GDSSOR iteration method are discussed together with its parameter regions. In Section 4, we propose the choice for parameters of the GDSSOR iteration method which includes the optimal parameter and the practical way of choosing iteration parameters. The inexact GDSSOR iteration method is studied in Section 5 and its convergence properties are also investigated. Section 6 is devoted to some numerical experiments to examine the feasibility and effectiveness of the GDSSOR and IGDSSOR iteration methods for solving the linear system (1). Finally, brief conclusions are made in Section 7.

We end this section with an introduction of some notations that will be used in the subsequent analysis. For a square matrix H, we indicate its trace by $tr(H)$, its spectrum and the spectral radius by $\sigma(H)$ and $\rho(H)$, respectively. $\|G\|_2$, $\|G\|_F$ and $\kappa(G) = \|G^{-1}\|_2\|G\|_2$ stand for the Euclidean norm, the Frobenius norm and the spectral condition number of the matrix G, respectively. Moreover, $\mathrm{diag}(a_1, a_2, \ldots, a_n)$ denotes a diagonal matrix with diagonal elements a_1, a_2, $\ldots$, a_n, and $J = \mathrm{tridiag}(a, b, c)$ represents a tridiagonal matrix with $J_{i+1,i} = a$, $J_{i,i} = b$ and $J_{i,i+1} = c$.

2. The generalized double steps scale-SOR (GDSSOR) iteration method

In this section, we first establish a new two-step iteration method called the generalized double steps scale (GDSS) iteration method. Moreover, enlightened by the idea of [11], we further present a successive-overrelaxation (SOR) acceleration scheme for the GDSS iteration, which yields the generalized double steps scale-SOR (GDSSOR) iteration method.

We first equivalently rewrite the linear system (1) as

$$(\alpha - i)A = (\alpha W + T) + i(\alpha T - W)x = (\alpha - i)b \tag{9}$$

by multiplying the complex number $(\alpha - i)$ with $\alpha > 0$ through both sides of (1). Similarly, premultiplying the complex system (1) with another complex number $(1 - \beta i)$ with $\beta > 0$ yields that

$$(1 - \beta i)A = [(\beta T + W) + i(T - \beta W)]x = (1 - \beta i)b. \tag{10}$$

Equations (9) and (10) come essentially from [7, 8] where Bai et al. introduced the scale technique for the complex symmetric linear system (1). Now, by alternately iterating between the two systems of fixed-point equations (9) and (10), we can establish the following generalized double steps scale iteration method or, in brief, the GDSS iteration method for solving the complex symmetric linear system (1).

The generalized double steps scale (GDSS) iteration method: Let α and β be two positive constants. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \ldots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha W + T)x^{(k+\frac{1}{2})} = i(W - \alpha T)x^{(k)} + (\alpha - i)b, \\ (\beta T + W)x^{(k+1)} = i(\beta W - T)x^{(k+\frac{1}{2})} + (1 - \beta i)b. \end{cases} \tag{11}$$

The iteration matrix of the GDSS iteration method is $M(\alpha,\beta)=(\beta T+W)^{-1}(T-\beta W)(\alpha W+T)^{-1}(W-\alpha T)$.

The GDSS iteration method reduces to the DSS one as $\alpha=\beta$. In the subsequence, we establish the SOR acceleration scheme and the associated theory mainly for the GDSS iteration. To this end, we first investigate the following sufficient conditions for guaranteeing the convergence of the GDSS iteration method.

Theorem 2.1. *Let $W,T\in\mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively. When $\mu_{\min}>0$, if the parameters α and β satisfy $\alpha\le\beta\le\alpha+2\mu_{\min}$ or $\beta<\alpha\le\frac{2(\beta^2+1)}{\mu_{\max}}+\beta$, then the GDSS iteration method is convergent; when $\mu_{\min}=0$, the GDSS iteration method is convergent if the parameters α and β satisfy $\beta<\alpha\le\frac{2(\beta^2+1)}{\mu_{\max}}+\beta$. Here, $\mu_{\max}$ and $\mu_{\min}$ are the maximum and minimum eigenvalues of the matrix $W^{-1}T$, respectively.*

Proof. We will study the conditions of α and β such that $\rho(M(\alpha,\beta))<1$. By the similarity invariance of the matrix spectrum, we have

$$\begin{aligned}\rho(M(\alpha,\beta)) &= \rho((\beta T+W)^{-1}(\beta W-T)(\alpha W+T)^{-1}(W-\alpha T))\\ &= \rho((\beta S+I)^{-1}(\beta I-S)(\alpha I+S)^{-1}(I-\alpha S))\\ &= \max_i\left|\frac{(\beta-\mu_i)(1-\alpha\mu_i)}{(1+\beta\mu_i)(\alpha+\mu_i)}\right|,\end{aligned}$$

where $S=W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$ and μ_i is an eigenvalue of the matrix S. To get $\rho(M(\alpha,\beta))<1$, it is enough to have

$$\left|\frac{(\beta-\mu_i)(1-\alpha\mu_i)}{(1+\beta\mu_i)(\alpha+\mu_i)}\right|<1,\ \text{for } 1\le i\le n,$$

which is equivalent to

$$(\alpha+\beta)(1+\mu_i^2)>0 \text{ and } (\beta-\alpha)\mu_i^2+2(\alpha\beta+1)\mu_i+(\alpha-\beta)>0. \tag{12}$$

If $\mu_{\min}>0$, then the convergence region of the GDSS iteration method follows in two different cases:

- $\beta\ge\alpha$. If $\alpha=\beta$, then (12) is valid for $\alpha,\beta>0$. If $\beta>\alpha$, then from the second inequality of (12) we obtain

$$\mu_i>\frac{-(\alpha\beta+1)+\sqrt{(\alpha^2+1)(\beta^2+1)}}{\beta-\alpha}. \tag{13}$$

  Since

$$\frac{-(\alpha\beta+1)+\sqrt{(\alpha^2+1)(\beta^2+1)}}{\beta-\alpha}<\frac{-2(\alpha\beta+1)+\alpha^2+\beta^2+2}{2(\beta-\alpha)}$$

  under the condition $\beta>\alpha$, it follows that (13) holds if

$$\mu_i\ge\frac{-2(\alpha\beta+1)+\alpha^2+\beta^2+2}{2(\beta-\alpha)}.$$

  This shows that $\beta\le\alpha+2\mu_{\min}$, which together with $\beta\ge\alpha$ gives $\alpha\le\beta\le\alpha+2\mu_{\min}$.

- $\beta<\alpha$. Solving the second inequality of (12) yields that

$$\mu_i<\frac{\alpha\beta+1+\sqrt{(\alpha^2+1)(\beta^2+1)}}{\alpha-\beta}. \tag{14}$$

  Note that

$$\frac{\alpha\beta+1+\sqrt{(\alpha^2+1)(\beta^2+1)}}{\alpha-\beta}>\frac{\beta^2+1+\beta^2+1}{\alpha-\beta}=\frac{2(\beta^2+1)}{\alpha-\beta}$$

  under the condition $\beta<\alpha$, thereby, (14) holds if

$$\mu_{\max}(\alpha-\beta)\le 2(\beta^2+1),$$

  from which we can deduce that $\beta<\alpha\le\frac{2(\beta^2+1)}{\mu_{\max}}+\beta$ in terms of the condition $\beta<\alpha$.

If $\mu_{\min}=0$, then $(\beta-\alpha)\mu_{\min}^2+2(\alpha\beta+1)\mu_{\min}+(\alpha-\beta)>0$ is invalid as $\beta\geq\alpha$. Thus we only consider $\beta<\alpha$ for this case. Similar to the above discussions, it can be seen that if $\beta<\alpha\leq\frac{2(\beta^2+1)}{\mu_{\max}}+\beta$, the GDSS iteration method is convergent. ■

Remark 2.1. Due to the fact that we are not easy to determine whether $\mu_{\min}$ is equal to 0 or not, the parameters α and β can be chosen to satisfy $\beta<\alpha\leq\frac{2(\beta^2+1)}{\mu_{\max}}+\beta$ in practical calculations.

From the iteration scheme (11) of the GDSS iteration method we can obtain the fixed-point equations

$$\begin{cases}(\alpha W+T)x=i(W-\alpha T)y+(\alpha-i)b,\\(\beta T+W)y=i(\beta W-T)x+(1-\beta i)b.\end{cases}\tag{15}$$

These two fixed-point equations have the following equivalence relationships with the original system of linear equations (1).

Theorem 2.2. *If x^* is the exact solution of equation (1), then it is also the exact solution of equations (15), and vice versa.*

Proof. (15) directly leads to

$$\begin{cases}\dfrac{1}{\alpha-i}(\alpha W+T)x=\dfrac{i}{\alpha-i}(W-\alpha T)y+b,\\[2mm]\dfrac{1}{1-\beta i}(\beta T+W)y=\dfrac{i}{1-\beta i}(\beta W-T)x+b.\end{cases}\tag{16}$$

Subtracting the first equation of (16) from the second equation of (16) results in

$$[(1-\beta i)\alpha+i(\alpha-i)\beta]Wx+[(1-\beta i)-i(\alpha-i)]Tx=[i(1-\beta i)+(\alpha-i)]Wy+[(\alpha-i)\beta-\alpha i(1-\beta i)]Ty,$$

which can be simplified as

$$(\alpha+\beta)(W-iT)(x-y)=0.$$

Taking into account $\alpha,\beta>0$ and the matrix $W-iT$ is nonsingular, it has $x=y$. Thus

$$\begin{cases}(W+iT)x=b,\\(W+iT)x=b,\end{cases}$$

could be derived from (16) directly, which implies that (1) and (15) have the same exact solutions. ■

(15) can be rewritten in the form

$$\bar{A}u=\begin{pmatrix}\alpha W+T & -i(W-\alpha T)\\ -i(\beta W-T) & \beta T+W\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}(\alpha-i)b\\(1-\beta i)b\end{pmatrix}=\bar{b}.\tag{17}$$

Theorem 2.1 immediately implies the following fact.

Theorem 2.3. *Let $W,T\in\mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively, and the parameters α and β satisfy the conditions of Theorem 2.1. Then the matrix $\bar{A}$ defined as in (17) is nonsingular.*

Proof. According to the conditions of this theorem, we infer that $\alpha W+T$ is symmetric positive definite, then

$$\begin{aligned}\bar{A}&=\begin{pmatrix}\alpha W+T & -i(W-\alpha T)\\ -i(\beta W-T) & \beta T+W\end{pmatrix}\\&=\begin{pmatrix}I & 0\\ -i(\beta W-T)(\alpha W+T)^{-1} & I\end{pmatrix}\begin{pmatrix}\alpha W+T & -i(W-\alpha T)\\ 0 & G(\alpha,\beta)\end{pmatrix},\end{aligned}$$

where

$$\begin{aligned}G(\alpha,\beta)&=(\beta T+W)+(\beta W-T)(\alpha W+T)^{-1}(W-\alpha T),\\&=(\beta T+W)[I+(\beta T+W)^{-1}(\beta W-T)(\alpha W+T)^{-1}(W-\alpha T)]\\&=(\beta T+W)[I-M(\alpha,\beta)].\end{aligned}$$

Since α and β satisfy the conditions of Theorem 2.1, we have $\rho(M(\alpha,\beta))<1$ by Theorem 2.1. Having in mind that $\beta T+W$ is symmetric positive definite, thus the conclusion of this theorem follows. ■

By combining the proof of Theorems 2.2 and 2.3, the following theorem is obtained immediately.

Theorem 2.4. *Assume that the conditions of Theorem 2.3 are satisfied. If x^* is the exact solution of (1), then $u^* = (x^*; x^*)$ is the exact solution of the linear system (17). Conversely, if $u^* = (x^*; y^*)$ is the exact solution of the linear system (17), then it must hold $x^* = y^*$ and x^* is the exact solution of (1).*

Applying the block Jacobi iteration for (15), or for the block 2×2 linear system (17), we have

$$\begin{pmatrix} \alpha W + T & 0 \\ 0 & \beta T + W \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & i(W - \alpha T) \\ i(\beta W - T) & 0 \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \begin{pmatrix} (\alpha - i)b \\ (1 - \beta i)b \end{pmatrix},$$

or equivalently,

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = J(\alpha, \beta) \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \begin{pmatrix} \alpha W + T & 0 \\ 0 & \beta T + W \end{pmatrix}^{-1} \begin{pmatrix} (\alpha - i)b \\ (1 - \beta i)b \end{pmatrix},$$

where

$$J(\alpha, \beta) = \begin{pmatrix} 0 & i(\alpha W + T)^{-1}(W - \alpha T) \\ i(\beta T + W)^{-1}(\beta W - T) & 0 \end{pmatrix}. \tag{18}$$

Now, we consider the block SOR iteration for (15) in the following:

$$\begin{pmatrix} \alpha W + T & 0 \\ -\omega i(\beta W - T) & \beta T + W \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \begin{pmatrix} (1-\omega)(\alpha W + T) & \omega i(W - \alpha T) \\ 0 & (1-\omega)(\beta T + W) \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \omega \bar{b}, \tag{19}$$

which can be regarded as the SOR acceleration of the GDSS iteration method. More precisely, we have the following algorithmic description of the GDS-SOR (GDSSOR) iteration method.

