Filomat 39:15 (2025), 5095–5108
https://doi.org/10.2298/FIL2515095S

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A modified USSOR method for solving the rank deficient linear least squares problem

Juan Song[a,b]

[a]*School of Science, Wuxi University, Wuxi 214105, P. R. China*
[b]*Jiangsu Key Laboratory for Numerical Simulation of Large Scale Complex Systems, P. R. China*

Abstract. The USSOR method has been presented for solving the rank deficient linear least squares problem by J. Song and Y. Song (CALCOLO, 54(2017) 95-115). However, the convergence rate is relatively slow. In order to improve the convergence rate, we present the modified USSOR (MUSSOR) method for solving the rank deficient linear least squares problem. Meanwhile, the convergence and optimal parameter of the MUSSOR method are studied. Numerical examples demonstrate the effectiveness and feasibility of the proposed method.

1. Introduction

In this paper, we consider the rank deficient linear least squares problem

$$Ax = b, \tag{1}$$

where $A \in C_r^{m\times n}$, with $m \geq n$ and $\text{rank}(A) = r < n$, $b \in C^m$, $x \in C^n$ with b known and x unknown.

The linear least squares problem arises in several fields, such as signal processing, machine learning, climate model, and so on. It is particularly difficult to solve because they are often large, ill-conditioned in actual applications. When A is rectangular and of full column rank matrix, Chen [5] suggested using iterative method to solve the linear least squares problem, and augmented the rectangular linear system to a nonsingular system. In [22], the USSOR method is applied to solve the full column rank linear least squares problem. However, when A is rank deficient, Chen encountered difficulty. In sequence, Plemmons, etc [16, 19] introduce how difficulties can be overcome. Miller and Neumann [17] extend the Chen's augmentation procedure for the full column rank case to the case A is deficient rank, and first proposed a class of SOR iterative method for solving rank deficient linear least squares problem. Recently, the SOR method is also widely applied to solve absolute value equations in [12, 14]. In sequence, the AOR iterative method is applied to solve rank deficient linear least squares problem in [4, 15, 21]. Then, the symmetric SOR (SSOR) method is studied for solving rank deficient linear least squares problem in [3, 7–11, 24]. In [20, 23], the USSOR method is applied to solve the rank deficient linear least squares problem. However, when the scale of coefficient matrix A is large, the calculation speed is relatively slow.

2020 *Mathematics Subject Classification*. Primary 65F10; Secondary 65F45, 65B99.
Keywords. Linear least squares problem, MUSSOR method, Semiconvergence, Optimal parameter.
Received: 03 July 2022; Revised: 15 September 2024; Accepted: 30 March 2025
Communicated by Predrag Stanimirović
Email address: song12368@163.com (Juan Song)
ORCID iD: https://orcid.org/0009-0006-0351-5028 (Juan Song)

In order to overcome this deficiency and improve the convergence rate of the USSOR method for solving the rank deficient linear least squares problem (1), in this paper, we present the MUSSOR method and give out the convergence analysis and optimal convergence factor. Furthermore, we demonstrate the effectiveness of proposed algorithms by comparing the computing time to obtain the solution to (1). For more comparison standards, see [1].

This paper is organized as follows. In Section 2, the MUSSOR method is proposed to solve the rank deficient linear least squares problem (1). In Section 3, the optimal parameter and optimal convergence factor of the MUSSOR method is studied. In Section 4, numerical examples are given out to show that the efficiency of the theoretical analysis, and demonstrate that the MUSSOR method is far superior to the USSOR method, with corresponding improvement of 63% in computing time under some conditions. Finally, some concluding remarks are given in Section 5.

We briefly introduce some explanations of notations to be used in the paper. For $A \in C^{m\times n}$, the symbols A^*, $A^\top$, $R(A)$, $N(A)$, $\rho(A)$, $\sigma(A)$, A^+ and $\|A\|$ stand for the conjugate transpose, transpose, range space, null space, spectral radius, spectrum, Moore-Penrose generalized inverse and 2-norm of A, respectively.

2. The MUSSOR method

In this section, we develop the MUSSOR method for solving the linear system (1) and discuss its convergence.

Firstly, we derive a new equivalent linear system of the (1).

Without loss of generality, assuming A has the 4-block partitioned form in [20],

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \in C_r^{m\times n},$$

with $A_{11} \in C_r^{r\times r}$, $A_{12} \in C^{r\times(n-r)}$, $A_{21} \in C^{(m-r)\times r}$, $A_{22} = A_{21}A_{11}^{-1}A_{12} \in C^{(m-r)\times(n-r)}$. Let

$$Q = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & I_{n-r} \end{pmatrix}.$$

Then the linear system (1) is equivalent to the following equivalent systems

$$\bar{A}w = b, \tag{2}$$

where

$$\bar{A} = \begin{pmatrix} I & A_{12} \\ A_{21}A_{11}^{-1} & A_{22} \end{pmatrix}, w = Q^{-1}x.$$

Obviously, if the solution w to the linear system (2) is known, then the solution x to the linear system (1) is

$$x = Qw.$$

In the following, we discuss the numerical solution of the linear system (2).

It is well-known that the least square solution of minimal norm to the linear system (2) is $w = \bar{A}^+b$. And $y \in C^n$ is the least squares solution to (2), i.e.,

$$\|b - \bar{A}y\|_2 = \min_{w\in C^n} \|b - \bar{A}w\|_2 \tag{3}$$

if and only if [13]

$$\delta = b - \bar{A}y \tag{4}$$

satisfies relation

$$\bar{A}^*\delta = 0,$$

where $\|w\|_2 = \sqrt{\sum_{i=1}^{n}|w_i|^2}$ for any $w \in C^n$.

