RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2019.45.187

ScienceAsia 45 (2019): **187–193**

A modified SSOR iterative method for a class of block two-by-two linear systems

Yajun Xie[a,b], **Changfeng Ma**[a,*]

[a] College of Mathematics and Informatics & FJKLMAA, Fujian Normal University, Fuzhou 350117 China
[b] Department of Mathematics and Physics, Fujian Jiangxia University, Fuzhou 350108 China

*Corresponding author, e-mail: macf@fjnu.edu.cn

Received 16 Aug 2018
Accepted 9 Apr 2019

ABSTRACT: An interesting phenomenon when it comes to solving a class of complex linear systems is that the modified symmetric successive overrelaxation (MSSOR) method with two parameters seems to enjoy the fine convergence performance over symmetric successive overrelaxation (SSOR) which is recently proposed by Liang [19] and Bai [30]. Owing to the flexible selection of the parameters, the MSSOR method possesses the valuable role to solve some large scale linear systems including the complex situation. Meanwhile, we also establish an accelerated MSSOR (AMSSOR) which can offer a meaningful improvement on MSSOR method. This has been illustrated experimentally and shown theoretically in this study. Our results may provide an analytical justification for popularity of SSOR and its accelerated version as efficient solvers for some linear systems.

KEYWORDS: MSSOR method, AMSSOR method, optimal convergence factor, numerical test

MSC2010: 65F10

INTRODUCTION

We consider the following complex linear system system

$$\mathscr{A} f = g, \tag{1}$$

where $\mathscr{A} = W + iT \in \mathbb{C}^{n\times n}$, $f = u + iv \in \mathbb{C}^n$, $g = p + iq \in \mathbb{C}^n$ with $u, v, p, q \in \mathbb{R}^n$ and $W, T \in \mathbb{R}^{n\times n}$ are symmetric with at least one of them being positive definite. Without loss of generality, we assume that W is positive definite. The linear system (1) can be written as

$$Ax \equiv \begin{pmatrix} W & -T \\ T & W \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix} \equiv b. \tag{2}$$

In recent years, a variety of iterative techniques are investigated to solve all kinds of linear systems including the complex situation [1–14, 21–23], such as saddle-point problem and various generalized forms [16, 18, 26]. In particular, the SOR, SSOR iterative schemes and their variants are presented for solving some large and sparse linear systems [17]. For example, Bai et al presented the GSOR [29] for augmented linear systems and SSOR-like precondition for non-Hermitian positive definite matrices [30], Zhang and Liang proposed the GSSOR and MUSOR for saddle-point problem [20, 28]. Recently, Wang et al studied a new SSOR-like method with four parameters for the augmented systems, analysed the convergence of the method and obtained the optimal convergence factor under some suitable conditions [25].

Consider the splitting $A = D - L - U$ with

$$D = \begin{pmatrix} W & 0 \\ 0 & W \end{pmatrix},\ L = \begin{pmatrix} 0 & 0 \\ -T & 0 \end{pmatrix},\ U = \begin{pmatrix} 0 & T \\ 0 & 0 \end{pmatrix}.$$

In Ref. 19, Liang and Zhang proposed the following efficient SSOR iterative scheme

$$\begin{aligned} (D-\omega L)x^{(k+\frac{1}{2})} &= \big((1-\omega)D + \omega U\big)x^{(k)} + \omega b, \\ (D-\omega U)x^{(k+1)} &= \big((1-\omega)D + \omega L\big)x^{(k+\frac{1}{2})} + \omega b, \end{aligned} \tag{3}$$

for solving the linear system (1), where $x = (u^{\mathrm{T}}, v^{T}) \in \mathbb{R}^{2n}$, and ω is a positive scalar.

Inspired by these jobs [19, 20, 25, 28], we establish the modified SSOR (MSSOR) iterative

$$\begin{aligned} (D-\omega L)x^{(k+\frac{1}{2})} &= \big((1-\omega)D + \omega U\big)x^{(k)} + \omega b, \\ (D-\tau U)x^{(k+1)} &= \big((1-\tau)D + \tau L\big)x^{(k+\frac{1}{2})} + \tau b, \end{aligned} \tag{4}$$

where $x = (u^{\mathrm{T}}, v^{\mathrm{T}}) \in \mathbb{R}^{2n}$, and ω and τ are the positive scalars. Apparently, if we select the parameter $\tau = \omega$, the proposed method is reduced to the SSOR method [19].