The GDS-SOR (GDSSOR) iteration method: Let α, β and ω be three positive constants. Given an initial vectors $x^{(0)}$ and $y^{(0)}$. For $k = 0, 1, 2, \ldots$, until the iteration sequence $\{((x^{(k)})^*, (y^{(k)})^*)^*\}$ converges, compute

$$\begin{cases} (\alpha W + T)x^{(k+1)} = (1-\omega)(\alpha W + T)x^{(k)} + \omega[i(W - \alpha T)y^{(k)} + (\alpha - i)b], \\ (\beta T + W)y^{(k+1)} = (1-\omega)(\beta T + W)y^{(k)} + \omega[i(\beta W - T)x^{(k+1)} + (1 - \beta i)b]. \end{cases} \tag{20}$$

The GDSSOR iteration method can be reformulated as

$$x^{(k+1)} = T(\alpha, \beta, \omega)x^{(k)} + \omega \begin{pmatrix} \alpha W + T & 0 \\ -\omega i(\beta W - T) & \beta T + W \end{pmatrix}^{-1} \bar{b},$$

where

$$T(\alpha, \beta, \omega) = \begin{pmatrix} (1-\omega)I & \omega i(\alpha W + T)^{-1}(W - \alpha T) \\ i\omega(1-\omega)(\beta T + W)^{-1}(\beta W - T) & (1-\omega)I + \omega^2 M(\alpha, \beta) \end{pmatrix} \tag{21}$$

is the iteration matrix of the GDSSOR iteration method.

When $\omega = 1$, it has

$$T(\alpha, \beta, 1) = \begin{pmatrix} 0 & i(\alpha W + T)^{-1}(W - \alpha T) \\ 0 & M(\alpha, \beta) \end{pmatrix}, \tag{22}$$

which can be viewed as the iteration matrix of the block Gauss-Seidel method for (15). It follows from the form of $T(\alpha, \beta, 1)$ in (22) that the spectral radius of $T(\alpha, \beta, 1)$ is the same as that of $M(\alpha, \beta)$.

The iteration scheme (20) indicates that at each step of the GDSSOR iteration, we need to solve two linear systems with $\alpha W + T$ and $\beta T + W$ as the coefficient matrices. Since the symmetric matrices W and T are positive definite and positive semi-definite matrices, respectively, both $\alpha W + T$ and $\beta T + W$ are symmetric positive definite. Thus we can solve them exactly by the Cholesky factorization or inexactly by the conjugate gradient (CG) method.

3. Convergence analysis of the GDSSOR iteration method for complex symmetric linear systems

The GDSSOR iteration method is convergent if and only if $\rho(T(\alpha, \beta, \omega)) < 1$. This section discusses the convergence of the GDSSOR iteration method, i.e., providing conditions to guarantee $\rho(T(\alpha, \beta, \omega)) < 1$. Before presenting our theorem, we start with four lemmas which are useful in our proof.

Lemma 3.1. *Let $W, T \in \mathbb{R}^{n \times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively. Then $\lambda_M = \mu^2$, where λ_M and μ are the eigenvalues of the matrices $M(\alpha, \beta)$ and $J(\alpha, \beta)$, respectively.*

Proof. Assume that μ is an eigenvalue of the matrix $J(\alpha,\beta)$. If $\mu=0$, then $|M(\alpha,\beta)|=|J(\alpha,\beta)|=0$, which shows that $\mu^2=0$ is an eigenvalue of the matrix $M(\alpha,\beta)$. Otherwise, $\mu\neq 0$, in this case we have

$$|\mu I-J(\alpha,\beta)|=\begin{vmatrix}\mu I & -i(\alpha W+T)^{-1}(W-\alpha T)\\ -i(\beta T+W)^{-1}(\beta W-T) & \mu I\end{vmatrix}=|\mu^2 I-M(\alpha,\beta)|=0,$$

which means that μ^2 is an eigenvalue of the matrix $M(\alpha,\beta)$. Conversely, if $\lambda_M=t^2$ is an eigenvalue of the matrix $M(\alpha,\beta)$, then it holds that $|\lambda_M I-M(\alpha,\beta)|=0$. If $\lambda_M=0$, then

$$|M(\alpha,\beta)|=0=\begin{vmatrix}0 & -i(\alpha W+T)^{-1}(W-\alpha T)\\ -i(\beta T+W)^{-1}(\beta W-T) & 0\end{vmatrix}.$$

Hence 0 is an eigenvalue of the matrix $J(\alpha,\beta)$. If $\lambda_M=t^2\neq 0$, then

$$|\lambda_M I-M(\alpha,\beta)|=\begin{vmatrix}tI & -i(\alpha W+T)^{-1}(W-\alpha T)\\ 0 & tI-\frac{1}{t}M(\alpha,\beta)\end{vmatrix}=|tI-J(\alpha,\beta)|=0.$$

Therefore, t is an eigenvalue of the matrix $J(\alpha,\beta)$ and the conclusion of this theorem is obtained. ■

Lemma 3.2. *[11, 29] If $\omega\neq 0$, if λ is a non-zero eigenvalue of the matrix $T(\alpha,\beta,\omega)$ of (21) and if μ satisfies*

$$(\lambda+\omega-1)^2=\lambda\omega^2\mu^2, \tag{23}$$

then μ is an eigenvalue of the block Jacobi matrix $J(\alpha,\beta)$ of (18). Conversely, if μ is an eigenvalue of $J(\alpha,\beta)$ and λ satisfies (23), then λ is an eigenvalue of $T(\alpha,\beta,\omega)$.

Lemma 3.3. *Let $W,T\in\mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively, and $M(\alpha,\beta)$ be the iteration matrix of the GDSS iteration method defined as in Theorem 2.1. Then λ_M is an eigenvalue of $M(\alpha,\beta)$ if*

$$\lambda_M=\frac{(\bar{\mu}-\beta)(1-\alpha\bar{\mu})}{(1+\beta\bar{\mu})(\alpha+\bar{\mu})},$$

where $\bar{\mu}$ is an eigenvalue of the matrix $W^{-1}T$.

Proof. Let x be the eigenvector associated with the eigenvalue $\bar{\mu}$ of the matrix $W^{-1}T$. Then $Tx=\bar{\mu}Wx$, which leads to $(\alpha W+T)x=(\alpha+\bar{\mu})Wx$ and $(\beta T+W)x=(\beta\bar{\mu}+1)Wx$. Thus, we can write

$$(\alpha W+T)^{-1}Wx=\frac{1}{\alpha+\bar{\mu}}x,\ (\beta T+W)^{-1}Wx=\frac{1}{\beta\bar{\mu}+1}x,$$

which gives the following equation

$$\begin{aligned}M(\alpha,\beta)x &= (\beta T+W)^{-1}(T-\beta W)(\alpha W+T)^{-1}(W-\alpha T)x\\ &= (1-\alpha\bar{\mu})(\beta T+W)^{-1}(T-\beta W)(\alpha W+T)^{-1}Wx\\ &= \frac{1-\alpha\bar{\mu}}{\alpha+\bar{\mu}}(\beta T+W)^{-1}(T-\beta W)x\\ &= \frac{(1-\alpha\bar{\mu})(\bar{\mu}-\beta)}{\alpha+\bar{\mu}}(\beta T+W)^{-1}Wx=\frac{(\bar{\mu}-\beta)(1-\alpha\bar{\mu})}{(1+\beta\bar{\mu})(\alpha+\bar{\mu})}x.\end{aligned}$$

This concludes the proof. ■

Lemma 3.4. *[38] Both roots of the real quadratic equation $x^2-bx+c=0$ are less than one in modulus if and only if $|c|<1$ and $|b|<1+c$.*

Theorem 3.1. *Let $W,T\in\mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively, and the parameters α and β satisfy the conditions of Theorem 2.1. When μ is real, the GDSSOR iteration method is convergent if and only if $0<\omega<2$; besides, when μ is a pure imaginary number, if $0<\omega<\frac{2}{1+\sqrt{\rho(M(\alpha,\beta))}}$, then the GDSSOR iteration method is convergent. Here, μ is the eigenvalue of the matrix $J(\alpha,\beta)$.*

Proof. Let λ be the eigenvalue of the matrix $T(\alpha,\beta,\omega)$. After rearranging Equation (23), we immediately obtain

$$\lambda^2+(2\omega-2-\omega^2\mu^2)\lambda+(\omega-1)^2=0. \tag{24}$$

According to Lemma 3.3, it is readily seen that all eigenvalues of the matrix $M(\alpha,\beta)$ are real. Then the eigenvalue μ of the matrix $J(\alpha,\beta)$ is a real or pure imaginary number in light of Lemma 3.1, which implies that $\mu^2\in\mathbb{R}$, and therefore Equation (24) is a real quadratic equation. By Lemma 3.4, $|\lambda|<1$ if and only if

$$|(\omega-1)^2|<1,\quad |2\omega-2-\omega^2\mu^2|<1+(\omega-1)^2. \tag{25}$$

By straightforwardly solving the first inequality of (25) we immediately obtain $0<\omega<2$. However, the solution of the second inequality of (25) is more involved and is demonstrated in the following. Easily, the second inequality of (25) can be simplified to

$$\omega^2(1-\mu^2)>0 \text{ and } \omega^2(1+\mu^2)-4\omega+4>0. \tag{26}$$

The combination of Theorem 2.1 and Lemma 3.1 leads to $|\mu|<1$. If $\mu^2\geq 0$, i.e., $\mu\in\mathbb{R}$, then it is obvious that (26) holds for $0<\omega<2$. It remains to consider the case $\mu^2<0$, or in other words μ is a pure imaginary number, then it follows from (26) that

$$0<\omega<\frac{2-2\sqrt{-\mu^2}}{1+\mu^2}=\frac{2-2|\mu|}{1-|\mu|^2}=\frac{2}{1+|\mu|} \text{ or } \omega>\frac{2+2\sqrt{-\mu^2}}{1+\mu^2}=\frac{2+2|\mu|}{1-|\mu|^2}>2,$$

which along with $0<\omega<2$ gives the convergence condition of the GDSSOR iteration method, namely

$$0<\omega<\frac{2}{1+\rho(J(\alpha,\beta))}=\frac{2}{1+\sqrt{\rho(M(\alpha,\beta))}}.$$

■

4. The choices for parameters of the GDSSOR iteration method

The main aim of this section is to determine the optimal iteration parameter ω and the corresponding optimal asymptotic convergence factor of the GDSSOR iteration method and provide a practical way for the choice of parameters of the GDSSOR iteration method.

Theorem 4.1. *Let the conditions of Theorem 3.1 be satisfied.*

- *If all eigenvalues μ of $J(\alpha,\beta)$ are pure imaginary numbers, then the optimal value of the parameter ω for the GDSSOR iteration method and the corresponding optimal convergence factor are*

$$\omega^*=\frac{2}{1+\sqrt{1+\rho(J(\alpha,\beta))^2}},\quad \rho(T(\alpha,\beta,\omega^*))=1-\frac{2}{1+\sqrt{1+\rho(J(\alpha,\beta))^2}}. \tag{27}$$

- *If all eigenvalues of $J(\alpha,\beta)$ are real numbers, then the optimal value of the parameter ω for the GDSSOR iteration method and the corresponding optimal convergence factor are*

$$\omega^*=\frac{2}{1+\sqrt{1-\rho(J(\alpha,\beta))^2}},\quad \rho(T(\alpha,\beta,\omega^*))=\frac{2}{1+\sqrt{1-\rho(J(\alpha,\beta))^2}}-1. \tag{28}$$

Proof. By simplifying Equation (23), we obtain

$$\lambda^2+[2(\omega-1)-\omega^2\mu^2]\lambda+(\omega-1)^2=0. \tag{29}$$

Now, we proceed the proof by investigating the two cases:

- μ is a pure imaginary number, i.e., $\mu^2<0$. We rewrite Equation (29) as $f_\omega(\lambda)=g(\lambda)$, where we define

$$f_\omega(\lambda)=\frac{\lambda+\omega-1}{\omega} \text{ and } g(\lambda)=\pm|\mu|\sqrt{-\lambda}.$$

  Clearly, $f_\omega(\lambda)$ passes through the point $(1,1)$ for all $\omega>0$. The straight line $f_\omega(\lambda)$ crosses the parabolas $g(\lambda)$. $f_\omega(\lambda)=g(\lambda)$ can be geometrically interpreted as the intersection of the curves $f_\omega(\lambda)$ and $g(\lambda)$, as illustrated in Figure 1. The discriminant of (29) is

$$\Delta(\omega,\mu)=\omega^2\mu^2(\omega^2\mu^2-4\omega+4). \tag{30}$$

When $\Delta(\omega,\mu) \geq 0$, the quadratic Equation (29) has two real roots λ_1 and λ_2. The largest abscissa of the intersection point decreases when the slope of $f_\omega(\lambda)$ increases until it becomes tangent to $g(\lambda)$. Under this condition, we have $\lambda_1 = \lambda_2$, and as a result $\Delta(\omega,\mu) = 0$, or equivalently,

$$-\omega^2\mu^2 + 4\omega - 4 = 0 \text{ or } \mu = 0. \tag{31}$$

If $\mu = 0$, then $|\lambda_1| = |\lambda_2| = |\omega - 1|$. In this case, $\omega = 1$ is the best choice because of $\lambda_1 = \lambda_2 = 0$.