Let $y = (y_1^*, y_2^*)^*$, $\delta = (\delta_1^*, \delta_2^*)^*$, $b = (b_1^*, b_2^*)^*$, $y_1, b_1, \delta_1 \in C^r$, $b_2, \delta_2 \in C^{m-r}$, $y_2 \in C^{n-r}$. According to (4), y satisfies (3) if and only if

$$\mathcal{A}\bar{z} = f, \tag{5}$$

where

$$\mathcal{A} = \begin{bmatrix} I & 0 & I_r & A_{12} \\ A_{21}A_{11}^{-1} & I_{m-r} & 0 & A_{22} \\ 0 & (A_{21}A_{11}^{-1})^* & I & 0 \\ 0 & A_{22}^* & A_{12}^* & 0 \end{bmatrix}, \quad \bar{z} = \begin{bmatrix} y_1 \\ \delta_2 \\ \delta_1 \\ y_2 \end{bmatrix}, \quad f = \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix}.$$

This is a block 4×4 consistent system.

Split $\mathcal{A}$ into

$$\begin{aligned} \mathcal{A} = & \begin{bmatrix} I & 0 & 0 & 0 \\ A_{21}A_{11}^{-1} & I_{m-r} & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I_{n-r} \end{bmatrix} \\ & - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -(A_{21}A_{11}^{-1})^* & 0 & 0 \\ 0 & -A_{22}^* & -A_{12}^* & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -I_r & -A_{12} \\ 0 & 0 & 0 & -A_{22} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{n-r} \end{bmatrix} \\ = & D - \tilde{L} - \tilde{U}. \end{aligned} \tag{6}$$

Obviously, D is nonsingular.

Let

$$L = D^{-1}\tilde{L}, U = D^{-1}\tilde{U}, \tau = \omega + \hat{\omega} - \omega\hat{\omega} \neq 0.$$

Then the MUSSOR method for solving (2) can be defined by

$$\bar{z}_{i+1} = \bar{S}_{\omega,\hat{\omega}}\bar{z}_i + \bar{c}, \tag{7}$$

where

$$\bar{S}_{\omega,\hat{\omega}} = (I - \hat{\omega}U)^{-1}[\hat{\omega}L + (1-\hat{\omega})I](I - \omega L)^{-1}[\omega U + (1-\omega)I],$$

and

$$\bar{c} = \tau(I - \hat{\omega}U)^{-1}(I - \omega L)^{-1}D^{-1}f.$$

Let

$$\begin{aligned} B &= A_{21}A_{11}^{-1}, K_1 = (I_r + \frac{1-\omega}{1-\hat{\omega}}A_{12}A_{12}^*)B^*, \\ K_2 &= -(1-\hat{\omega})I + \omega\hat{\omega}B^*B + \frac{\hat{\omega}(1-\omega)}{1-\hat{\omega}}A_{12}A_{12}^* + \omega\hat{\omega}\frac{1-\omega}{1-\hat{\omega}}A_{12}A_{12}^*B^*B. \\ \tilde{K}_1 &= I - \omega\hat{\omega}(\frac{1-\omega}{1-\hat{\omega}}A_{12}A_{12}^* + I_r)B^*B, \\ \tilde{K}_2 &= \omega\hat{\omega}(\frac{1-\omega}{1-\hat{\omega}}A_{12}A_{12}^* + I_r)B^*, \tilde{K}_3 = -\hat{\omega}I + \frac{\omega\hat{\omega}}{1-\hat{\omega}}A_{12}A_{12}^*. \end{aligned}$$

Then by direct computation, it can conclude for $\hat{\omega} \neq 1$

$$\begin{aligned} & \bar{S}_{\omega,\hat{\omega}} = \\ = & \begin{bmatrix} (1-\tau)I & \hat{\omega}(1-\omega)\tau K_1 & \tau K_2 & -\tau A_{12} \\ 0 & (1-\tau)I - (1-\omega)\hat{\omega}\tau BB^* & \tau B[(1-\hat{\omega})I - \omega\hat{\omega}B^*B] & 0 \\ 0 & (\omega-1)\tau B^* & (1-\tau)I - \omega\tau B^*B & 0 \\ 0 & -\frac{(1-\omega)^2\tau}{1-\hat{\omega}}A_{22}^* & \frac{\tau(\omega-1)}{1-\hat{\omega}}A_{12}^*(I + \omega B^*B) & I \end{bmatrix}, \end{aligned} \tag{8}$$

$$\bar{c} = \tau \begin{bmatrix} \tilde{K}_1 & \tilde{K}_2 & \tilde{K}_3 & -\frac{\hat{\omega}}{1-\hat{\omega}}A_{12} \\ \omega\hat{\omega}BB^*B - B & I - \omega\hat{\omega}BB^* & \hat{\omega}B & 0 \\ \omega B^*B & -\omega B^* & I & 0 \\ \frac{\omega(1-\omega)}{1-\hat{\omega}}A_{22}^*B & -\frac{\omega(1-\omega)}{1-\hat{\omega}}A_{22}^* & -\frac{\omega}{1-\hat{\omega}}A_{12}^* & \frac{1}{1-\hat{\omega}}I \end{bmatrix} f,$$

By (8), we see

$$\sigma(\bar{S}_{\omega,\hat{\omega}}) = \{1-\tau, 1\} \cup \sigma(T_{\omega,\hat{\omega}}), \tag{9}$$

where $T_{\omega,\hat{\omega}}$ is

$$T_{\omega,\hat{\omega}} = \begin{bmatrix} (1-\tau)I - (1-\omega)\hat{\omega}\tau BB^* & \tau B[(1-\hat{\omega})I - \omega\hat{\omega}B^*B] \\ -(1-\omega)\tau B^* & (1-\tau)I - \omega\tau B^*B \end{bmatrix}. \tag{10}$$

Obviously, $\omega = \hat{\omega}$, $\hat{\omega} = 0$, the MUSSOR method (7) will be reduced to the MSSOR method, the MSOR method, respectively.

In order to more effectively utilize the MUSSOR method to obtain the solution of the (1), in the following, we discuss the semiconvergence.