THE MSSOR ITERATIVE METHOD

We now describe briefly the MSSOR approach. In practice, iterative (4) generates

$$\begin{pmatrix} u^{(k+1)} \\ v^{(k+1)} \end{pmatrix} = \mathbb{H}_{\omega\tau} \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} + \mathbb{M}_{\omega\tau}^{-1} b, \tag{5}$$

where

$$\mathbb{H}_{\omega\tau} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}, \quad \mathbb{M}_{\omega\tau}^{-1} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}, \tag{6}$$

$$\begin{aligned}
H_{11} &= (1-\omega)(1-\tau)I \\
&\quad -\tau(1-\omega)(\tau+\omega-\tau\omega)(W^{-1}T)^2, \\
H_{12} &= (1-\tau)(\tau+\omega-\tau\omega)W^{-1}T \\
&\quad -\tau\omega(\tau+\omega-\tau\omega)(W^{-1}T)^3, \\
H_{21} &= -(1-\omega)(\tau+\omega-\tau\omega)W^{-1}T, \\
H_{22} &= (1-\omega)(1-\tau)I+\omega(\tau+\omega-\tau\omega)(W^{-1}T)^2, \\
M_{11} &= \omega(1-\tau)W^{-1}-\omega\tau^2(W^{-1}T)^2W^{-1} \\
&\quad -\omega^2\tau(1-\tau)(W^{-1}T)^2W^{-1}+\tau W^{-1}, \\
M_{12} &= \omega\tau(1-\tau)W^{-1}TW^{-1}+\tau^2W^{-1}TW^{-1}, \\
M_{21} &= -\omega\tau W^{-1}TW^{-1}-(1-\tau)\omega^2W^{-1}TW^{-1}, \\
M_{22} &= \omega(1-\tau)W^{-1}+\tau W^{-1}.
\end{aligned}$$

By some calculations, we can obtain the following algorithm for solving the system (2).

Algorithm 1 MSSOR method.

Step 1: Input $W, T \in \mathbb{R}^{n\times n}$ and $p, q \in \mathbb{R}^n$. Given initial guesses $v^0, u^0 \in \mathbb{R}^n$, arbitrary small positive number ε, and positive scalars ω, $\tau \in (0,2)$. Set $k=0$.

Step 2: If $r_k := \|Ax^k - b\| < \varepsilon$, stop, where $x^k = ((u^k)^{\mathrm{T}}, (v^k)^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{2n}$; otherwise, go to Step 3.

Step 3: Compute $v^{k+1} = (1-\omega)(1-\tau)v^k - (\tau+\omega-\tau\omega)W^{-1}((1-\omega)Tu^k + \omega TW^{-1}(Tv^k+p)-q)$.

Step 4: Compute $u^{k+1} = (1-\omega)(1-\tau)u^k + W^{-1}(\omega(1-\tau)Tv^k + \tau Tv^{k+1} + (\tau+\omega-\tau\omega)p)$.

Step 5: Set $k := k+1$, return to Step 2.

THE OPTIMAL PARAMETERS OF MSSOR METHOD

In this section, we will give a way to choose the optimal parameters ω_{opt} and τ_{opt} for Algorithm 1. Eventually, we are surprised to discover that the optimal convergence factor of the proposed method may be the same best effect as one in Ref. 19. However, owing to more flexible and wide selection for the parameters, the proposed approach leads to the favourable convergence results in numerical tests process which will be verified later. Firstly, we show the following well-known lemmas.

Lemma 1 (Ref. 27) *Both roots of the real quadratic equation $x^2 - bx + c = 0$ satisfy the modulus less than one if and only if $|c| < 1$ and $|b| < 1 + c$.*

Since the matrix W is positive definite and T is a symmetric matrix, thus the matrix $S := W^{-1}T$ is similar to $W^{-1/2}TW^{-1/2}$. As a result, it generates the following lemma.

Lemma 2 (Ref. 24) *Suppose that matrix W and T be symmetric positive definite and symmetric, respectively. Then the eigenvalues of the matrix $S := W^{-1}T$ are all real.*

We now take a series of similar transformations for the iterative matrix $\mathbb{H}_{\omega\tau}$.

$$Q\mathbb{H}_{\omega\tau}Q^{-1} = \begin{pmatrix} \tilde{H}_{11} & \tilde{H}_{12} \\ \tilde{H}_{21} & \tilde{H}_{22} \end{pmatrix} \triangleq \widetilde{\mathbb{H}}_{\omega\tau}, \tag{7}$$

where

$$Q := \begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix},$$

U is unitary.