If $\mu \neq 0$, then $-\omega^2\mu^2 + 4\omega - 4 = 0$. It can be seen from Theorem 3.1 that $0 < \omega < \frac{2}{1+\sqrt{\rho(M(\alpha,\beta))}}$, thus

$$0 < \hat{\omega} = \frac{2}{1+\sqrt{1-\mu^2}} = \frac{2}{1+\sqrt{1+|\mu|^2}} \leq 1$$

and $|\lambda_1| = |\lambda_2| = 1 - \hat{\omega}$. Note that $\pm\rho(J(\alpha,\beta))\sqrt{\lambda}$ is an envelope for all the curves $g(\lambda)$. So the minimum value of $|\lambda|$ is attained at $1 - \omega^*$ with

$$\omega^* = \frac{2}{1+\sqrt{1+\rho(J(\alpha,\beta))^2}}.$$

When $\Delta(\omega,\mu) < 0$, the quadratic Equation (29) has two conjugate complex roots λ_1 and λ_2. By some calculations, we deduce that $|\lambda_1| = |\lambda_2| = 1 - \omega$ with $0 < \omega < \omega^* < 1$. This investigation shows that $|\lambda_1|$ and $|\lambda_2|$ satisfy

$$|\lambda_1| = |\lambda_2| = 1 - \omega > 1 - \omega^* = 1 - \frac{2}{1+\sqrt{1+\rho(J(\alpha,\beta))^2}}.$$

- μ is real, then $\mu^2 \geq 0$ and it follows from (29) that the eigenvalues λ of the matrix $T(\alpha,\beta,\omega)$ satisfy

$$\frac{\lambda+\omega-1}{\omega} = \pm|\mu|\sqrt{\lambda}.$$

It can be viewed as the intersection points of the straight line

$$f_\omega(\lambda) = \frac{\lambda+\omega-1}{\omega},$$

which passes through the point $(1,1)$, and the parabolas $h(\lambda) = \pm|\mu|\sqrt{\lambda}$. Recalling that (30) is the discriminant of (29). When $\Delta(\omega,\mu) \geq 0$, the quadratic Equation (29) has two real roots λ_1 and λ_2. For each μ, these roots are abscissas of the intersections of $f_\omega(\lambda)$ and $h(\lambda)$, as illustrated in Figure 2.

Similar to the analysis in the case that μ is a pure imaginary number, the optimal value of the parameter ω is the choice that guarantees that $f_\omega(\lambda)$ is a tangent line of $h(\lambda)$. This shows that $\lambda_1 = \lambda_2$. Then $\Delta(\omega,\mu) = 0$ and (31) holds true.

If $\mu = 0$, then $|\lambda_1| = |\lambda_2| = |\omega - 1|$. In this case, $\omega = 1$ is the best choice because of $\lambda_1 = \lambda_2 = 0$ in this case.

If $\mu \neq 0$, then $\omega^2\mu^2 - 4\omega + 4 = 0$. The convergent domain $0 < \omega < 2$ implies that

$$1 \leq \hat{\omega} = \frac{2}{1+\sqrt{1-\mu^2}} < 2$$

as $0 \leq \mu^2 = |\mu|^2 < 1$, and we have $\lambda_1 = \lambda_2 = \hat{\omega} - 1$. Since $\pm\rho(J(\alpha,\beta))\sqrt{\lambda}$ is an envelope for all the curves $h(\lambda)$, the minimum value of $|\lambda|$ is attained at $\omega^* - 1$ with

$$\omega^* = \frac{2}{1+\sqrt{1-\rho(J(\alpha,\beta))^2}}.$$

Now, we turn to consider the case that $\Delta(\omega,\mu) < 0$, which implies that the quadratic Equation (29) has two conjugate complex roots λ_1 and λ_2. By straightforwardly solving (29), we obtain $|\lambda_1| = |\lambda_2| = \omega - 1$ with $1 \leq \omega^* < \omega < 2$. Thereby,

$$|\lambda_1| = |\lambda_2| = \omega - 1 > \omega^* - 1 = \frac{2}{1+\sqrt{1-\rho(J(\alpha,\beta))^2}} - 1.$$

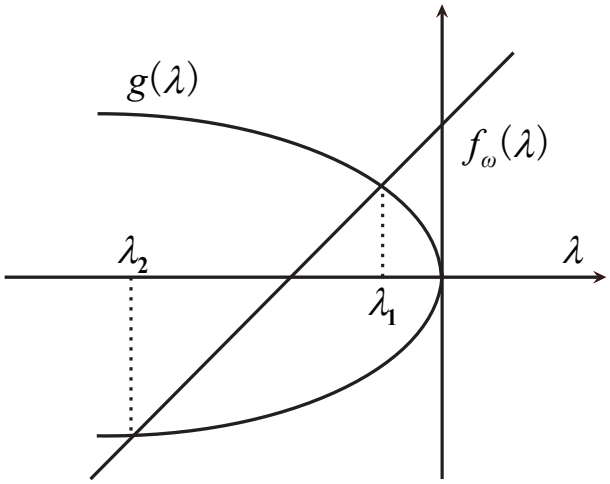

Figure 1: Plot of the curves of $f_\omega(\lambda)$ and $g(\lambda)$.

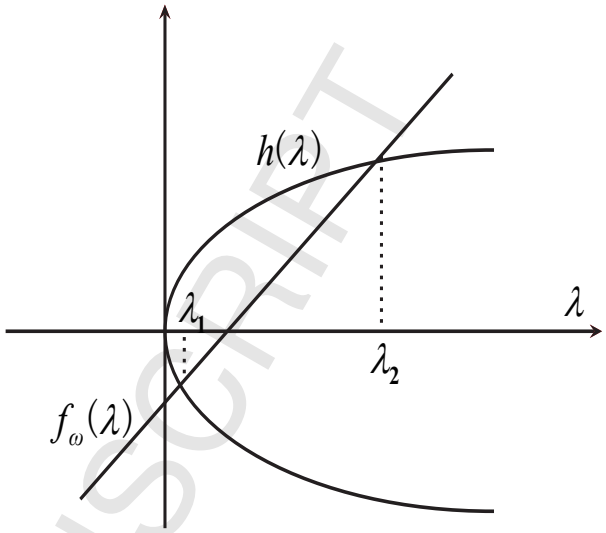

Figure 2: Plot of the curves of $f_\omega(\lambda)$ and $h(\lambda)$.

By combining the two cases above, we obtain the optimal value of ω and the corresponding optimal convergence factor $\rho(T(\alpha,\beta,\omega^*))$ in (27) and (28). ■

Note that the parameter ω^* derived in Theorem 4.1 is related to $\rho(J(\alpha,\beta))$. However, computing $\rho(J(\alpha,\beta))$ is very difficult and almost impossible when the problem size of $J(\alpha,\beta)$ are large enough. Here, a practical method for determining the parameter ω is to find ω to minimize

$$\begin{aligned}\Gamma(\omega) &= \left\| \begin{pmatrix} (1-\omega)(\alpha W+T) & \omega i(W-\alpha T) \\ 0 & (1-\omega)(\beta T+W) \end{pmatrix} \right\|_F^2 \\ &= (1-\omega)^2 tr((\alpha W+T)^2) + (1-\omega)^2 tr((\beta T+W)^2) + \omega^2 tr((W-\alpha T)^2) \\ &= \omega^2 tr((\alpha W+T)^2 + (\beta T+W)^2 + (W-\alpha T)^2) \\ &\quad -2\omega tr((\alpha W+T)^2 + (\beta T+W)^2) + tr((\alpha W+T)^2 + (\beta T+W)^2).\end{aligned}$$

This method comes essentially from [20]. Let $a = tr((\alpha W+T)^2 + (\beta T+W)^2 + (W-\alpha T)^2)$ and $b = tr((\alpha W+T)^2 + (\beta T+W)^2)$. By direct computations, we get

$$\frac{d\Gamma(\omega)}{d\omega} = 2a\omega - 2b.$$

It is not difficult to see that

$$\frac{d\Gamma(\omega)}{d\omega} \le 0, \text{ for } 0 < \omega \le \frac{b}{a}, \text{ and } \frac{d\Gamma(\omega)}{d\omega} \ge 0, \text{ for } \omega \ge \frac{b}{a}.$$

Hence, $\omega_{pre} = \frac{b}{a}$ is the optimal iteration parameter that minimizes $\Gamma(\omega)$, where

$$\omega_{pre} = \frac{tr((\alpha W+T)^2 + (\beta T+W)^2)}{tr((\alpha W+T)^2 + (\beta T+W)^2 + (W-\alpha T)^2)} = \frac{(\alpha^2+1)tr(W^2) + (\beta^2+1)tr(T^2) + 2(\alpha+\beta)tr(WT)}{(\alpha^2+2)tr(W^2) + (\alpha^2+\beta^2+1)tr(T^2) + 2\beta tr(WT)}. \quad (32)$$

Moreover, the convergence behavior of the GDSSOR iteration method also depends on the choice of the parameters α and β. Next, a practical way for the choice of parameters α and β of the GDSSOR iteration method is given by adopting the idea of [15].

Two sub-systems of linear equations in the GDSSOR iteration method need to be solved, which have coefficient matrices $\alpha W+T$ and $\beta T+W$, respectively. If either of them are solved inefficiently, then the convergence speed of the GDSSOR method will be deteriorated. We can rewrite the GDSSOR iteration method in (20) into the equivalent form

$$\begin{cases} (\alpha I + W^{-\frac{1}{2}} T W^{-\frac{1}{2}})\tilde{x}^{(k+1)} = (1-\omega)(\alpha I + W^{-\frac{1}{2}} T W^{-\frac{1}{2}})\tilde{x}^{(k)} + \omega[i(I - \alpha W^{-\frac{1}{2}} T W^{-\frac{1}{2}})\tilde{y}^{(k)} + (\alpha - i)\tilde{b}], \\ (\beta W^{-\frac{1}{2}} T W^{-\frac{1}{2}} + I)\tilde{y}^{(k+1)} = (1-\omega)(\beta W^{-\frac{1}{2}} T W^{-\frac{1}{2}} + I)\tilde{y}^{(k)} + \omega[i(\beta I - W^{-\frac{1}{2}} T W^{-\frac{1}{2}})\tilde{x}^{(k+1)} + (1-\beta i)\tilde{b}], \end{cases}$$

where $\tilde{x}^{(k)} = W^{\frac{1}{2}} x^{(k)}$, $\tilde{y}^{(k)} = W^{\frac{1}{2}} y^{(k)}$ and $\tilde{b} = W^{-\frac{1}{2}} b$. Then the GDSSOR iteration method may have fast convergence rate if α and β minimize the function $\tau(\alpha,\beta) := |(\kappa(\alpha I + W^{-\frac{1}{2}} T W^{-\frac{1}{2}}) - \kappa(\beta W^{-\frac{1}{2}} T W^{-\frac{1}{2}} + I))|$. It holds

that there exists an unitary matrix such that $U^*W^{-\frac{1}{2}}TW^{-\frac{1}{2}}U = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n)$ with $\tilde{\lambda}_i \geq 0$ $(i = 1, 2, \ldots, n)$ due to the fact that $W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$ is symmetric positive semi-definite. By direct calculations, we have

$$\tau(\alpha,\beta) = |\|(\alpha I + \bar{S})^{-1}\|_2 \cdot \|\alpha I + \bar{S}\|_2 - \|(\beta\bar{S} + I)^{-1}\|_2 \cdot \|\beta\bar{S} + I\|_2| = \left|\frac{\alpha + \tilde{\lambda}_{\max}}{\alpha + \tilde{\lambda}_{\min}} - \frac{\beta\tilde{\lambda}_{\max} + 1}{\beta\tilde{\lambda}_{\min} + 1}\right|,$$

where $\bar{S} = W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$, and $\tilde{\lambda}_{\max}$ and $\tilde{\lambda}_{\min}$ are the maximal and the minimal eigenvalues of the matrix $\bar{S}$, respectively. If

$$\|(\alpha I + W^{-\frac{1}{2}}TW^{-\frac{1}{2}})^{-1}\|_2 \cdot \|\alpha I + W^{-\frac{1}{2}}TW^{-\frac{1}{2}}\|_2 = \|(\beta W^{-\frac{1}{2}}TW^{-\frac{1}{2}} + I)^{-1}\|_2 \cdot \|\beta W^{-\frac{1}{2}}TW^{-\frac{1}{2}} + I\|_2,$$

then $\tau(\alpha, \beta) = 0$. As a result, we have

$$\frac{\alpha + \tilde{\lambda}_{\max}}{\alpha + \tilde{\lambda}_{\min}} = \frac{\beta\tilde{\lambda}_{\max} + 1}{\beta\tilde{\lambda}_{\min} + 1},$$

which results in $(\alpha\beta - 1)(\tilde{\lambda}_{\max} - \tilde{\lambda}_{\min}) = 0$. $\tilde{\lambda}_{\max} \neq \tilde{\lambda}_{\min}$ may be supposed as it holds in many cases. Thus, $\alpha\beta = 1$, and for a broad class of problems we can choose proper parameters $\alpha \geq 1$ and $\beta \leq 1$ satisfying $\alpha\beta = 1$ for the GDSSOR iteration method according to Remark 2.1.