According to the definition of semiconvergence in [2], we need to find that $\omega, \hat{\omega}$ satisfies the following three conditions:

(i) $\rho(\bar{S}_{\omega,\hat{\omega}}) \le 1$;

(ii) $\lambda \in \sigma(\bar{S}_{\omega,\hat{\omega}})$, $|\lambda| = 1 \Rightarrow \lambda = 1$;

(iii) $index(I - \bar{S}_{\omega,\hat{\omega}}) \le 1$

First, we look at the condition (iii). It is easy to conclude that

$$\begin{aligned} I - \bar{S}_{\omega,\hat{\omega}} &= \tau \begin{bmatrix} I & -\hat{\omega}(1-\omega)K_1 & -K_2 & A_{12} \\ 0 & I + \hat{\omega}(1-\omega)BB^* & -(1-\hat{\omega})B + \omega\hat{\omega}BB^*B & 0 \\ 0 & (1-\omega)B^* & I + \omega B^*B & 0 \\ 0 & \frac{(1-\omega)^2}{1-\hat{\omega}}A_{22}^* & \frac{\omega(1-\omega)}{1-\hat{\omega}}A_{12}^*B^*B + \frac{1-\omega}{1-\hat{\omega}}A_{12}^* & 0 \end{bmatrix} \\ &= \tau \begin{bmatrix} I & 0 & 0 \\ 0 & I & \hat{\omega}B \\ 0 & 0 & I \\ 0 & 0 & \frac{1-\omega}{1-\hat{\omega}}A_{12}^* \end{bmatrix} \begin{bmatrix} I & -\hat{\omega}(1-\omega)K_1 & -K_2 & A_{12} \\ 0 & I & -B & 0 \\ 0 & (1-\omega)B^* & I + \omega B^*B & 0 \end{bmatrix} \\ &= \bar{F}\bar{G}. \end{aligned}$$

Since $\tau \neq 0$ and

$$\begin{aligned} & det \begin{bmatrix} I & -\hat{\omega}(1-\omega)K_1 & -K_2 & A_{12} \\ 0 & I & -B & 0 \\ 0 & (1-\omega)B^* & I + \omega B^*B & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & \hat{\omega}B \\ 0 & 0 & I \\ 0 & 0 & \frac{1-\omega}{1-\hat{\omega}}A_{12}^* \end{bmatrix} \\ = \; & det \begin{bmatrix} I & -\hat{\omega}(1-\omega)K_1 & -\hat{\omega}^2(1-\omega)K_1B - K_2 + \frac{1-\omega}{1-\hat{\omega}}A_{12}A_{12}^* \\ 0 & I & (\hat{\omega}-1)B \\ 0 & (1-\omega)B^* & I + \tau B^*B \end{bmatrix} \\ = \; & det \begin{bmatrix} I & -\hat{\omega}(1-\omega)K_1 & -\hat{\omega}^2(1-\omega)K_1B - K_2 + \frac{1-\omega}{1-\hat{\omega}}CA_{12}^* \\ 0 & I & (\hat{\omega}-1)B \\ 0 & 0 & I + B^*B \end{bmatrix} \\ \neq \; & 0, \end{aligned}$$

According to results in Cline [6], *index*$(I - \bar{S}_{\omega,\hat{\omega}}) \leq 1$. Hence, (iii) follows.

From (10), it is easy to see that the submatrix $T_{\omega,\hat{\omega}}$ of the $\bar{S}_{\omega,\hat{\omega}}$ is the same as the submatrix $T_{\omega,\hat{\omega}}$ of the $S_{\omega,\hat{\omega}}$ [20]. Based on the semiconvergence conditions, (9) and the above discussions, it is easy to conclude the following semiconvergence conditions of MUSSOR method, which is consistent with USSOR method [20].

Theorem 2.1. *The MUSSOR method for solving the rank deficient linear least squares problem is semiconvergent if* $\omega \neq 1$, $\hat{\omega} \neq 1$ *and*

Case I: $\mu^* > 1, 0 < \tau < \frac{2}{1+\mu^*}$;

Case II: $\mu^* \leq 1, 0 < \tau < 1$.

where μ_i, $i = 1, 2, ..., m$, *is the eigenvalue of* $J_s = \begin{bmatrix} 0 & B \\ -B^* & 0 \end{bmatrix}$, $\mu^* = \max_i |\mu_i| = \|B\|_2 = \|A_{21}A_{11}^{-1}\|_2$.

Although the semiconvergence conditions are consistent, the MUSSOR method for solving problem (1) is much faster than the USSOR method [20]. This will be demonstrated in both the following algorithm and numerical examples.

When $\omega = \hat{\omega}$, we can obtain the following semiconvergence conditions of the MSSOR method for solving the rank deficient linear least squares problem, which is consistent with SSOR method [24].

Corollary 2.2. *The MSSOR iteration method for solving the rank deficient linear least squares problem is semiconvergent if*

Case I: $\mu^* > 1, 0 < \omega < 1 - \sqrt{\frac{\mu^*-1}{\mu^*+1}}$, *or* $1 + \sqrt{\frac{\mu^*-1}{\mu^*+1}} < \omega < 2$;

Case II: $\mu^* \leq 1, 0 < \omega < 1$, *or* $1 < \omega < 2$.

where $\mu^* = \max_i |\mu_i| = \|B\|_2 = \|A_{21}A_{11}^{-1}\|_2$.

Although the semiconvergence conditions are consistent, the MSSOR method for solving problem (1) is much faster than the SSOR method [24]. This will be demonstrated in both the following algorithm and numerical examples.

When $\hat{\omega} = 0$, we can obtain the following semiconvergence conditions of the MSOR method for solving the rank deficient linear least squares problem.

Corollary 2.3. *The MSOR iteration method for solving the rank deficient linear least squares problem is semiconvergent if*

Case I: $\mu^* > 1, 0 < \omega < \frac{2}{1+\mu^*}$,

Case II: $\mu^* \leq 1, 0 < \omega < 1$.

where $\mu^* = \max_i |\mu_i| = \|B\|_2 = \|A_{21}A_{11}^{-1}\|_2$.

In the following, we give out the algorithm of the MUSSOR method for solving the rank deficient linear least squares problem (1). Firstly, we will show the relationship between the solutions of (2) and (5). Let $z(x_0)$ be the solution of (5). Then the solution can be represented as $z(x_0) = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \\ \bar{z}_3 \\ \bar{z}_4 \end{bmatrix} = \mathcal{A}^+ f + e(x_0)$, where $e(x_0) \in N(\mathcal{A})$. In order to compute the solution of (2) by (5), we need the following lemma.