By some calculations of matrix permutations, we easily obtain that the matrix $\widetilde{\mathbb{H}}_{\omega\tau}$ is similar to $\hat{H}_{\omega\tau}$ which is constructed by some 2-by-2 matrices acting as the diagonal element. Furthermore, the two-order matrix denoted by $(\hat{H}_{\omega\tau})_{ii}$, the ith diagonal block of $\hat{H}_{\omega\tau}$, $i = 1, 2, \ldots, n$, is also similar to

$$\begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}, \tag{8}$$

where

$$\begin{aligned}
h_{11} &= (1-\tau)(1-\omega) - \tau(1-\omega)(\tau+\omega-\tau\omega)\mu_i^2, \\
h_{12} &= (1-\tau)(\tau+\omega-\tau\omega)\mu_i - \tau\omega(\tau+\omega-\tau\omega)\mu_i^3, \\
h_{21} &= -(1-\omega)(\tau+\omega-\tau\omega)\mu_i, \\
h_{22} &= (1-\tau)(1-\omega) - \omega(\tau+\omega-\tau\omega)\mu_i^2,
\end{aligned}$$

and $\mu_1, \ldots, \mu_n$ are the eigenvalues of $W^{-1}T$.

The eigenvalues of the above two-order matrix can be determined by the following real quadratic equation

$$\begin{aligned}
\lambda^2 - \big(2(1-\tau)(1-\omega) - (\tau+\omega-\tau\omega)^2\mu_i^2\big)\lambda& \\
+ (1-\tau)^2(1-\omega)^2 = 0.&
\end{aligned} \tag{9}$$

According to (7) and (8), we see that λ's are also the eigenvalues of the iterative matrix $\mathbb{H}_{\omega\tau}$. From (9), we obtain

$$\lambda - (1-\tau)(1-\omega) = \pm(\tau+\omega-\tau\omega)\mu_i\sqrt{-\lambda}. \tag{10}$$

Theorem 1 *Let the matrices W, T be symmetric positive definite and symmetric, respectively, and $S := W^{-1}T$. Then the MSSOR for the linear system (2) is convergent if the following conditions hold:*
(i) $\tau,\omega \in (1,2)$ *or* $\tau,\omega \in (0,1)$, *when* $\rho(S) \leqslant 1$;
(ii) $\tau < ((1+\rho(S))\omega - 2)/(1+\rho(S))(\omega-1)$, *when* $\rho(S) > 1$ *and* $0 < \omega < 1$;
(iii) $\tau > ((1+\rho(S))\omega - 2)/(1+\rho(S))(\omega-1)$, *when* $\rho(S) > 1$ *and* $1 < \omega < 2$.

Proof: By Lemma 1, we obtain that the roots of the quadratic equations (9) satisfy $|\lambda| < 1$ if and only if

$$(1-\tau)^2(1-\omega)^2 < 1 \tag{11}$$

and

$$\left|2(1-\tau)(1-\omega)-(\tau+\omega-\tau\omega)^2\mu_i^2\right| < 1+(1-\tau)^2(1-\omega)^2. \tag{12}$$

It follows from (11) that

$$-1 < (1-\tau)(1-\omega) < 1. \tag{13}$$

Clearly, (13) holds if

$$0 < \tau < 2, \quad 0 < \omega < 2. \tag{14}$$

We now discuss (12) under the condition (14) being satisfied. In fact, by (12), we have

$$(\tau+\omega-\tau\omega)^2\mu_i^2 < \big(1+(1-\tau)(1-\omega)\big)^2. \tag{15}$$

Notice that $\mu_i \leqslant \rho(S)$. Hence (15) holds if

$$1+(1-\tau)(1-\omega) > (\tau+\omega-\tau\omega)\rho(S). \tag{16}$$

Next we consider three cases: (a) $\rho(S) \leqslant 1$; (b) $\rho(S) > 1$ and $0 < \omega < 1$; (c) $\rho(S) > 1$ and $1 < \omega < 2$.

Case (a): This case implies that (16) holds if

$$1+(1-\tau)(1-\omega) > (\tau+\omega-\tau\omega).$$

After simple calculation, one gets that $\tau-1$ and $\omega-1$ have the same positive (or negative) sign, that is, $\tau,\omega \in (1,2)$ or $\tau,\omega \in (0,1)$.

Case (b): This case indicates that (16) holds if

$$((1+\rho(S))(\omega-1))\,\tau-(1+\rho(S))\,\omega+2 > 0. \tag{17}$$

Observing $0 < \omega < 1$, by (17) we have

$$\tau < \frac{(1+\rho(S))\omega-2}{(1+\rho(S))(\omega-1)}.$$

Case (c): Similarly, from (17) we obtain

$$\tau > \frac{(1+\rho(S))\omega-2}{(1+\rho(S))(\omega-1)},$$

which completes the proof. □

Next, we give an important theorem to minimize the spectral radius of iteration matrix $\mathbb{H}_{\tau\omega}$.