The above analyses show that we can adopt proper α and β satisfying $\alpha\beta = 1$, and ω obtained in (32) for the GDSSOR iteration method in the practical implements, whereas it may be very time consuming to find the proper α and β satisfying $\alpha\beta = 1$ to make the GDSSOR iteration method effective.

Theorem 4.1 shows that the optimal convergence factor of the GDSSOR iteration method is strictly monotonic increasing about $\rho(J(\alpha,\beta))^2$ or $\rho(M(\alpha,\beta))$, so it makes sense to chose the parameters α and β to minimize it. However, it is difficult to find the optimal values of α and β in general. Instead, we can consider to minimize the upper bound of $\rho(M(\alpha,\beta))$ to get the quasi-optimal parameters α^* and β^*.

Theorem 4.2. *Let $W, T \in \mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively. Then, the parameters β^* and α^* that minimize the upper bound*

$$\gamma(\alpha,\beta) = \max_i \left|\frac{\beta - \mu_i}{1 + \beta\mu_i}\right| \cdot \max_i \left|\frac{1 - \alpha\mu_i}{\alpha + \mu_i}\right|$$

of $\rho(M(\alpha,\beta))$ are:

$$\beta^* = \frac{\mu_{\min}\mu_{\max} - 1 + \sqrt{(1+\mu_{\min}^2)(1+\mu_{\max}^2)}}{\mu_{\min} + \mu_{\max}}, \quad \alpha^* = \frac{1 - \mu_{\min}\mu_{\max} + \sqrt{(1+\mu_{\min}^2)(1+\mu_{\max}^2)}}{\mu_{\min} + \mu_{\max}}, \tag{33}$$

where $\mu_{\min}$ and $\mu_{\max}$ denote the minimum and maximum eigenvalues of the matrix $S = W^{-1}T$, respectively.

Proof. From Theorem 2.1, we have

$$\begin{aligned}
\rho(M(\alpha,\beta)) &= \rho((\beta T + W)^{-1}(\beta W - T)(\alpha W + T)^{-1}(W - \alpha T)) \\
&= \rho((\beta S + I)^{-1}(\beta I - S)(\alpha I + S)^{-1}(I - \alpha S)) \\
&\leq \|(\beta S + I)^{-1}(\beta I - S)\|_2 \|(\alpha I + S)^{-1}(I - \alpha S)\|_2 \\
&= \max_i \left|\frac{\beta - \mu_i}{1 + \beta\mu_i}\right| \cdot \max_i \left|\frac{1 - \alpha\mu_i}{\alpha + \mu_i}\right| =: \gamma(\alpha,\beta),
\end{aligned}$$

where $S = W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$ and μ_i is an eigenvalue of the matrix S. We now compute the quasi-optimal parameters α^* and β^* which minimize the upper bound $\gamma(\alpha,\beta)$ of $\rho(M(\alpha,\beta))$.

Note that $g(\mu) = \frac{\mu-\beta}{\beta\mu+1}$ is an increasing function with respect to the variable μ, we have

$$\max_i |g(\mu_i)| = \max_i \left|\frac{\beta - \mu_i}{1 + \beta\mu_i}\right| = \max_{\mu\in\sigma(W^{-1}T)} \left|\frac{\mu - \beta}{\beta\mu + 1}\right| = \max\left\{\frac{\mu_{\max} - \beta}{\beta\mu_{\max} + 1}, \frac{\beta - \mu_{\min}}{\beta\mu_{\min} + 1}\right\}.$$

If $\mu_{\max} \leq \beta$, then for any $\mu \in \sigma(W^{-1}T)$, it holds that $\mu - \beta \leq 0$ and

$$\max_i \left|\frac{\beta - \mu_i}{1 + \beta\mu_i}\right| = \max_{\mu\in\sigma(W^{-1}T)} \frac{\beta - \mu}{\beta\mu + 1} = \frac{\beta - \mu_{\min}}{\beta\mu_{\min} + 1}$$

as $\frac{\beta-\mu}{\beta\mu+1}$ is monotonic decreasing about μ. If $\mu_{\max} \geq \beta$, then $\mu_{\max} - \beta \geq 0$. First, we assume that $\mu_{\min} \neq 0$. Below we distinguish two cases to discuss.

(i) $\mu_{\min} \leq \beta$, it holds that $\mu_{\min} - \beta \leq 0$ and therefore

$$\max_i \left| \frac{\beta - \mu_i}{1+\beta\mu_i} \right| = \max \left\{ \frac{\mu_{\max} - \beta}{\beta\mu_{\max}+1}, \frac{\beta - \mu_{\min}}{\beta\mu_{\min}+1} \right\}.$$

(ii) $\beta \leq \mu_{\min}$, then $\mu_{\min} - \beta \geq 0$. By considering the monotone property of the function $\frac{\mu-\beta}{\beta\mu+1}$ with respect to μ again, we obtain

$$\max_i \left| \frac{\beta - \mu_i}{1+\beta\mu_i} \right| = \max_{\mu \in \sigma(W^{-1}T)} \frac{\mu - \beta}{\beta\mu+1} = \frac{\mu_{\max} - \beta}{\beta\mu_{\max}+1}.$$

Now, summarizing the above discussions, we have the following results

$$\max_i \left| \frac{\beta - \mu_i}{1+\beta\mu_i} \right| = \begin{cases} \dfrac{\beta - \mu_{\min}}{\beta\mu_{\min}+1}, & \text{for } \beta \geq \mu_{\max}, \\ \max \left\{ \dfrac{\mu_{\max} - \beta}{\beta\mu_{\max}+1}, \dfrac{\beta - \mu_{\min}}{\beta\mu_{\min}+1} \right\}, & \text{for } \mu_{\min} \leq \beta \leq \mu_{\max}, \\ \dfrac{\mu_{\max} - \beta}{\beta\mu_{\max}+1}, & \text{for } \beta \leq \mu_{\min}. \end{cases}$$

With concrete computations, we obtain

$$\frac{\partial}{\partial\beta} \max_i \left| \frac{\beta - \mu_i}{1+\beta\mu_i} \right| = \begin{cases} \dfrac{1+\mu_{\min}^2}{(\beta\mu_{\min}+1)^2} > 0, & \text{for } \beta \in [\beta^*, \mu_{\max}] \cup [\mu_{\max}, +\infty) = [\beta^*, +\infty), \\ -\dfrac{1+\mu_{\max}^2}{(\beta\mu_{\max}+1)^2} < 0, & \text{for } \beta \in (0, \mu_{\min}] \cup [\mu_{\min}, \beta^*] = (0, \beta^*]. \end{cases}$$

From the monotonicity of $\max_i |g(\mu_i)|$ with respect to β, we can assert that if β^* is the minimum point of $\max_i |g(\mu_i)|$, then it is located in the interval $[\mu_{\min}, \mu_{\max}]$ and must satisfy $\frac{\mu_{\max}-\beta^*}{\beta^*\mu_{\max}+1} = \frac{\beta^*-\mu_{\min}}{\beta^*\mu_{\min}+1}$, which can be transformed into

$$(\beta^*)^2(\mu_{\max} + \mu_{\min}) - 2\beta^*(\mu_{\max}\mu_{\min} - 1) - (\mu_{\max} + \mu_{\min}) = 0.$$

In view of $\beta > 0$, we solve the above equation and derive

$$\beta^* = \frac{\mu_{\min}\mu_{\max} - 1 + \sqrt{(1+\mu_{\min}^2)(1+\mu_{\max}^2)}}{\mu_{\min} + \mu_{\max}}. \tag{34}$$

For the case that $\mu_{\min} = 0$, similarly to the above demonstration, we have

$$\max_i \left| \frac{\beta - \mu_i}{1+\beta\mu_i} \right| = \begin{cases} \beta, & \text{for } \beta \geq \mu_{\max}, \\ \max \left\{ \dfrac{\mu_{\max} - \beta}{\beta\mu_{\max}+1}, \beta \right\}, & \text{for } 0 < \beta \leq \mu_{\max}. \end{cases}$$

Then $\max_i |g(\mu_i)|$ attains its minimum if $\frac{\mu_{\max}-\beta}{\beta\mu_{\max}+1} = \beta$, that is $\beta^* = \frac{\sqrt{1+\mu_{\max}^2}-1}{\mu_{\max}}$, which is exactly the value of (34) with $\mu_{\min} = 0$.

With a quite similar strategy utilized in the above proof, the optimal parameter α^* defined as in (33) can also be derived. The proof of this theorem is completed. ■

A direct calculation verifies that the parameters α^* and β^* developed in (33) exactly satisfy the condition $\alpha^*\beta^* = 1$. The numerical results in Section 6 illustrate the effectiveness of the GDSSOR iteration method by using the parameters α^*, β^* and ω_{pre}.

Remark 4.1. Some remarks on the computations of the parameters α^*, β^* and ω_{pre} are as follows.

- When we compute the parameters α^*, β^* of the GDSSOR iteration method according to (33), the maximum and minimum eigenvalues of the matrix $W^{-1}T$ are estimated by a few iterations of the power and the inverse power methods, respectively. Besides, when we compute the parameter ω_{pre} of the GDSSOR iteration method in terms of (32), the traces of the matrices W^2, T^2 and WT need to be computed. To reduce workload, they are calculated according to the formula

$$tr(AB) = \sum_{i,j=1}^{n} (A \circ B^T)_{ij}, \ \ A, B \in \mathbb{C}^{n\times n}, \tag{35}$$

  where $\circ$ denotes the Hadamard product. In this way, the calculation of the matrix AB is avoid [37].

- Suppose that the matrices W and T are dense. Then computing $W^{-1}T$ requires $\mathcal{O}(n^3)$ flops, and estimating the maximum and minimum eigenvalues of the matrix $W^{-1}T$ by the power and the inverse power methods needs $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ flops, respectively. Therefore, the overall overheads for computing α^* and β^* in (33) are $\mathcal{O}(n^3)$ flops. Besides, computing the traces of the matrices W^2, T^2 and WT according to (35) requires $\mathcal{O}(n^2)$ flops, which means that the overall overheads for computing ω_{pre} in (32) are $\mathcal{O}(n^2)$ flops. In practice, since the matrices W and T are generally sparse, the total cost can be reduced.

5. The inexact GDSSOR iteration method

As mentioned in [10], at each step of the HSS method, we should solve two subsystems with the coefficient matrices $\alpha I + H$ and $\alpha I + S$, which is very costly and impractical in actual implementations. To further improve the efficiency of the HSS iteration method, Bai et al. [10, 12] employed the iterative methods for solving the two subproblems, e.g., solving the linear systems with coefficient matrix $\alpha I + H$ by the CG and those with coefficient matrix $\alpha I + S$ by the Lanczos or the CG for normal equations (CGNE) method, to some prescribed accuracies, and obtained two special but quite practical inexact HSS (IHSS) iterations, briefly called as IHSS(CG, Lanczos) and IHSS(CG, CGNE). Recent years, lots of work on inexact iteration methods has been contributed to solve the linear systems, including the inexact LHSS (ILHSS) iteration method [23], the inexact non-alternating preconditioned HSS (INPHSS) method [32], the inexact quasi-HSS (IQHSS) iteration method [6], the inexact two-step parameterized (ITSP) iteration method [21] and so forth.