Lemma 2.4. *Let $\bar{S}_{\omega,\hat{\omega}}$ be the MUSSOR iteration matrix induced by the splitting (6) of $\mathcal{A}$. Then, we have*

$$(I - \bar{S}_{\omega,\hat{\omega}})^{+}(I - \bar{S}_{\omega,\hat{\omega}})z(x_0) = \mathcal{A}^{+} f, \tag{11}$$

where $\omega \neq 1$, $\hat{\omega} \neq 1$, $\omega + \hat{\omega} - \omega\hat{\omega} \neq 0$

Proof. The proof is similar to the Theorem 2.5 in [20]. Here we omit it. □

By simple computations, it is easy to obtain that

$$(I - \bar{S}_{\omega,\hat{\omega}})^{+}(I - \bar{S}_{\omega,\hat{\omega}}) = \mathcal{A}^{+}\mathcal{A} = \tag{12}$$
$$\begin{bmatrix} (I + A_{12}A_{12}^{*})^{-1} & 0 & 0 & (I + A_{12}A_{12}^{*})^{-1}A_{12} \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ A_{12}^{*}(I + A_{12}A_{12}^{*})^{-1} & 0 & 0 & A_{12}^{*}(I + A_{12}A_{12}^{*})^{-1}A_{12} \end{bmatrix}.$$

By [17, Lemma 2.4], (11) and (12), we can obtain

$$\bar{u}_1 = (I + A_{12}A_{12}^{*})^{-1}(\bar{z}_1 + A_{12}\bar{z}_4),\ \bar{u}_4 = A_{12}^{*}\bar{u}_1. \tag{13}$$

By simple computations, we give out the following algorithm of the MUSSOR method for solving the rank deficient linear least squares problem.

Algorithm 2.5.

(1) Give an initial vector $y_0 \in C^n$;

(2) Using (4) to compute $\delta_1^{(0)}$, $\delta_2^{(0)}$, $y_1^{(0)}$, $y_2^{(0)}$;

(3) For $k = 0, 1, 2, \cdots$ until convergence, do

$$\begin{aligned} \delta_1^{(k+1)} &= -(1-\omega)\tau B^{*}\delta_2^{(k)} + [(1-\tau)I - \omega\tau B^{*}B]\delta_1^{(k)} + \omega\tau B^{*}(Bb_1 - b_2), \\ \delta_2^{(k+1)} &= (1-\tau)\delta_2^{(k)} - \tau B b_1 + \tau b_2 + \hat{\omega}B(\delta_1^{(k+1)} - \delta_1^{(k)}) + \tau B\delta_1^{(k)}, \\ y_1^{(k+1)} &= (1-\tau)y_1^{(k)} - \frac{\hat{\omega}(1-\omega)}{1-\hat{\omega}}A_{12}A_{12}^{*}(\delta_1^{(k+1)} - \delta_1^{(k)}) \\ &\quad -\hat{\omega}(\delta_1^{(k+1)} - \delta_1^{(k)}) - \tau A_{12}y_2^{(k)} + \tau b_1 - \tau\delta_1^{(k)}, \\ y_2^{(k+1)} &= \frac{1-\omega}{1-\hat{\omega}}A_{12}^{*}(\delta_1^{(k+1)} - \delta_1^{(k)}) + y_2^{(k)}. \end{aligned}$$

End

(4) Using (13) to obtain

$$w = \bar{A}^{+}b = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_4 \end{bmatrix}.$$

(5) $x = \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & I \end{bmatrix} w.$

Comparing the Algorithm 2.5 with the Algorithm 2 in the [20], although the step (5) is added to the Algorithm 2.5, we can see that the Algorithm 2.5 is simpler than the Algorithm 2 in the [20], because the for-loop can reduce several times A_{11}^{-1} multiplication computation. Hence, the computation time of the Algorithm 2.5 for solving the (1) must be less than the computation time of the Algorithm 2 in the [20], and numerical examples in Section 4 will demonstrate our arguments.

When $\omega = \hat{\omega}$, we can get the algorithm of the MSSOR method for solving the rank deficient linear least squares problem (1).

When $\hat{\omega} = 0$, we can get the algorithm of the MSOR method for solving the rank deficient linear least squares problem (1).

3. Optimal parameter of the MUSSOR method

In order to improve the convergence rate, we further study the optimal parameter of the MUSSOR method for solving (1).

From the proof process of Theorem 2.2 in [20], we can conclude that the following lemma.

Lemma 3.1. *Suppose λ is an eigenvalue of $T_{\omega,\hat{\omega}}$ and $\tau > 0$, $\omega, \hat{\omega} \neq 1$. If λ and ν satisfy*

$$\lambda^2 - [2(1-\tau) - \tau^2\nu]\lambda + (1-\tau)^2 = 0, \tag{14}$$

then $\nu \in \sigma(B^*B)$. Conversely, if $\nu \in \sigma(B^*B)$, and λ satisfies (14), then $\lambda \in \sigma(T_{\omega,\hat{\omega}})$ holds.

Based on the Lemma 3.1 and Theorem 2.1, we can prove the following results.

Theorem 3.2. *Let $0 < \tau < 1$. Then*

$$\rho(T_{\omega,\hat{\omega}}) = \begin{cases} \frac{1}{2}\tau^2\|B\|^2 + \frac{1}{2}\tau\|B\|\sqrt{\tau^2\|B\|^2 - 4(1-\tau)} + \tau - 1, & if \quad \Delta(\tau, \|B\|^2) \geq 0, \\ 1-\tau, & if \quad \Delta(\tau, \|B\|^2) < 0, \end{cases} \tag{15}$$

where $\Delta(\tau, \|B\|^2) = \tau^2\|B\|^2 - 4(1-\tau)$.