Theorem 2 *Let the matrices W, T be symmetric positive definite and symmetric, respectively. Suppose that the conditions of Theorem 1 are satisfied. Let the two parameters satisfy $\omega = k\tau$. Then the optimal parameters of the MSSOR are*

$$\omega_{\text{opt}} = \frac{1+k \pm \sqrt{(1+k)^2-4k^2\varsigma}}{2k^2} \tag{18}$$

and

$$\tau_{\text{opt}} = \frac{1+k \pm \sqrt{(1+k)^2-4k^2\varsigma}}{2k}. \tag{19}$$

The optimal spectral radius is

$$\rho_{\text{opt}}(\mathbb{H}_{\tau\omega}) = 1-\frac{2}{\sqrt{1+\mu_{\min}^2}+1}, \tag{20}$$

where $0 < k < 1$ *and* $\varsigma = 2/(\sqrt{1+\mu_{\min}^2}+1)$.

Proof: From (9), we have the discriminant

$$\Delta = \big((1-\eta)^2\mu_i^2-4\eta\big)(1-\eta)^2\mu_i^2,$$

where $\eta = (1-\tau)(1-\omega)$, and the two roots of (9)

$$\lambda_{1,2} = \frac{2\eta-(1-\eta)^2\mu_i^2 \pm \sqrt{\Delta}}{2}.$$

If $\Delta \geqslant 0$, namely, $(1-\eta)^2\mu_i^2-4\eta \geqslant 0$, then $\lambda_{1,2} \leqslant 0$, and the spectral radius

$$\rho(\mathbb{H}_{\tau\omega}) = \max_{\eta}\left|\frac{2\eta-(1-\eta)^2\mu_i^2-\sqrt{\Delta}}{2}\right|.$$

Accordingly, the optimal spectral radius $\rho_{\text{opt}}(\mathbb{H}_{\tau\omega})$ reaches on $\Delta = 0$, that is,

$$\big((1-\eta)^2\mu_i^2-4\eta\big)(1-\eta)^2\mu_i^2 = 0. \tag{21}$$

If $\mu_i = 0$, by (10) we have $\lambda_1 = \lambda_2 = (1-\tau)(1-\omega)$. Thus the optimal parameters $\tau_{\text{opt}} = 1$ and $\omega_{\text{opt}} = 1$. If $\mu_i \neq 0$, it follows from (21) that $(1-\eta)^2\mu_i^2-4\eta = 0$ owing to $\eta \neq 1$ by (13). Thus, we obtain

$$\eta = 1-\frac{2}{\sqrt{1+\mu_i^2}+1}. \tag{22}$$

Hence

$$(1-\tau)(1-\omega)=1-\frac{2}{\sqrt{1+\mu_i^2}+1}. \tag{23}$$

It follows from (23) and the relation $\omega=k\tau$ that

$$\omega_{\text{opt}}=\frac{1+k\pm\sqrt{(1+k)^2-4k^2\varsigma}}{2k^2},$$
$$\tau_{\text{opt}}=\frac{1+k\pm\sqrt{(1+k)^2-4k^2\varsigma}}{2k},$$

where $0<k<1$, $\varsigma=2/(\sqrt{1+\mu_{\min}^2}+1)$, and the optimal spectral radius

$$\rho_{\text{opt}}(\mathbb{H}_{\tau\omega})=1-\frac{2}{\sqrt{1+\mu_{\min}^2}+1}.$$

If $\Delta<0$, the quadratic equation (9) has two conjugate complex roots and

$$|\lambda_{1,2}|=\left|\frac{2\eta-(1-\eta)^2\mu_i^2\pm\sqrt{-\Delta}\mathrm{i}}{2}\right|=|\eta|.$$

Observing that $1-\eta>0$ and combining with the precondition of $\Delta<0$, we immediately obtain

$$0<1-\eta<\frac{2}{\sqrt{1+\mu_i^2}+1},$$

i.e.,

$$0<1-\frac{2}{\sqrt{1+\mu_i^2}+1}<\eta<1.$$

Consequently,

$$|\eta|>1-\frac{2}{\sqrt{1+\mu_i^2}+1}, \quad i=1,2,\dots,n.$$

Furthermore,

$$|\eta|>1-\frac{2}{\sqrt{1+\rho(S)^2}+1}>1-\frac{2}{\sqrt{1+\mu_{\min}^2}+1}.$$

Based on the above analysis, we conclude that the optimal parameters and spectral radius satisfy (18)–(20), respectively. This completes the proof. □

Remark 1 As a matter of fact, by taking $k=1$ in (18) and (19), we can see that $\omega_{\text{opt}}=\tau_{\text{opt}}$, the optimal parameter is reduced to the single $\omega_{\text{opt}}=1\pm(\sqrt{1+\mu_{\min}^2}-1)/\mu_{\min}$ as mentioned in Ref. 19.