At each step of the GDSSOR iteration two linear systems with coefficient matrix $\alpha W + T$ and $\beta T + W$ need to be solved, which is very costly and impractical in actual implementations. To overcome this disadvantage, we can solve these two linear systems by employing the CG method owing to the fact that they are symmetric positive definite. With this approach the linear subsystems are solved inexactly, leading to the inexact GDSSOR iteration method as follows.

The inexact GDSSOR (IGDSSOR) iteration method

Given an initial guess $(\bar{x}^{(0)}; \bar{y}^{(0)})$, for $k = 0, 1, 2, \ldots$, until $(\bar{x}^{(k)}; \bar{y}^{(k)})$ converges, solve $\bar{x}^{(k+1)}$ approximately from

$$(\alpha W + T)\bar{x}^{(k+1)} \approx (1-\omega)(\alpha W + T)\bar{x}^{(k)} + \omega[i(W - \alpha T)\bar{y}^{(k)} + (\alpha - i)b]$$

by employing the CG method with $\bar{x}^{(k)}$ and $\bar{y}^{(k)}$ as the initial guesses; then solve $\bar{y}^{(k+1)}$ approximately from

$$(\beta T + W)\bar{y}^{(k+1)} \approx (1-\omega)(\beta T + W)\bar{y}^{(k)} + \omega[i(\beta W - T)\bar{x}^{(k+1)} + (1 - \beta i)b]$$

by employing the CG method with $\bar{y}^{(k)}$ and $\bar{x}^{(k+1)}$ as the initial guesses, where α, β and ω are positive constants.

To simplify numerical implementation and convergence analysis, we may rewrite the above IGDSSOR iteration method as the following equivalent scheme:

Given an initial guess $(\bar{x}^{(0)}; \bar{y}^{(0)})$, for $k = 0, 1, 2, \ldots$, until $(\bar{x}^{(k)}; \bar{y}^{(k)})$ converges,

1. Compute $\bar{r}^{(k)} = \omega[(\alpha - i)b - (\alpha W + T)\bar{x}^{(k)} + i(W - \alpha T)\bar{y}^{(k)}]$;
2. Approximate the solution of $(\alpha W + T)\bar{z}^{(k)} = \bar{r}^{(k)}$ by the CG method until $\bar{z}^{(k)}$ is such that the residual $\bar{p}^{(k)} = \bar{r}^{(k)} - (\alpha W + T)\bar{z}^{(k)}$ satisfies $\|\bar{p}^{(k)}\| \leq \eta_k \|\bar{r}^{(k)}\|$;
3. Compute $\bar{x}^{(k+1)} = \bar{x}^{(k)} + \bar{z}^{(k)}$;
4. Compute $\bar{t}^{(k)} = \omega[(1 - \beta i)b - (\beta T + W)\bar{y}^{(k)} + i(\beta W - T)\bar{x}^{(k+1)}]$;
5. Approximate the solution of $(\beta T + W)\bar{h}^{(k)} = \bar{t}^{(k)}$ by the CG method until $\bar{h}^{(k)}$ is such that the residual $\bar{q}^{(k)} = \bar{t}^{(k)} - (\beta T + W)\bar{h}^{(k)}$ satisfies $\|\bar{q}^{(k)}\| \leq \eta_k \|\bar{t}^{(k)}\|$;
6. Compute $\bar{y}^{(k+1)} = \bar{y}^{(k)} + \bar{h}^{(k)}$.

Here, $\|.\|$ is a norm of a vector, and $\{\eta_k\}$ is a prescribed tolerance.

We can find that the IGDSSOR iteration scheme is induced by the matrix splitting $\omega\bar{A} = M_{\alpha,\beta} - N_{\alpha,\beta}$, where

$$M_{\alpha,\beta} = \begin{pmatrix} \alpha W + T & 0 \\ -i\omega(\beta W - T) & \beta T + W \end{pmatrix}, \quad N_{\alpha,\beta} = \begin{pmatrix} (1-\omega)(\alpha W + T) & \omega i(W - \alpha T) \\ 0 & (1-\omega)(\beta T + W) \end{pmatrix}.$$

Then, the IGDSSOR iteration method computes the approximate solution $\bar{u}^{(k+1)} = \bar{u}^{(k)} + \bar{s}^{(k)}$ at $(k+1)$th iteration such that $\bar{s}^{(k)} = (\bar{z}^{(k)}; \bar{h}^{(k)})$ is the solution of the system

$$M_{\alpha,\beta}\bar{s}^{(k)} = \bar{g}^{(k)},$$

in which $\bar{g}^{(k)} = (\bar{r}^{(k)}; \bar{t}^{(k)}) = \omega(\bar{b} - \bar{A}\bar{u}^{(k)})$ with $\bar{u}^{(k)} = (\bar{x}^{(k)}; \bar{y}^{(k)})$ and $\bar{b} = ((\alpha - i)b; (1 - \beta i)b)$. In addition, the stopping criterion is $\|\bar{v}^{(k)}\| \leq \eta_k \|\bar{g}^{(k)}\|$, where $\bar{v}^{(k)} = \bar{g}^{(k)} - M_{\alpha,\beta}\bar{s}^{(k)}$.

In what follows, we analyze the convergence of the IGDSSOR method. For this propose, we first quote the following useful result.

Lemma 5.1. *[28] For every matrix $B \in \mathbb{C}^{n\times n}$ and $\varepsilon > 0$, there exists a norm $\|\cdot\|$ on $\mathbb{C}^n$ such that for the corresponding induced norm, $\|B\| \le \rho(B) + \varepsilon$.*

Theorem 5.1. *Let $W, T \in \mathbb{R}^{n\times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively, and let $\bar{A}$ be defined as in (17) and $\omega\bar{A} = M_{\alpha,\beta} - N_{\alpha,\beta}$, where α, β and ω satisfy the conditions of Theorems 2.1 and 2.4, respectively. There exists a norm $\|\cdot\|$ on $\mathbb{C}^{2n}$ such that for the corresponding induced norm, $\|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\| < 1$. Also, if $\{\bar{u}^{(k)}\}$ is a sequence defined as*

$$\bar{u}^{(k+1)} = \bar{u}^{(k)} + \bar{s}^{(k)}, \text{ with } M_{\alpha,\beta}\bar{s}^{(k)} = \bar{v}^{(k)} + \bar{g}^{(k)}, \tag{36}$$

satisfying $\|\bar{v}^{(k)}\| \le \eta_k\|\bar{g}^{(k)}\|$, where $\bar{g}^{(k)} = \omega(\bar{b} - \bar{A}\bar{u}^{(k)})$, and u^* is the exact solution of the system (17), then it holds that

$$\|\bar{u}^{(k+1)} - u^*\| \le (\|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\| + \eta_k\omega\|M_{\alpha,\beta}^{-1}\|\|\bar{A}\|)\|\bar{u}^{(k)} - u^*\|.$$

In particular, if

$$\eta_{\max} < \frac{1 - \|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\|}{\omega\|M_{\alpha,\beta}^{-1}\|\|\bar{A}\|}, \tag{37}$$

then the sequence $\{\bar{u}^{(k)}\}$ converges to u^*, where $\eta_{\max} = \max\limits_k\{\eta_k\}$.

Proof. Since $\bar{A} = \frac{1}{\omega}M_{\alpha,\beta} - \frac{1}{\omega}N_{\alpha,\beta}$ is the GDSSOR splitting of $\bar{A}$ and α, β and ω are located in their convergence domains, we have $\rho(M_{\alpha,\beta}^{-1}N_{\alpha,\beta}) = \rho(T(\alpha,\beta,\omega)) < 1$. According to Lemma 5.1, for $\gamma = \frac{1-\rho(T(\alpha,\beta,\omega))}{2}$, there exists a norm $\|\cdot\|$ on $\mathbb{C}^{2n}$ such that for the corresponding induced norm it has

$$\|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\| \le \rho(M_{\alpha,\beta}^{-1}N_{\alpha,\beta}) + \varepsilon < 1,$$

It follows from (36) that

$$\begin{aligned}
\bar{u}^{(k+1)} &= \bar{u}^{(k)} + M_{\alpha,\beta}^{-1}(\bar{v}^{(k)} + \bar{g}^{(k)}) \\
&= \bar{u}^{(k)} + M_{\alpha,\beta}^{-1}(\bar{v}^{(k)} + \omega(\bar{b} - \bar{A}\bar{u}^{(k)})) \\
&= (I - \omega M_{\alpha,\beta}^{-1}\bar{A})\bar{u}^{(k)} + M_{\alpha,\beta}^{-1}\bar{v}^{(k)} + \omega M_{\alpha,\beta}^{-1}\bar{b} \\
&= M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\bar{u}^{(k)} + M_{\alpha,\beta}^{-1}\bar{v}^{(k)} + \omega M_{\alpha,\beta}^{-1}\bar{b}.
\end{aligned} \tag{38}$$

Since u^* is the exact solution of the system (17), it holds that

$$u^* = M_{\alpha,\beta}^{-1}N_{\alpha,\beta}u^* + \omega M_{\alpha,\beta}^{-1}\bar{b}. \tag{39}$$

Subtracting Equation (39) from Equation (38) leads to

$$\bar{u}^{(k+1)} - u^* = M_{\alpha,\beta}^{-1}N_{\alpha,\beta}(\bar{u}^{(k)} - u^*) + M_{\alpha,\beta}^{-1}\bar{v}^{(k)},$$

which results in

$$\|\bar{u}^{(k+1)} - u^*\| \le \|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\|\|\bar{u}^{(k)} - u^*\| + \|M_{\alpha,\beta}^{-1}\|\|\bar{v}^{(k)}\|. \tag{40}$$

Moreover, we can derive

$$\|\bar{v}^{(k)}\| \le \eta_k\|\bar{g}^{(k)}\| = \eta_k\|\omega(\bar{b} - \bar{A}\bar{u}^{(k)})\| = \eta_k\omega\|\bar{A}(u^* - \bar{u}^{(k)})\| \le \eta_k\omega\|\bar{A}\|\|u^* - \bar{u}^{(k)}\|. \tag{41}$$

The combination of Inequalities (40) and (41) yields the following result

$$\begin{aligned}
\|\bar{u}^{(k+1)} - u^*\| &\le \|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\|\|\bar{u}^{(k)} - u^*\| + \eta_k\omega\|M_{\alpha,\beta}^{-1}\|\|\bar{A}\|\|u^* - \bar{u}^{(k)}\| \\
&= (\|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\| + \eta_k\omega\|M_{\alpha,\beta}^{-1}\|\|\bar{A}\|)\|u^* - \bar{u}^{(k)}\|.
\end{aligned}$$

Then a sufficient condition for guaranteeing the above inequality is

$$\|M_{\alpha,\beta}^{-1}N_{\alpha,\beta}\| + \eta_{\max}\omega\|M_{\alpha,\beta}^{-1}\|\|\bar{A}\| < 1,$$

which is valid if (37) holds true. Up to now, the proof has been completed. ■

6. Numerical experiments

In this section, we carry out four examples to examine the feasibility and effectiveness of the GDSSOR and the IGDSSOR iteration methods, and show the advantages of the GDSSOR iteration method over the PMHSS, PPNHSS, MSNS, DSS and PFPAE ones, in terms of both the number of iteration steps (denoted as "IT") and the elapsed CPU time in seconds (denoted as "CPU"). In the meanwhile, we compare the performance of the IGDSSOR iteration method with those of the inexact versions of the PMHSS, PPNHSS, CRI, DSS and PFPAE ones. Here, the preconditioned matrix V in the PMHSS and the PPNHSS iteration methods is taken as W. The experimentally found optimal parameters of the tested iteration methods used in actual computations are obtained experimentally by minimizing the corresponding iteration steps. In all tables, the parameters α_{pre} and β_{pre} are the ones that satisfy $\alpha\beta = 1$ and minimize the IT of the GDSSOR iteration method, and the parameters α^* and β^* are the ones obtained in (33). In addition, the parameter ω_{pre} is the one computed by (32).

We performe all experiments using MATLAB (version R2016a) on a personal computer with Intel (R) Pentium (R) CPU G3240T 2.70 GHz, 4.0 GB memory and XP operating system. In our computations, all iteration methods are started from the zero vector, and terminated once the stopping criterion $\text{RES} = \sqrt{\frac{\|b-Ax^{(k)}\|_2}{\|b\|_2}} < 10^{-6}$ is satisfied.