Proof. The two roots of (14) can be written as

$$\lambda_1(\tau,\nu) = \frac{1}{2}\left[\phi(\tau,\nu) + \tau\sqrt{\nu}\sqrt{\Delta(\tau,\nu)}\right], \ \lambda_2(\tau,\nu) = \frac{1}{2}\left[\phi(\tau,\nu) - \tau\sqrt{\nu}\sqrt{\Delta(\tau,\nu)}\right], \tag{16}$$

where

$$\phi(\tau,\nu) = 2(1-\tau) - \tau^2\nu, \ \Delta(\tau,\nu) = \tau^2\nu - 4(1-\tau). \tag{17}$$

Let

$$\lambda(\tau,\nu) = \max\{|\lambda_1(\tau,\nu)|, |\lambda_2(\tau,\nu)|\}.$$

Then we have the following two cases:

(i) If $\Delta(\tau, \|B\|^2) < 0$, then $\Delta(\tau,\nu) < 0$. Hence, $\lambda(\tau,\nu) = |\lambda_1(\tau,\nu)| = |\lambda_2(\tau,\nu)| = 1-\tau$;

(ii) If $\Delta(\tau, \|B\|^2) \geq 0$, then there exists at least one $\nu \in \sigma(B^*B)$ such that $\Delta(\tau,\nu) \geq 0$. From (17), we obtain that

$$\phi(\tau,\nu) = -\Delta(\tau,\nu) - 2(1-\tau) \leq -2(1-\tau) \leq 0.$$

So, from (16), it gets

$$\lambda(\tau,\nu) = |\lambda_2(\tau,\nu)| = -\lambda_2(\tau,\nu).$$

Therefore, we obtain

$$\begin{aligned} \rho(T_{\omega,\hat{\omega}}) &= \max_{\nu\in\sigma(B^*B)}\{\lambda(\tau,\nu)\} \\ &= \max_{\nu\in\sigma(B^*B)}\{-\lambda_2(\tau,\nu)\} \\ &= \frac{1}{2}\max_{\nu\in\sigma(B^*B)}\left\{\tau^2\nu - 2(1-\tau) + \tau\sqrt{\nu}\sqrt{\tau^2\nu - 4(1-\tau)}\right\} \\ &= \frac{1}{2}[\tau^2\|B\|^2 - 2(1-\tau) + \tau\|B\|\sqrt{\tau^2\|B\|^2 - 4(1-\tau)}]. \quad \square \end{aligned}$$

In the following, we study the optimal parameter and corresponding optimal convergence factor.

Theorem 3.3. *Let the parameters $(\omega, \hat{\omega})$ satisfy the conditions in Theorem 2.1. Then the optimal parameters $(\omega_{opt}, \hat{\omega}_{opt})$ of the MUSSOR method for solving the (1) satisfies*

$$\omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = \frac{-2 + 2\sqrt{1 + \|B\|^2}}{\|B\|^2}, \tag{18}$$

and the corresponding optimal convergence factor is

$$\upsilon(\bar{S}_{\omega_{opt},\hat{\omega}_{opt}}) = \frac{\sqrt{1 + \|B\|^2} - 1}{\sqrt{1 + \|B\|^2} + 1}.$$

Proof. If $\Delta(\tau, \|B\|^2) \geq 0$, it is easy to see that $\lambda_1(\tau, \nu)\lambda_2(\tau, \nu) = (1 - \tau)^2$ from (14). Then, we have

$$\lambda(\tau, \nu) = \max\{|\lambda_1(\tau, \nu)|, |\lambda_2(\tau, \nu)|\} \geq 1 - \tau.$$

Hence, from (9) and (15), it gets

$$\upsilon(\bar{S}_{\omega,\hat{\omega}}) = \max\{1 - \tau, \rho(T_{\omega,\hat{\omega}})\} = \begin{cases} \frac{1}{2}\tau^2\|B\|^2 + \frac{1}{2}\tau\|B\|\sqrt{\tau^2\|B\|^2 - 4(1 - \tau)} + \tau - 1, & if \quad \Delta(\tau, \|B\|^2) \geq 0, \\ 1 - \tau, & if \quad \Delta(\tau, \|B\|^2) < 0. \end{cases} \tag{19}$$

Let $\Delta(\tau, \|B\|^2) = 0$, that is $\tau^2\|B\|^2 - 4(1 - \tau) = 0$. Then its positive root is

$$\tau_* = \frac{-2 + 2\sqrt{1 + \|B\|^2}}{\|B\|^2},$$

which satisfies $0 < \tau_* < 2/(1 + \|B\|)$.

If $\|B\| > 1$, then we have that

$$\Delta(\tau, \|B\|^2) \geq 0 \ \text{ iff } \ \tau_* \leq \tau < \frac{2}{1 + \|B\|}$$

and

$$\Delta(\tau, \|B\|^2) < 0 \ \text{ iff } \ 0 < \tau < \tau_*.$$

Obviously, $\tau\|B\|$, $\tau^2\|B\|^2$ and $\tau - 1$ are increasing functions on the interval $\tau_* \leq \tau < 2/(1 + \|B\|)$, so that $\frac{1}{2}\tau^2\|B\|^2 + \frac{1}{2}\tau\|B\|\sqrt{\tau^2\|B\|^2 - 4(1 - \tau)} + \tau - 1$ is an increasing function. And $1 - \tau$ is decreasing function on the interval $0 < \tau < \tau_*$.