ACCELERATED VARIANT OF THE MSSOR METHOD

In this section, we develop the accelerated variant of the MSSOR method. The basic idea and refined analysis are similar to the discussions in Ref. 1.

By preconditioned scheme we obtain

$$\begin{pmatrix} I & I\\ -I & I\end{pmatrix}\begin{pmatrix} W & -T\\ T & W\end{pmatrix}\begin{pmatrix} u\\ v\end{pmatrix}=\begin{pmatrix} I & I\\ -I & I\end{pmatrix}\begin{pmatrix} p\\ q\end{pmatrix}\equiv\hat{b}.$$

It can be formulated as

$$\hat{A}x\equiv\begin{pmatrix} \hat{W} & -\hat{T}\\ \hat{T} & \hat{W}\end{pmatrix}\begin{pmatrix} u\\ v\end{pmatrix}=\begin{pmatrix} \hat{p}\\ \hat{q}\end{pmatrix}\equiv\hat{b}, \tag{24}$$

where

$$\hat{W}=W+T,\ \hat{T}=T-W,\ \hat{p}=p+q,\ \hat{q}=q-p. \tag{25}$$

Applying the MSSOR technique to the above linear system (24), we immediately have the accelerated version of the MSSOR (AMSSOR) method. The concrete scheme is described as follows.

Algorithm 2 (AMSSOR method)

Step 1: Input $W,T\in\mathbb{R}^{n\times n}$, $p,q\in\mathbb{R}^n$. Given initial guesses v^0, $u^0\in\mathbb{R}^n$, arbitrary small positive number ε, and positive scalars ω, $\tau\in(0,2)$. Set $k=0$.

Step 2: If $r_k:=\|\hat{A}x^k-\hat{b}\|<\varepsilon$, stop, where $x^k=((u^k)^{\mathrm{T}},(v^k)^{\mathrm{T}})^{\mathrm{T}}\in\mathbb{R}^{2n}$, otherwise, go to Step 3.

Step 3: According to formulae (24) and (25), compute $v^{k+1}=(1-\omega)(1-\tau)v^k-(\tau+\omega-\tau\omega)\hat{W}^{-1}((1-\omega)\hat{T}u^k+\omega\hat{T}\hat{W}^{-1}(\hat{T}v^k+\hat{p})-\hat{q})$.

Step 4: Compute $u^{k+1}=(1-\omega)(1-\tau)u^k+\hat{W}^{-1}(\omega(1-\tau)\hat{T}v^k+\tau\hat{T}v^{k+1}+(\tau+\omega-\tau\omega)\hat{p})$.

Step 5: Set $k:=k+1$, return to Step 2.

NUMERICAL EXPERIMENTS

In this section, numerical examples are executed to illustrate the effectiveness and robustness of the proposed methods for solving the complex linear systems (1). We compare the convergence performances of these methods with the SSOR and ASSOR methods[19] by the iteration step (IT), elapsed CPU time in seconds (CPU) and relative residual error (RES), RES $=\|b-Ax^k\|_2/\|b\|_2$. In actual computations, the running is terminated when the current iteration satisfies RES $<10^{-6}$ or if the number of iteration exceeds the prescribed iteration steps $k_{\max}=100$ in Algorithm 1 or Algorithm 2.

All numerical experiments were performed by MATLAB R2011b 7.1.3 on a PC equipped with an Intel (R) Core(TM) i7-2670QM, CPU running at

2.20GHZ with 8 GB of RAM under the Microsoft Windows 7 operating system.

In the next example, we consider the complex symmetric linear system arises in centred difference discretization of R_{22}-Padé approximations in the time integration of parabolic partial differential equations[1].