Example 6.1. *We consider the following complex symmetric linear system [7, 8]:*

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

where M and K are the inertia and the stiffness matrices, C_V and C_H are the viscous and the hysteretic damping matrices, respectively, and ϖ is the driving circular frequency. We take $C_H = \mu K$ with μ a damping coefficient, $M = I$, $C_V = 10I$, and K the five-point centered difference matrix approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0,1]\times[0,1]$ with the mesh-size $h = 1/(m+1)$. The matrix $K \in \mathbb{R}^{n\times n}$ possesses the tensor-product form $K = I \otimes V_m + V_m \otimes I$, with $V_m = h^{-2}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}$. Hence, K is an $n \times n$ block-tridiagonal matrix, with $n = m^2$. In addition, we set the right-hand side vector b to be $b = (1+i)A\mathbf{1}$, with $\mathbf{1}$ being the vector of all entries equal to 1. As before, we normalize the system by multiplying both sides through by h^2.

Example 6.2. *Consider the complex Helmholtz equation [17, 33, 39]:*

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

with σ_1 and σ_2 being real coefficient functions. Here, u satisfies Dirichlet boundary conditions in the square $D = [0,1]\times[0,1]$. By discretizing this equation with finite differences on an $m \times m$ grid with mesh size $h = 1/(m+1)$, we obtain a complex linear system

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

where the matrix $K \in \mathbb{R}^{n\times n}$ possesses the tensor-product form

$$K = I \otimes B_m + B_m \otimes I \text{ with } B_m = \frac{1}{h^2}\cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}.$$

Actually, K is the five-point centered difference matrix approximating the negative Laplacian operator $L = -\Delta$. In our tests, let the right-hand side vector $b = (1+i)A\mathbf{1}$ with $\mathbf{1}$ being the vector of all entries are equal to 1. In addition, we normalize the complex linear system by multiplying both sides by h^2.

Example 6.3. *[7, 8, 17, 30] Consider the linear system of equations $(W + iT)x = b$, with*

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

where $V = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}$, $V_c = V - e_1 e_m^T - e_m e_1^T \in \mathbb{R}^{m\times m}$ and e_1 and e_m are the first and last unit vectors in $\mathbb{R}^m$, respectively. We take the right-hand side vector b to be $b = (1+i)A\mathbf{1}$, with $\mathbf{1}$ being the vector of all entries equal to 1.

Here T and W correspond to the five-point centered difference matrices approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions and periodic boundary conditions, respectively, on a uniform mesh in the unit square $[0,1]\times[0,1]$ with the mesh-size $h = \frac{1}{m+1}$.

Table 1: Numerical results of Example 6.1 for the six iteration methods when $(\varpi, \mu) = (\pi, 0.02)$.

Method	m	16	32	48	64
	$\alpha_{\exp}$	0.8	0.9	0.9	0.9
PMHSS	IT	69	74	75	76
	CPU	0.0421	1.0196	4.7032	14.5300
	RES	9.27e-07	9.11e-07	9.63e-07	8.87e-07
	$\alpha_{\exp}$	1.1	1.1	1.1	1.1
	$\omega_{\exp}$	1.3	1.35	1.35	1.35
PPNHSS	IT	32	33	33	33
	CPU	0.0193	0.4656	2.0347	6.6392
	RES	8.73e-07	9.65e-07	8.44e-07	8.64e-07
	$\alpha_{\exp}$	0.17	0.07	0.04	0.03
MSNS	IT	18	30	47	60
	CPU	0.0198	0.5002	3.7481	14.9883
	RES	8.88e-07	7.42e-07	7.79e-07	8.31e-07
	$\alpha_{\exp}$	0.12	0.09	0.09	0.08
DSS	IT	40	47	51	51
	CPU	0.0251	0.6716	3.2774	10.6399
	RES	9.18e-07	9.31e-07	9.28e-07	8.62e-07
	$\alpha_{\exp}$	0.65	0.65	0.65	0.65
	$\omega_{\exp}$	1.3	1.4	1.4	1.4
PFPAE	IT	50	51	50	50
	CPU	0.0182	0.5474	2.4550	7.3635
	RES	8.44e-07	9.61e-07	9.75e-07	8.87e-07
	$\alpha_{\exp}$	1.25	1.32	1.32	1.32
	$\beta_{\exp}$	0.72	0.72	0.72	0.72
	$\omega_{\exp}$	0.89	0.89	0.89	0.89
GDSSOR	IT	14	14	14	14
	CPU	0.0087	0.1891	0.8675	2.8197
	RES	7.49e-07	8.23e-07	6.79e-07	6.78e-07
	α_{pre}	1.7	1.8	1.8	1.8
	β_{pre}	0.5882	0.5556	0.5556	0.5556
	ω_{pre}	0.8258	0.8275	0.8253	0.8245
GDSSOR	IT	17	17	17	17
	CPU	0.0102	0.2176	1.0021	3.4041
	RES	7.98e-07	6.76e-07	7.69e-07	8.14e-07
	α^*	1.3055	1.3210	1.3248	1.3263
	β^*	0.7660	0.7570	0.7549	0.7540
	ω_{pre}	0.7635	0.7536	0.7518	0.7512
GDSSOR	IT	21	21	21	21
	CPU	0.0144	0.2995	1.3718	4.3808
	RES	5.11e-07	6.96e-07	6.96e-07	6.81e-07

Table 2: When $(\sigma_1, \sigma_2) = (10, 100)$, numerical results of Example 6.2 for different iteration methods.

Method	m	16	32	48	64
	$\alpha_{\exp}$	0.74	0.9	0.9	1
PMHSS	IT	63	74	77	79
	CPU	0.0415	0.9304	4.8985	16.3990
	RES	9.12e-07	8.77e-07	9.52e-07	8.62e-07
	$\alpha_{\exp}$	1	1	1	1
	$\omega_{\exp}$	1.25	1.3	1.3	1.32
PPNHSS	IT	32	34	36	36
	CPU	0.0191	0.4823	2.2656	7.4313
	RES	8.75e-07	9.80e-07	7.99e-07	7.52e-07
	$\alpha_{\exp}$	0.35	0.09	0.04	0.022
MSNS	IT	7	8	9	10
	CPU	0.0057	0.1303	0.6378	2.2171
	RES	1.87e-07	1.02e-07	1.67e-07	3.41e-07
	$\alpha_{\exp}$	0.17	0.08	0.055	0.042
DSS	IT	42	81	116	151
	CPU	0.0275	1.1052	7.8307	33.9649
	RES	7.92e-07	9.71e-07	9.25e-07	9.79e-07
	$\alpha_{\exp}$	0.68	0.66	0.66	0.66
	$\omega_{\exp}$	1.22	1.35	1.35	1.35
PFPAE	IT	49	53	53	53
	CPU	0.0186	0.5452	2.8217	9.4808
	RES	9.18e-07	8.37e-07	8.00e-07	9.35e-07
	$\alpha_{\exp}$	1.37	1.37	1.37	1.37
	$\beta_{\exp}$	0.72	0.72	0.72	0.72
	$\omega_{\exp}$	0.88	0.88	0.88	0.88
GDSSOR	IT	15	15	15	15
	CPU	0.0094	0.2013	0.8945	2.9131
	RES	3.23e-07	4.02e-07	4.61e-07	4.10e-07
	α_{pre}	1.6	1.8	1.85	1.85
	β_{pre}	0.6250	0.5556	0.5405	0.5405
	ω_{pre}	0.8304	0.8223	0.8215	0.8190
GDSSOR	IT	17	17	17	17
	CPU	0.0103	0.2196	1.0047	3.2237
	RES	6.19e-07	8.86e-07	9.51e-07	9.41e-07
	α^*	1.2779	1.3236	1.3323	1.3359
	β^*	0.7825	0.7555	0.7506	0.7485
	ω_{pre}	0.7789	0.7483	0.7418	0.7396
GDSSOR	IT	20	21	22	22
	CPU	0.0135	0.2878	1.6043	4.5364
	RES	6.08e-07	9.76e-07	6.05e-07	6.64e-07

Table 3: Numerical results of Example 6.3 for different iteration methods.

Method	m	16	32	48	64
	$\alpha_{\exp}$	0.5	0.5	0.5	0.5
PMHSS	IT	61	60	60	60
	CPU	0.0373	0.8555	3.8803	15.0931
	RES	8.09e-07	9.92e-07	9.78e-07	9.78e-07
	$\alpha_{\exp}$	0.6	0.6	0.6	0.6
	$\omega_{\exp}$	2.8	1.8	1.5	1.4
PPNHSS	IT	11	17	21	24
	CPU	0.0096	0.2734	1.4115	4.8246
	RES	9.68e-07	6.30e-07	6.71e-07	8.93e-07
	$\alpha_{\exp}$	0.7	0.35	0.25	0.2
MSNS	IT	132	263	368	488
	CPU	0.1219	4.0745	28.3546	124.5053
	RES	9.65e-07	9.97e-07	9.71e-07	9.89e-07
	$\alpha_{\exp}$	0.23	0.23	0.22	0.23
DSS	IT	28	28	26	27
	CPU	0.0179	0.3826	1.6330	5.6305
	RES	6.63e-07	9.46e-07	8.21e-07	8.84e-07
	$\alpha_{\exp}$	0.95	0.85	0.78	0.8
	$\omega_{\exp}$	3	1.9	1.6	1.4
PFPAE	IT	21	29	36	41
	CPU	0.0081	0.3024	1.7502	6.7512
	RES	7.72e-07	8.85e-07	8.86e-07	9.05e-07
	$\alpha_{\exp}$	1.85	1.75	1.4	1.4
	$\beta_{\exp}$	0.27	0.5	0.5	0.5
	$\omega_{\exp}$	0.982	0.96	0.93	0.89
GDSSOR	IT	8	10	12	14
	CPU	0.0048	0.1336	0.7375	2.6482
	RES	9.52e-07	9.26e-07	5.88e-07	5.29e-07
	α_{pre}	3.5	2.5	2	1.8
	β_{pre}	0.2857	0.4	0.5	0.5556
	ω_{pre}	0.9705	0.9329	0.8958	0.8753
GDSSOR	IT	9	12	14	15
	CPU	0.0055	0.1627	0.9019	2.9402
	RES	7.78e-07	3.63e-07	3.37e-07	5.17e-07
	α^*	3.0022	1.9786	1.6153	1.4368
	β^*	0.3331	0.5054	0.6191	0.6960
	ω_{pre}	0.9559	0.8938	0.8526	0.8269
GDSSOR	IT	10	14	16	17
	CPU	0.0068	0.1946	1.0950	3.5471
	RES	8.50e-07	3.96e-07	4.10e-07	8.93e-07

Table 4: Numerical results of Example 6.4 for different iteration methods.

Method	m	16	32	48	64
	$\alpha_{\exp}$	1.3	1.3	1.3	1.3
PMHSS	IT	43	43	43	43
	CPU	0.0326	0.6504	2.9208	8.8443
	RES	8.29e-07	9.16e-07	9.33e-07	9.25e-07
	$\alpha_{\exp}$	0.25	0.25	0.25	0.25
	$\omega_{\exp}$	0.65	0.65	0.65	0.65
PPNHSS	IT	10	11	11	11
	CPU	0.0064	0.1483	0.6920	2.0652
	RES	2.74e-07	2.51e-07	4.91e-07	6.69e-07
	$\alpha_{\exp}$	1.5	1.05	0.85	0.75
MSNS	IT	60	85	105	122
	CPU	0.0571	1.4068	8.9291	30.6964
	RES	8.89e-07	9.97e-07	9.36e-07	9.70e-07
	$\alpha_{\exp}$	0.5	0.5	0.5	0.48
DSS	IT	14	14	15	15
	CPU	0.0092	0.2060	0.9632	2.8916
	RES	4.86e-07	6.23e-07	5.84e-07	6.42e-07
	$\alpha_{\exp}$	0.95	0.95	0.95	0.95
	$\omega_{\exp}$	0.6	0.65	0.62	0.62
PFPAE	IT	19	19	19	20
	CPU	0.0086	0.2291	1.0020	2.9518
	RES	6.92e-07	9.06e-07	9.88e-07	5.84e-07
	$\alpha_{\exp}$	0.72	0.7	0.65	0.65
	$\beta_{\exp}$	1.36	1.5	1.7	1.7
	$\omega_{\exp}$	0.985	0.982	0.983	0.983
GDSSOR	IT	8	8	8	8
	CPU	0.0049	0.1056	0.4953	1.4284
	RES	3.45e-07	7.67e-07	6.62e-07	7.27e-07
	α_{pre}	0.6579	0.6250	0.6061	0.5882
	β_{pre}	1.52	1.6	1.65	1.7
	ω_{pre}	0.9897	0.9866	0.9850	0.9837
GDSSOR	IT	8	8	8	8
	CPU	0.0049	0.1090	0.4701	1.4612
	RES	1.27e-07	3.17e-07	5.47e-07	8.23e-07
	α^*	0.6448	0.6150	0.6030	0.5966
	β^*	1.5510	1.6260	1.6583	1.6761
	ω_{pre}	0.9889	0.9860	0.9848	0.9842
GDSSOR	IT	8	8	8	8
	CPU	0.0062	0.1089	0.5159	1.5663
	RES	1.56e-07	3.71e-07	5.85e-07	7.43e-07

Example 6.4. *[7, 8, 17] Consider the linear system of equations*

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

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

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

and the right-hand side vector b with its jth entry b_j being given by

$$b_j=\frac{(1-i)j}{\tau(j+1)^2},\ j=1,2,\ldots,n.$$

In our tests, we take $\tau=h$. Besides, we normalize coefficient matrix and right-hand side by multiplying both by h^2.