Therefore, the optimal parameters $(\omega_{opt}, \hat{\omega}_{opt})$ satisfy

$$\omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = \tau_* = \frac{-2 + 2\sqrt{1 + \|B\|^2}}{\|B\|^2}$$

and the corresponding optimal convergence factor is

$$\begin{aligned} \upsilon(\bar{S}_{\omega_{opt},\hat{\omega}_{opt}}) &= \min_{\omega,\hat{\omega}}\{\upsilon(\bar{S}_{\omega,\hat{\omega}})\} \\ &= \begin{cases} \frac{1}{2}\tau_*^2\|B\|^2 + \frac{1}{2}\tau_*\|B\|\sqrt{\tau_*^2\|B\|^2 - 4(1 - \tau_*)} + \tau_* - 1, & if\ \tau_* \leq \tau < \frac{2}{1+\|B\|}, \\ 1 - \tau_*, & if\ 0 < \tau < \tau_*, \end{cases} \\ &= 1 - \tau_* \\ &= \frac{\sqrt{1 + \|B\|^2} - 1}{\sqrt{1 + \|B\|^2} + 1}. \end{aligned}$$

Similarly, if $\|B\| \leq 1$, then we have

$$\Delta(\tau, \|B\|^2) \geq 0 \ \text{ iff } \ \tau_* \leq \tau < 1$$

and

$$\Delta(\tau,\|B\|^2) < 0 \ \text{ iff } \ 0 < \tau < \tau_*.$$

In like manner, since $\tau\|B\|$, $\tau^2\|B\|^2$ and $\tau - 1$ are increasing functions on the interval $\tau_* \leq \tau < 1$ and $1-\tau$ is decreasing function on the interval $0 < \tau < \tau_*$. Therefore, from (19), the optimal parameters $(\omega_{opt}, \hat{\omega}_{opt})$ satisfy

$$\omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = \tau_* = \frac{-2 + 2\sqrt{1+\|B\|^2}}{\|B\|^2}$$

and the corresponding optimal convergence factor is

$$\begin{aligned} \upsilon(\bar{S}_{\omega_{opt},\hat{\omega}_{opt}}) &= \min_{\omega,\hat{\omega}}\{\upsilon(\bar{S}_{\omega,\hat{\omega}})\} \\ &= \begin{cases} \frac{1}{2}\tau_*^2\|B\|^2 + \frac{1}{2}\tau_*\|B\|\sqrt{\tau_*^2\|B\|^2 - 4(1-\tau_*)} + \tau_* - 1, & if\ \tau_* \leq \tau < 1, \\ 1-\tau_*, & if\ 0 < \tau < \tau_*, \end{cases} \\ &= 1 - \tau_* \\ &= \frac{\sqrt{1+\|B\|^2} - 1}{\sqrt{1+\|B\|^2} + 1}. \end{aligned}$$

Based on the above discussion, it can be concluded that when $\tau_* = \frac{-2+2\sqrt{1+\|B\|^2}}{\|B\|^2}$, that is $1 - \tau_* = \frac{\sqrt{1+\|B\|^2}-1}{\sqrt{1+\|B\|^2}+1}$. $\upsilon(\bar{S}_{\omega_{opt},\hat{\omega}_{opt}})$ takes the minimum value $\frac{\sqrt{1+\|B\|^2}-1}{\sqrt{1+\|B\|^2}+1}$, which means the convergence speed is the fastest. □

The figures of parameters $(\omega,\hat{\omega})$ and the corresponding convergence factors $\upsilon(\bar{S}_{\omega,\hat{\omega}})$ can be described as the following Fig.1 and Fig.2, where the abscissa and ordinate denote τ and $\upsilon(\bar{S}_{\omega,\hat{\omega}})$, respectively.

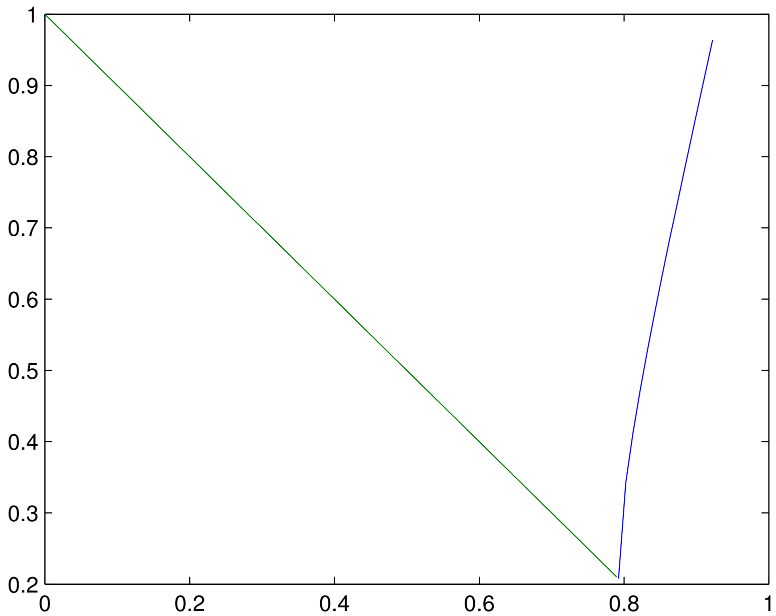

Figure 1: $\upsilon(\bar{S}_{\omega,\hat{\omega}})$ for $\|B\| > 1$

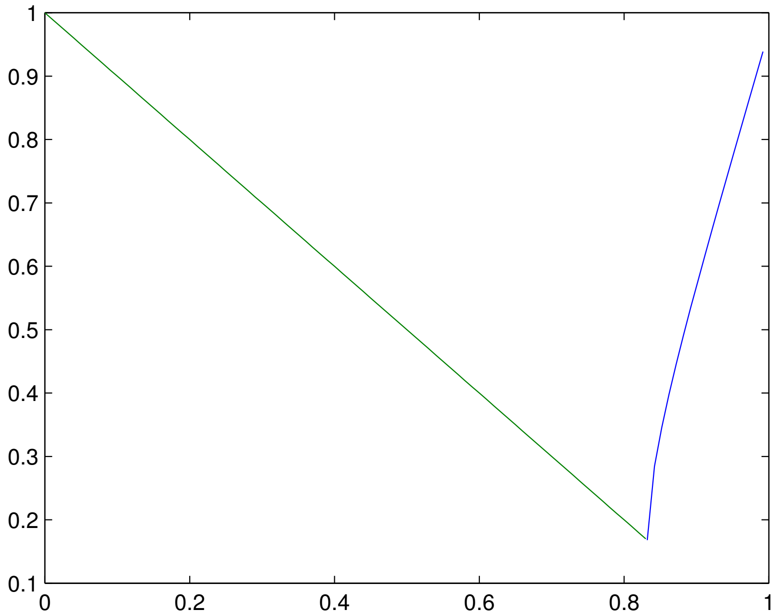

Figure 2: $\upsilon(\bar{S}_{\omega,\hat{\omega}})$ for $||B|| \leq 1$

When $\omega = \hat{\omega}$, from Theorem 3.3, we can derive the optimal parameters of the MSSOR method.