Example 1 Consider the complex linear system

$$\left[\left(K+\frac{3-\sqrt{3}}{\tau}I_{m^2}\right)+\mathrm{i}\left(K+\frac{3+\sqrt{3}}{\tau}I_{m^2}\right)\right]\hat{x}=\hat{b}\in\mathbb{R}^n,$$

where τ is the time step-size, $K = I_m \otimes V_m + V_m \otimes I_m$, and $V_m = h^{-2}\,\mathrm{tridiag}\{-1,2,-1\} \in \mathbb{R}^{m\times m}$ with $n = m^2$. K is the five-point centred difference approximation of negative Laplacian operator $L = -\Delta$ with homogeneous Dirichlet boundary conditions on uniform mesh in the unit square $[0,1]\times[0,1]$. The symbol $\otimes$ denotes the Kronecker product and $h = 1/(m+1)$ represents the discretization meshsize. In this example, we take the matrices $W = K + ((3-\sqrt{3})/\tau)I_{m^2}$ and $T = K + ((3+\sqrt{3})/\tau)I_{m^2}$. The vector $\hat{b} = (1-i)j/\tau(j+1)^2$, $j = 1,2,\dots,n$.

Next, we consider another complex symmetric linear system arises in direct analysis of an n degree of freedom (n-DOF) linear system[15].

Example 2 Consider the complex linear system

$$\left[(-\nu^2 M + K) + \mathrm{i}(\nu C_V + C_H)\right]\hat{x} = \hat{b} \in \mathbb{R}^n,$$

where K defined as in Example 1 is the stiffness matrix, M is the inertia matrix, C_V and C_H are the viscous and hysteretic damping matrices, respectively, and ν is the driving circular frequency. Here the matrices $W = -\nu^2 M + K$ and $T = \nu C_V + C_H$, where $C_H = \mu K$, $C_V = 10 I_{m^2}$, and $M = I_{m^2}$. We also choose different values of ν, μ. The right-side vector $\hat{b}$ is selected such that the exact solution of the linear system (1) is $(1+\mathrm{i})\cdot[1,1,\dots,1]^{\mathrm{T}} \in \mathbb{C}^n$.

All numerical results are shown for the various problem sizes in Tables 1–5. From these results, we see that the proposed methods MSSOR and AMSSOR keep almost the favourable convergence results with SSOR and ASSOR. In some situations, our approaches are compared to the existing methods in terms of both iteration steps and CPU time. In particular, the fact of superiority has clearly been elucidated between ASSOR and AMSSOR methods (Tables 2 and 3. The optimal parameters for SSOR and ASSOR complied with Table 1[19]. Our experiment parameters are chosen according to the numerical test effect.

CONCLUSIONS

Two efficient iterative methods are presented for solving a class of block two-by-two linear systems. To some extent, the proposed methods are regarded as generalized versions of SSOR and ASSOR, respectively. The selections of the optimal parameters are analysed in detail under a proper condition. Finally, we give some numerical examples to demonstrate that the introduced iterative algorithms are effective and workable. Meanwhile, our results may provide an analytical justification for the popularity of SSOR technique and its accelerated version as efficient solvers for some linear systems.

Acknowledgements: The work is supported by National Key Research and Development Programme of China (No. 2018YFC0603500), the New Century Training Plan of Fujian Province University, and the Educational Reform Project of Fujian Jiangxia University (No. 2017[52], J2018B013).

REFERENCES

1. Axelsson O, Kucherov A (2000) Real valued iterative methods for solving complex symmetric linear systems. *Numer Linear Algebra Appl* **7**, 197–218.
2. Axelsson O, Neytcheva M, Ahmad B (2014) A comparison of iterative methods to solve complex valued linear algebraic systems. *Numer Algorithms* **66**, 811–41.
3. Bai ZZ (2006) Structured preconditioners for nonsingular matrices of block two-by-two structures. *Math Comput* **75**, 791–815.
4. Bai ZZ (2013) Rotated block triangular preconditioning based on PMHSS. *Sci China Math* **56**, 2523–38.
5. Ren L, Xin J (2018) Almost global existence for the Neumann problem of quasilinear wave equations outside star-shaped domains in 3D. *Electron J Differ Equations* **312**, 1–22.
6. Wang B, Meng F, Fang Y (2017) Efficient implementation of RKN-type Fourier collocation methods for second-order differential equations. *Appl Numer Math* **119**, 164–78.
7. Han M, Sheng L, Zhang X (2018) Bifurcation theory for finitely smooth planar autonomous differential systems. *J Differ Equations* **264**, 3596–618.
8. Han M, Hou X, Sheng L, Wang C (2018) Theory of rotated equations and applications to a population model. *Discrete Cont Dyn Syst A* **38**, 2171–85.
9. Li F, Du G (2018) General energy decay for a degenerate viscoelastic Petrovsky-type plate equation with boundary feedback. *J Appl Anal Comput* **8**, 390–401.
10. Li M, Wang J (2018) Exploring delayed mittag-Leffler type matrix functions to study finite time stability of fractional delay differential equations. *Appl Math Comput* **324**, 254–65.