6.1. The experimental results of exact implementations

For all the tested exact iteration methods, we apply the Cholesky factorization or the LU factorization of the coefficient matrices for solving the sub-systems.

Tables 1–4 list the experimentally found optimal parameters, IT, CPU times and RES of the tested exact iteration methods for Examples 6.1–6.4 with the varying of problems size, respectively. Furthermore, the numerical results of the GDSSOR iteration method with the parameters $\alpha_{pre},\beta_{pre},\omega_{pre}$ and $\alpha^*,\beta^*,\omega_{pre}$ are also reported. To further validate the superiority of the GDSSOR iteration method to the other tested iteration methods, we depict the spectral radii of the six different tested iteration matrices with the experimentally found optimal parameters in Figure 3. In addition, we compare the spectral radius of the GDSSOR iteration method in conjunction with the experimentally found optimal parameters (denoted by 'GDSSOR-exp'), with $\alpha_{pre},\beta_{pre},\omega_{pre}$ (denoted by 'GDSSOR-pre') and with $\alpha^*,\beta^*,\omega_{pre}$ (denoted by 'GDSSOR-*') for Examples 6.1-6.4 in Figure 4.

Some observations on Tables 1-4 and Figures 3-4 are given below.

- Through numerical experiments in Tables 1-4, it can be observed that the GDSSOR iteration method perform much better than the other ones, especially when problem size increases. The exception is in Table 2 where the MSNS iteration method leads to better performance than the GDSSOR one. However, the MSNS iteration method converges slowly for Examples 6.1, 6.3 and 6.4, and its IT grows rapidly with the problem size increases.

- Compared with other tested iteration methods, the GDSSOR iteration method is not sensitive to m, in the sense the iterations barely change.

- The performances of the GDSSOR iteration method with the parameters $\alpha_{pre},\beta_{pre},\omega_{pre}$, $\alpha^*,\beta^*,\omega_{pre}$ and the experimentally found optimal parameters are comparable. Thus, the parameters $\alpha_{pre},\beta_{pre},\omega_{pre}$ and $\alpha^*,\beta^*,\omega_{pre}$ can be considered as reasonable approximations of the optimal parameters of the GDSSOR iteration method. In addition, with these parameters, the GDSSOR iteration method outperforms the other ones in terms of IT and CPU times except for the MSNS one in Table 2.

- As observed in Figure 3, the spectral radius of the GDSSOR iteration method is always smaller than those of the other ones, except for the case that the spectral radius of the GDSSOR iteration method is larger than that of the MSNS one for Example 6.2, which is in accordance with the numerical results of Tables 1-4.

- Figure 4 shows that the parameters $\alpha_{pre},\beta_{pre},\omega_{pre}$ and $\alpha^*,\beta^*,\omega_{pre}$ can be considered as reasonable approximations of the optimal parameters of the GDSSOR iteration method, and we can choose them in practical computation as substitutions for the GDSSOR one.

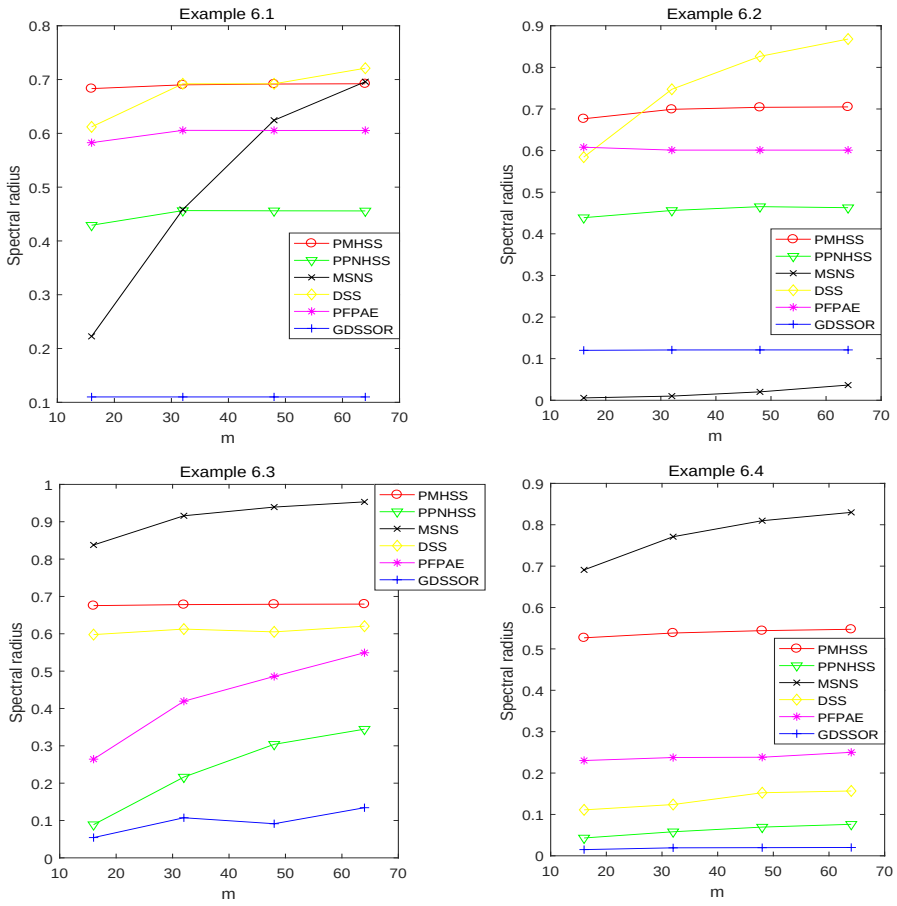

Figure 3: The spectral radii of the iteration matrices of the tested iteration methods.

Table 5: Numerical results of Example 6.1 for the tested inexact iteration methods when $(\varpi, \mu) = (\pi, 0.02)$.

Method	m	16	32	48	64
IPMHSS	$\alpha_{\exp}$	0.8	0.9	0.9	0.9
	IT	69	74	75	76
	CPU	0.1785	2.0047	12.0925	40.5837
	RES	9.33e-07	9.52e-07	9.88e-07	9.27e-07
IPPNHSS	$\alpha_{\exp}$	1.1	1.1	1.1	1.1
	$\omega_{\exp}$	1.3	1.35	1.35	1.35
	IT	32	33	33	33
	CPU	0.1019	1.2557	8.0738	30.0332
	RES	9.11e-07	9.63e-07	8.42e-07	8.65e-07
ICRI	$\alpha_{\exp}$	1	1	1	1
	IT	30	29	28	28
	CPU	0.0813	1.0511	7.1880	26.4427
	RES	8.79e-07	7.72e-07	8.92e-07	7.12e-07
IDSS	$\alpha_{\exp}$	0.11	0.09	0.08	0.08
	IT	41	48	51	51
	CPU	0.1019	1.3790	8.7841	35.0523
	RES	7.28e-07	9.60e-07	8.79e-07	8.50e-07
IPFPAE	$\alpha_{\exp}$	0.65	0.65	0.65	0.65
	$\omega_{\exp}$	1.3	1.4	1.4	1.4
	IT	50	51	50	50
	CPU	0.0828	0.9541	6.2611	24.6392
	RES	8.47e-07	9.61e-07	9.75e-07	8.88e-07
IGDSSOR	$\alpha_{\exp}$	1.4	1.4	1.4	1.4
	$\beta_{\exp}$	0.76	0.7	0.7	0.7
	$\omega_{\exp}$	0.88	0.87	0.87	0.87
	IT	15	15	15	15
	CPU	0.0589	0.8091	5.1041	20.2407
	RES	4.72e-07	6.46e-07	7.18e-07	7.67e-07

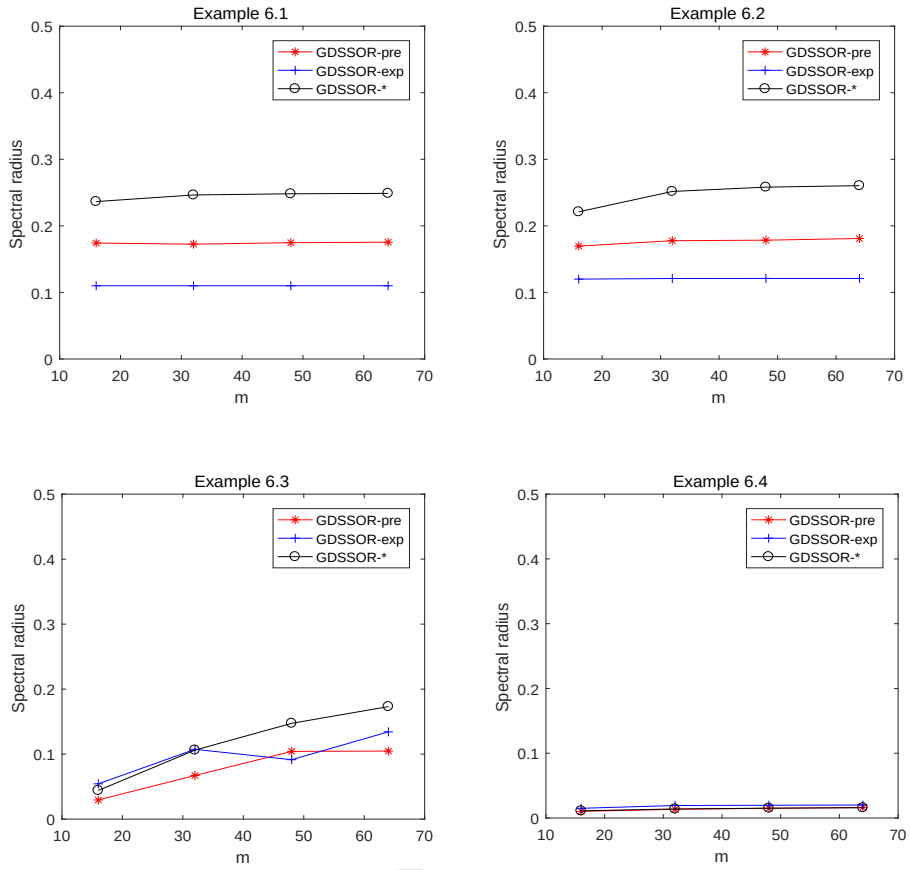

Figure 4: The spectral radii of the GDSSOR iteration matrix with different parameters.

Table 6: When $(\sigma_1, \sigma_2) = (10, 100)$, numerical results of Example 6.2 for different inexact iteration methods.

Method	m	16	32	48	64
IPMHSS	$\alpha_{\exp}$	0.74	0.9	0.9	1
	IT	63	74	77	79
	CPU	0.1725	1.8744	11.0587	41.8175
	RES	9.91e-07	8.83e-07	9.56e-07	8.65e-07
IPPNHSS	$\alpha_{\exp}$	1	1	1	1
	$\omega_{\exp}$	1.25	1.3	1.32	1.32
	IT	32	34	35	36
	CPU	0.0689	0.8711	5.2270	19.3306
	RES	8.78e-07	9.85e-07	8.95e-07	7.54e-07
ICRI	$\alpha_{\exp}$	1	1	1	1
	IT	40	38	38	37
	CPU	0.1014	0.9979	5.9074	22.2308
	RES	7.34e-07	9.80e-07	7.40e-07	8.51e-07
IDSS	$\alpha_{\exp}$	0.17	0.08	0.041	0.027
	IT	43	84	106	129
	CPU	0.1050	2.0926	12.0974	64.5551
	RES	9.07e-07	9.95e-07	9.20e-07	9.12e-07
IPFPAE	$\alpha_{\exp}$	0.68	0.68	0.68	0.66
	$\omega_{\exp}$	1.22	1.32	1.37	1.35
	IT	49	53	53	53
	CPU	0.0667	0.7491	4.3588	15.7308
	RES	9.44e-07	9.17e-07	9.10e-07	9.37e-07
IGDSSOR	$\alpha_{\exp}$	1.37	1.37	1.37	1.37
	$\beta_{\exp}$	0.72	0.72	0.72	0.72
	$\omega_{\exp}$	0.88	0.88	0.88	0.88
	IT	15	15	15	15
	CPU	0.0422	0.5326	3.2859	11.6236
	RES	3.68e-07	4.35e-07	5.24e-07	4.84e-07

Table 7: Numerical results of Example 6.3 for different inexact iteration methods.