Corollary 3.4. *Let*

$$\omega \in \mathcal{D} = \begin{cases} (0,\tilde{\omega}_2)\bigcup(\tilde{\omega}_1, 2), & if \quad ||B|| > 1, \\ (0,2), \ \omega \neq 1, & if \quad ||B|| \leq 1, \end{cases}$$

where

$$\tilde{\omega}_1 = 1 + \sqrt{\frac{||B|| - 1}{||B|| + 1}}, \ \tilde{\omega}_2 = 1 - \sqrt{\frac{||B|| - 1}{||B|| + 1}}.$$

Then the MSSOR method has two optimal parameters ω_{opt1} and ω_{opt2}, that are given by

$$\omega_{opt1} = 1 - \frac{||B||}{1 + \sqrt{1 + ||B||^2}}, \ \omega_{opt2} = 1 + \frac{||B||}{1 + \sqrt{1 + ||B||^2}}. \tag{20}$$

And the corresponding optimal convergence factor is

$$\upsilon(\bar{S}_{\omega_{opt1},\omega_{opt1}}) = \upsilon(\bar{S}_{\omega_{opt2},\omega_{opt2}}) = \frac{\sqrt{1 + ||B||^2} - 1}{\sqrt{1 + ||B||^2} + 1}.$$

Proof. On the one hand, from Theorem 3.1, when $\hat{\omega} = \omega$, we have $\tau = 2\omega - \omega^2$, so that $0 < \tau < 1$ if and only if $0 < \omega < 2$, $\omega \neq 1$. Similarly, $0 < \tau < 2/(1 + ||B||)$ if and only if $0 < \omega < \tilde{\omega}_2$ or $\tilde{\omega}_1 < \omega < 2$.

On the other hand, by Theorem 3.3 and (18), we know that the optimal parameter ω_{opt} satisfies the equation

$$2\omega_{opt} - \omega_{opt}^2 = \frac{-2 + 2\sqrt{1 + ||B||^2}}{||B||^2}.$$

Solving this equation we obtain two optimal parameters ω_{opt1} and ω_{opt2}, which are defined by (20). Therefore, for $i = 1, 2$, it gets

$$v(\bar{S}_{\omega_{opti},\omega_{opti}}) = (1 - \omega_{opti})^2 = \frac{\sqrt{1 + \|B\|^2} - 1}{\sqrt{1 + \|B\|^2} + 1}. \quad \square$$

When $\hat{\omega} = 0$, from Theorem 3.3, we can derive the optimal parameters of the MSOR method. When Q=I, we can obtain the optimal parameter of the USSOR method [20] for solving (1). When Q=I and $\omega = \hat{\omega}$, we can obtain the optimal parameter of the SSOR method for solving (1), which is consistent with the Theorem 3.2 [3].

Remark 3.5. *From the Fig.1 and Fig.2, it is easy to see that for the curve $v(\bar{S}_{\omega,\hat{\omega}})$, the slope of the line on the left of the optimal point $\tau_* = (-2 + 2\sqrt{1 + \|B\|^2})/\|B\|^2$ is less than the slope of the curve on the right. In fact, by simple computation, we know that the on the left of the optimal point τ_*, the slope of the straight line $1 - \tau$ is -1, and on the right, the slope of curve $\frac{1}{2}\tau^2\|B\|^2 + \frac{1}{2}\tau\|B\|\sqrt{\tau^2\|B\|^2 - 4(1 - \tau)} + \tau - 1$ on the optimal point τ_* is ∞. That is, on the left of the theoretical optimal point τ_*, the convergence is superior to the right. Since in the practical calculation, the calculated final optimal value are often approximate optimal values. Therefore, in the practical example, the τ smaller than the $\tau_* = (-2 + 2\sqrt{1 + \|B\|^2})/\|B\|^2$ should be taken as much as possible.*

Remark 3.6. *From Theorem 3.3, we know that the optimal values lies in the line $\omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = (-2 + 2\sqrt{1 + \|B\|^2})/\|B\|^2$ in theory. And, from Corollary 3.4, for the MSSOR method, the optimal value is the two points $\omega_{opt1} = 1 - \|B\|/(1 + \sqrt{1 + \|B\|^2})$ and $\omega_{opt2} = 1 + \|B\|/(1 + \sqrt{1 + \|B\|^2})$. From the computation process of the optimal parameter, we know that there are two times of approximate calculation process possibly. The first time is that in the computation of matrix B in the formula $B = A_{21}A_{11}^{-1}$. In the general case, the calculation of inverse matrix A_{11}^{-1} is approximate. The second time is that in the computation of $\sqrt{1 + \|B\|^2}$. In the general case, the $\sqrt{1 + \|B\|^2}$ is irrational number, and its calculation is also approximate. Therefore, the calculated final optimal value in the practical examples are approximate optimal values. Obviously, comparing with the MSSOR method, the freedom of the optimal value of the MUSSOR method is bigger. And the bigger freedom may help us find the closer to the actual optimal value.*

Remark 3.7. *From Theorem 3.3 and Corollary 3.4, we see that the MUSSOR method is of the same optimal convergence rate as the MSSOR method. However, from the Remark 1 and 2, we know that the closer to the actual optimal value may be obtained when using the MUSSOR method, and corresponding calculated optimal convergence rate may be much faster than that of the MSSOR method, which will be shown by numerical experiments in next section.*

4. Numerical examples

In this section, three numerical examples are tested to demonstrate the effectiveness and feasibility of the MUSSOR method and show the advantages of the MUSSOR method over the USSOR method [20] from two aspects: the iteration times (abbreviated as "IT") and the elapsed computation time (abbreviated as "CPU", unit: second, abbreviated as "s"). For convenience, all tests are started from the initial zero vector, are terminated if the current iterations satisfy $\|x_{k+1} - x_k\| < 10^{-9}$. All numerical experiments are realized by Matlab 7.9 on Intel(R) Core (TM) i3 CPU.