Table 1 Numerical results for Example 1 with 16×16.

Methods		SSOR	ASSOR	MSSOR	AMSSOR
Parameters		$\omega_{\mathrm{opt}}=0.33$	$\omega_{\mathrm{opt}}=0.80$	$\omega_{\exp}=0.26$, $\tau_{\exp}=0.35$	$\omega_{\exp}=1.6$, $\tau_{\exp}=1.1$
	IT	19	9	19	6
$\tau=h$	CPU	0.081	0.0396	0.0787	0.0360
	RES	6.9117×10^{-7}	5.4486×10^{-9}	6.4008×10^{-7}	1.1024×10^{-9}
	IT	18	9	19	6
$\tau=2h$	CPU	0.0913	0.0408	0.0903	0.0288
	RES	5.3516×10^{-7}	9.8011×10^{-9}	7.8282×10^{-7}	1.3915×10^{-9}
	IT	18	10	18	6
$\tau=3h$	CPU	0.0794	0.0400	0.0724	0.0257
	RES	5.3320×10^{-7}	1.6191×10^{-9}	7.6330×10^{-7}	1.4537×10^{-9}

Table 2 Numerical results for Example 1 with 32×32.

Methods		SSOR	ASSOR	MSSOR	AMSSOR
Parameters		$\omega_{\mathrm{opt}}=0.29$	$\omega_{\mathrm{opt}}=0.77$	$\omega_{\exp}=0.26$, $\tau_{\exp}=0.35$	$\omega_{\exp}=1.6$, $\tau_{\exp}=1.1$
	IT	21	10	21	6
$\tau=h$	CPU	2.5123	1.1977	2.2836	0.6241
	RES	5.6263×10^{-7}	2.2899×10^{-9}	7.0772×10^{-7}	1.4529×10^{-9}
	IT	21	10	21	6
$\tau=2h$	CPU	2.4961	1.0744	2.3419	0.6429
	RES	5.6000×10^{-7}	1.9733×10^{-9}	8.2911×10^{-7}	1.5122×10^{-9}
	IT	21	10	21	6
$\tau=3h$	CPU	2.1786	1.1978	2.1682	0.7445
	RES	5.6489×10^{-7}	2.2899×10^{-9}	8.1649×10^{-7}	1.4881×10^{-9}

Table 3 Numerical results for Example 1 with 64×64.

Methods		SSOR	ASSOR	MSSOR	AMSSOR
Parameters		$\omega_{\mathrm{opt}}=0.29$	$\omega_{\mathrm{opt}}=0.77$	$\omega_{\exp}=0.26$, $\tau_{\exp}=0.35$	$\omega_{\exp}=1.6$, $\tau_{\exp}=1.1$
	IT	21	10	21	6
$\tau=h$	CPU	105.1976	47.0798	99.8367	27.8777
	RES	5.6959×10^{-7}	2.0626×10^{-9}	8.4682×10^{-7}	1.5466×10^{-9}
	IT	21	10	21	6
$\tau=2h$	CPU	94.8650	43.9566	92.3694	27.6447
	RES	6.0926×10^{-7}	2.5794×10^{-9}	8.5401×10^{-7}	1.4636×10^{-9}
	IT	21	10	21	5
$\tau=3h$	CPU	98.0843	38.1562	97.2459	23.4235
	RES	5.6979×10^{-7}	2.8105×10^{-9}	8.4858×10^{-7}	8.7778×10^{-9}

Table 4 Numerical results for Example 2 with 16×16.

Methods		SSOR	ASSOR	MSSOR	AMSSOR
Parameters		$\omega_{\mathrm{opt}}=0.26$	$\omega_{\mathrm{opt}}=0.61$	$\omega_{\exp}=0.24$, $\tau_{\exp}=0.28$	$\omega_{\exp}=1.7$, $\tau_{\exp}=1.3$
	IT	23	11	23	10
$\nu=\pi$, $\mu=0.02$	CPU	0.137	0.513	0.1298	0.0437
	RES	9.6629×10^{-7}	2.8227×10^{-8}	5.3485×10^{-7}	5.3559×10^{-8}
	IT	23	13	23	11
$\nu=1$, $\mu=0.01$	CPU	0.1183	0.0788	0.1130	0.0722
	RES	9.6531×10^{-7}	6.9059×10^{-8}	9.4939×10^{-7}	9.6248×10^{-8}
	IT	23	11	23	10
$\nu=2$, $\mu=0.1$	CPU	0.1092	0.0572	0.1084	0.0487
	RES	9.6565×10^{-7}	7.4053×10^{-8}	9.9018×10^{-7}	8.9181×10^{-8}

Table 5 Numerical results for Example 2 with 32×32.