Method	m	16	32	48	64
IPMHSS	$\alpha_{\exp}$	0.5	0.5	0.5	0.5
	IT	61	61	61	61
	CPU	0.2769	3.9190	31.6788	120.1593
	RES	8.25e-07	8.16e-07	7.96e-07	8.66e-07
IPPNHSS	$\alpha_{\exp}$	0.6	0.6	0.6	0.6
	$\omega_{\exp}$	2.8	1.9	1.55	1.4
	IT	11	16	20	24
	CPU	0.0439	0.9653	8.9415	36.4089
	RES	7.48e-07	7.24e-07	8.34e-07	9.40e-07
ICRI	$\alpha_{\exp}$	1	1	1	1
	IT	37	38	35	36
	CPU	0.1388	2.0345	17.7485	58.3309
	RES	8.60e-07	8.55e-07	7.66e-07	9.49e-07
IDSS	$\alpha_{\exp}$	0.22	0.23	0.21	0.22
	IT	28	29	27	28
	CPU	0.1190	2.0178	12.7516	51.6989
	RES	9.97e-07	8.22e-07	7.76e-07	9.22e-07
IPFPAE	$\alpha_{\exp}$	0.95	0.85	0.78	0.8
	$\omega_{\exp}$	3	1.9	1.65	1.4
	IT	21	29	37	41
	CPU	0.0451	1.0410	8.6678	38.2567
	RES	7.75e-07	9.03e-07	9.40e-07	9.30e-07
IGDSSOR	$\alpha_{\exp}$	1.84	1.83	1.4	1.4
	$\beta_{\exp}$	0.23	0.52	0.5	0.5
	$\omega_{\exp}$	0.982	0.952	0.93	0.89
	IT	9	11	12	14
	CPU	0.0336	0.6666	5.7557	24.3263
	RES	3.67e-07	5.34e-07	5.62e-07	4.86e-07

Table 8: Numerical results of Example 6.4 for different inexact iteration methods.

Method	m	16	32	48	64
IPMHSS	$\alpha_{\exp}$	1.3	1.3	1.3	1.3
	IT	43	43	43	43
	CPU	0.1417	1.7934	11.2244	38.3304
	RES	8.31e-07	9.20e-07	9.34e-07	9.27e-07
IPPNHSS	$\alpha_{\exp}$	0.1	0.12	0.13	0.13
	$\omega_{\exp}$	0.65	0.63	0.62	0.62
	IT	9	10	11	11
	CPU	0.0231	0.2538	1.5360	5.3587
	RES	9.85e-07	7.06e-07	2.61e-07	3.34e-07
ICRI	$\alpha_{\exp}$	1	1	1	1
	IT	41	41	41	41
	CPU	0.1128	1.2175	6.4714	21.1218
	RES	8.22e-07	8.94e-07	9.18e-07	9.30e-07
IDSS	$\alpha_{\exp}$	0.55	0.5	0.5	0.22
	IT	13	14	15	15
	CPU	0.0369	0.4380	2.4748	7.6301
	RES	3.36e-07	6.20e-07	5.70e-07	9.95e-07
IPFPAE	$\alpha_{\exp}$	0.95	0.95	0.95	0.95
	$\omega_{\exp}$	0.65	0.65	0.62	0.62
	IT	17	19	19	20
	CPU	0.0241	0.2616	1.4866	4.9158
	RES	9.34e-07	9.06e-07	9.95e-07	5.85e-07
IGDSSOR	$\alpha_{\exp}$	0.72	0.7	0.65	0.65
	$\beta_{\exp}$	1.36	1.5	1.7	1.7
	$\omega_{\exp}$	0.985	0.982	0.983	0.983
	IT	8	9	9	9
	CPU	0.0226	0.2477	1.4822	4.8207
	RES	8.70e-07	5.34e-07	3.00e-07	4.16e-07

6.2. The experimental results of inexact implementations

We adopt the CG method as the inner solver for the tested inexact iteration methods. For the inexact iteration methods, the stopping criterion for the CG iteration method is 10^{-2}.

We report the experimentally found optimal parameters as well as the numerical results of the inexact iteration methods for Examples 6.1–6.4 in Tables 5–8, respectively.

By comparing the results in Tables 5–8, it can be clearly seen that although all tested inexact iteration methods can successfully compute approximate solutions satisfying the prescribed stopping criterion. The IGDSSOR iteration method always outperforms the other five ones in terms of IT and CPU times, and the advantage of the IGDSSOR iteration method becomes more pronounced as the system size increases. Besides, with problem size increases, the IT of the IGDSSOR iteration method keeps constant or grows very slowly.

7. Conclusions

For the large sparse complex symmetric linear systems, we first generalize the double-step scale splitting (DSS) iteration method to obtain the generalized DSS (GDSS) one and, based on it, we further establish the generalized double steps scale-successive-overrelaxation (GDSSOR) iteration method by applying the SOR acceleration scheme for the GDSS iteration. The convergence properties, optimal parameters and a practical way for the choice of iteration parameters of the GDSSOR iteration method are studied in details. The inexact version of the GDSSOR iteration method and its convergent condition are also derived. Moreover, the presented numerical experiments show that the GDSSOR method is superior to some existing ones in terms of the iterations and CPU times.

Lastly, what we want to point out here is that although the practical choice of the parameters α and β satisfy $\alpha\beta = 1$ has been obtained, they depend on the extreme eigenvalues of the matrix $W^{-1}T$, which are difficult to be obtained when the size of $W^{-1}T$ is large enough. Hence we should investigate the more practical choice of the parameters α and β in our further work.

Acknowledgments

We would like to express our sincere thanks to the anonymous reviewers for their valuable suggestions and constructive comments which greatly improved the presentation of this paper.

References

[1] S. R. Arridge. Optical tomography in medical imaging. *Inverse Problems*, 15:41–93, 1999.

[2] Z.-Z. Bai. On the convergence of additive and multiplicative splitting iterations for systems of linear equations. *J. Comput. Appl. Math.*, 154:195–214, 2003.

[3] Z.-Z. Bai. Several splittings for non-Hermitian linear systems. *Science in China Series A: Mathematics*, 51:1339–1348, 2008.

[4] Z.-Z. Bai. 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] Z.-Z. Bai. On SSOR-like preconditioners for non-Hermitian positive definite matrices. *Numer. Linear Algebra Appl.*, 23:37–60, 2016.

[6] Z.-Z. Bai. Quasi-HSS iteration methods for non-Hermitian positive definite linear systems of strong skew-Hermitian parts. *Numer. Linear Algebra Appl.*, DOI:10.1002/nla.2116, 2017.

[7] Z.-Z. Bai, M. Benzi, and F. Chen. Modified HSS iteration methods for a class of complex symmetric linear systems. *Computing*, 87:93–111, 2010.

[8] Z.-Z. Bai, M. Benzi, and F. Chen. On preconditioned MHSS iteration methods for complex symmetric linear systems. *Numer. Algorithms*, 56:297–317, 2011.

[9] Z.-Z. Bai and G. H. Golub. Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. *IMA J. Numer. Anal.*, 27:1–23, 2007.

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

[11] Z.-Z. Bai, G. H. Golub, and M. K. Ng. On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. *Numer Linear Algebra Appl.*, 14:319–335, 2007.

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

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

[14] D. Bertaccini. Efficient solvers for sequences of complex symmetric linear systems. *Electron. Trans. Numer. Anal.*, 18:49–64, 2004.

[15] F. Chen. On choices of iteration parameter in HSS method. *Appl. Math. Comput.*, 271:832–837, 2015.

[16] M. Dehghan, M. Dehghani-Madiseh, and M. Hajarian. A generalized preconditioned MHSS method for a class of complex symmetric linear systems. *Math. Model. Anal.*, 18:561–576, 2013.

[17] D. Hezari, V. Edalatpour, and D. K. Salkuyeh. Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations. *Numer. Linear Algebra Appl.*, 22:761–776, 2015.

[18] D. Hezari, D. K. Salkuyeh, and V. Edalatpour. A new iterative method for solving a class of complex symmetric system of linear equations. *Numer. Algorithms*, 73:927–955, 2016.

[19] V. E. Howle and S. A. Vavasis. An iterative method for solving complex-symmetric systems arising in electrical power modeling. *SIAM J. Matrix Anal. Appl.*, 26:1150–1178, 2005.

[20] Y.-M. Huang. A practical formula for computing optimal parameters in the HSS iteration methods. *J. Comput. Appl. Math.*, 255:142–149, 2014.

[21] Z.-G. Huang, L.-G. Wang, Z. Xu, and J.-J. Cui. An efficient two-step iterative method for solving a class of complex symmetric linear systems. *Comput. Math. Appl.*, 75:2473–2498, 2018.

[22] C.-X. Li and S.-L. Wu. A single-step HSS method for non-Hermitian positive definite linear systems. *Appl. Math. Lett.*, 44:26–29, 2015.

[23] L. Li, T.-Z. Huang, and X.-P. Liu. Modified Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems. *Numer. Linear Algebra Appl.*, 14:217–235, 2007.

[24] X. Li, A.-L. Yang, and Y.-J. Wu. Lopsided PMHSS iteration method for a class of complex symmetric linear systems. *Numer. Algorithms*, 66:555–568, 2014.

[25] B. Poirier. Efficient preconditioning scheme for block partitioned matrices with structured sparsity. *Numer. Linear Algebra Appl.*, 7:715–726, 2000.

[26] H. N. Pour and H. S. Goughery. New Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems. *Numer. Algorithms*, 69:207–225, 2015.

[27] M. Pourbagher and D. K. Salkuyeh. On the solution of a class of complex symmetric linear systems. *Appl. Math. Lett.*, 76:14–20, 2018.

[28] D. Serre. *Matrices: Theory and Applications.* Springer, New York, 2002.

[29] J. Wang, X.-P. Guo, and H.-X. Zhong. Accelerated GPMHSS Method for Solving Complex Systems of Linear Equations. *East Asian J. Appl. Math.*, 7:143–155, 2017.

[30] T. Wang, Q.-Q. Zheng, and L.-Z. Lu. A new iteration method for a class of complex symmetric linear systems. *J. Comput. Appl. Math.*, 325:188–197, 2017.

[31] S.-L. Wu. Several variants of the Hermitian and skew-Hermitian splitting iteration method to solve a class of complex symmetric linear systems. *Numer. Linear Algebra Appl.*, 22:338–356, 2015.

[32] Y.-J. Wu, X. Li, and J.-Y. Yuan. A non-alternating preconditioned HSS iteration method for non-Hermitian positive definite linear systems. *Comput. Appl. Math.*, 36:367–381, 2017.

[33] X.-Y. Xiao and X. Wang. A new single-step iteration method for solving complex symmetric linear systems. *Numer. Algorithms*, DOI: 10.1007/s11075-017-0393-y, 2017.

[34] X.-Y. Xiao, X. Wang, and H.-W. Yin. Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems. *Comput. Math. Appl.*, 74:2269–2280, 2017.

[35] X.-Y. Xiao, X. Wang, and H.-W. Yin. Efficient preconditioned NHSS iteration methods for solving complex symmetric linear systems. *Comput. Math. Appl.*, 75:235–247, 2018.

[36] X.-Y. Xiao and H.-W. Yin. Efficient parameterized HSS iteration methods for complex symmetric linear systems. *Comput. Math. Appl.*, 73:87–95, 2017.

[37] A.-L. Yang. Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners. *Numer. Linear Algebra Appl.*, DOI:10.1002/nla.2169, 2018.

[38] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

[39] M.-L. Zeng and C.-F. Ma. A parameterized SHSS iteration method for a class of complex symmetric system of linear equations. *Comput. Math. Appl.*, 71:2124–2131, 2016.

[40] Z. Zheng, F.-L. Huang, and Y.-C. Peng. Double-step scale splitting iteration method for a class of complex symmetric linear systems. *Appl. Math. Lett.*, 73:91–97, 2017.