Methods		SSOR	ASSOR	MSSOR	AMSSOR
Parameters		$\omega_{\text{opt}} = 0.26$	$\omega_{\text{opt}} = 0.60$	$\omega_{\text{exp}} = 0.24$, $\tau_{\text{exp}} = 0.28$	$\omega_{\text{exp}} = 1.7$, $\tau_{\text{exp}} = 1.3$
	IT	24	12	23	11
$\nu = \pi$, $\mu = 0.02$	CPU	3.1287	1.5598	3.0862	1.5111
	RES	5.5262×10^{-7}	4.1854×10^{-9}	9.7797×10^{-7}	1.1270×10^{-9}
	IT	23	13	23	11
$\nu = 1$, $\mu = 0.01$	CPU	3.1022	1.5688	2.9378	1.4435
	RES	9.6529×10^{-7}	4.1506×10^{-9}	9.4926×10^{-7}	1.1979×10^{-9}
	IT	23	8	23	8
$\nu = 2$, $\mu = 0.1$	CPU	2.7483	0.9733	2.6972	0.9590
	RES	9.6536×10^{-7}	5.0130×10^{-9}	9.4996×10^{-7}	5.8064×10^{-9}

11. Wang B, Wu X, Meng F (2017) Trigonometric collocation methods based on Lagrange basis polynomials for multi-frequency oscillatory second order differential equations. *J Comput Appl Math* **313**, 185–201.
12. Wang B (2017) Exponential Fourier collocation methods for solving first-order differential equations. *J Comp Math* **35**, 711–36.
13. Tian H, Han M (2017) Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems. *J Differ Equations* **263**, 7448–74.
14. Guo L, Liu L, Wu Y (2015) Existence of positive solutions for singular fractional differential equations with infinite-point boundary conditions. *Nonlinear Anal Model Control* **21**, 635–50.
15. Benzi M, Bertaccini D (2008) Block preconditioning of real-valued iterative algorithms for complex linear systems. *IMA J Numer Anal* **28**, 598–618.
16. Benzi M, Golub GH, Liesen J (2005) Numerical solution of saddle point problems. *Acta Numer* **14**, 1–137.
17. Golub GH, Wu X, Yuan JY (2001) SOR-like methods for augmented systems. *BIT Numer Math* **41**, 71–85.
18. Huang ZG, Wang LG, Xu Z, Cui JJ (2018) A modified generalized shift-splitting preconditioner for non-symmetric saddle point problems. *Numer Algorithms* **78**, 297–331.
19. Liang ZZ, Zhang GF (2016) On SSOR iterative method for a class of block two-by-two linear systems. *Numer Algorithms* **71**, 655–71.
20. Liang ZZ, Zhang GF (2014) Modified unsymmetric SOR method for saddle-point problem. *Appl Math Comput* **234**, 584–98.
21. Li XA, Zhang WH, Wu YJ (2018) On symmetric block triangular splitting iteration method for a class of complex symmetric system of linear equations. *Appl Math Lett* **79**, 131–7.
22. Najafi HS, Edalatpanah SA (2015) On the modified symmetric successive over-relaxation method for augmented systems. *Comput Appl Math* **34**, 607–17.
23. Najafi HS, Edalatpanah SA (2014) A new modified SSOR iteration method for solving augmented linear systems. *Int J Comput Math* **91**, 539–52.
24. Salkuyeh DK, Hezari D, Edalatpour V (2015) Generalized successive overrelaxation iterative method for a class of complex symmetric linear system of equations. *Int J Comput Math* **92**, 802–15.
25. Wang HD, Huang ZD (2017) On a new SSOR-like method with four parameters for the augmented systems. *East Asian J Appl Math* **7**, 82–100.
26. Xie YJ, Ma CF (2016) A modified positive-definite and skew-Hermitian splitting preconditioner for generalized saddle point problems from the Navier-Stokes equation. *Numer Algorithms* **72**, 243–58.
27. Young DM (1971) *Iterative Solution of Large Linear Systems*, Academic Press.
28. Zhang GF, Lu QH (2008) On generalized symmetric SOR method for augmented systems. *J Compt Appl Math* **219**, 51–8.
29. Bai ZZ, Parlett BN, Wang ZQ (2005) On generalized successive overrelaxation methods for augmented linear systems. *Numer Math* **102**, 1–38.
30. Bai ZZ (2016) On SSOR-like preconditioners for non-Hermitian positive definite matrices. *Numer Linear Algebra Appl* **23**, 37–60.