Example 4.1. Consider linear system $Ax = b$, where A is the following 400×60 random matrix, $D = rand(400, 55), A(:, 1:55) = D, A(:, 56) = D(:, 16); A(:, 57) = D(:, 3) + 2 * D(:, 8), A(:, 58) = D(:, 6) + D(:, 9), A(:, 59) = D(:, 10), A(:, 60) = D(:, 11) + D(:, 12)$, and $b = [b1, b2]$, with $b1 = round(100 * rand(55, 1)), b2 = round(100 * rand(345, 1))$.

For Example 4.1, $m = \|B\|_2 = 469.4698$, the USSOR method and the MUSSOR method for solving (1) are tested. When using the USSOR method and the MUSSOR method, we use Theorem 2.2 in [20] and Theorem 2.1 to obtain parameter $\omega, \hat{\omega}$. The detail computational results are listed in Table 4-1, where $k1$

and $k2$ denotes the iteration times of the MUSSOR method and the USSOR method, and the CPU1 and CPU2 denotes the computation time of the MUSSOR method and the USSOR method, respectively. From the Table 4-1, we can see that the CPU1 is less than CPU2. That is to say that the MUSSOR is superior to the USSOR method under some conditions, and the corresponding improvement of CPU time is about 63%.

Table 4-1 The numerical results for Example 4.1

ω	$\hat{\omega}$	τ	$k1$	CPU1	$k2$	CPU2
0.0021	0.0021	0.0042	5985	5.003	5905	13.6800
1.9979	1.9979	0.0042	6123	5.1736	6675	15.6466
1.9976	1.9983	0.0041	6422	5.4540	6781	15.4365
0.0023	0.0018	0.0042	5985	5.1077	5905	14.1363
0.0030	0.0009	0.0039	6443	5.4141	6330	14.8008
0.0040	0.0002	0.0042	5983	5.1414	5892	13.5773
0.0033	0.0002	0.0035	6995	5.7533	7174	15.8530
0.0033	0.0009	0.0042	5983	4.8281	5930	13.3489

Example 4.2. Consider linear system $Ax = b$, where A is the following 600×60 random matrix, $D = rand(600, 55), A(:, 1 : 55) = D, A(:, 56) = D(:, 18); A(:, 57) = D(:, 5) + 5 * D(:, 8), A(:, 58) = D(:, 16) + D(:, 9), A(:, 59) = D(:, 20), A(:, 60) = D(:, 21) + 3 * D(:, 20)$, and $b = [b1, b2]$, where $b1 = round(100 * rand(55, 1)), b2 = round(100 * rand(545, 1))$.

For Example 4.2, the USSOR method and the MUSSOR method for solving (1) are tested. It is easy to get rank(A)=55, $m = \|B\|_2 = 184.4469$. When using the USSOR method and the MUSSOR method, we can use Theorem 2.1 in this paper and Theorem 2.2 in [20] to obtain the convergence parameter $\omega, \hat{\omega}$, respectively, and use the Theorem 3.3 to obtain the optimal parameter $\tau_{opt} = \omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = 0.0108$. The detail computational results are listed in Table 4-2, where $k1$ and $k2$ denotes the iteration times, and the CPU1 and CPU2 denotes the computation time of the MUSSOR method and the USSOR method, respectively. From the Table 4-2, we can see that the CPU1 is less than CPU2. That is to say that the MUSSOR method is superior to the USSOR method, and the corresponding improvement of CPU time is about 50%. When the optimal parameter $\tau_{opt} = 0.0108$ is taken, the MUSSOR method is the best.

Table 4-2 The numerical results for Example 4.2

ω	$\hat{\omega}$	τ	$k1$	CPU1	$k2$	CPU2
0.0054	0.0054	0.0108	2466	3.1141	2454	6.6939
0.0068	0.0040	0.0108	2466	3.015118	2376	6.4274
0.0045	0.0060	0.0105	2580	3.3933	2441	6.7401
0.0040	0.0060	0.0100	2709	3.4724	2568	7.1851
0.0075	0.0030	0.0105	2538	3.1763	2442	6.7254

Example 4.3. Consider linear system $Ax = b$, where A is the following 3000×520 random matrix, $D = rand(3000, 480), A(:, 1 : 480) = D, A(:, 481) = D(:, 16), A(:, 482) = 3 * D(:, 15), A(:, 483) = D(:, 26), A(:, 484) = D(:, 13), A(:, 485) = D(:, 11) + D(:, 32), A(:, 486) = D(:, 71) + D(:, 92), A(:, 487) = D(:, 145), A(:, 488) = D(:, 241) + D(:, 162), A(:, 489) = D(:, 151) + D(:, 182), A(:, 490) = D(:, 161) + D(:, 172), A(:, 491) = D(:, 206), A(:, 492) = D(:, 315) + 2 * D(:, 16), A(:, 493) = D(:, 360) + D(:, 190), A(:, 494) = D(:, 125), A(:, 495) = D(:, 300) + D(:, 322), A(:, 496) = D(:, 231) + D(:, 242), A(:, 497) = D(:, 145), A(:, 498) = D(:, 141) + D(:, 162), A(:, 499) = D(:, 251) + D(:, 282), A(:, 500) = D(:, 361) + D(:, 372), A(:, 501) = D(:, 16), A(:, 502) = D(:, 5) + 2 * D(:, 6), A(:, 503) = D(:, 26) + D(:, 9), A(:, 504) = D(:, 25), A(:, 505) = D(:, 1) + D(:, 22), A(:, 506) = D(:, 31) + D(:, 42), A(:, 507) = D(:, 45), A(:, 508) = D(:, 41) + D(:, 62), A(:, 509) = D(:, 51) + D(:, 82), A(:, 510) = D(:, 61) + D(:, 72), A(:, 511) = D(:, 106), A(:, 512) = D(:, 215) + 2 * D(:, 6), A(:, 513) = D(:, 260) + D(:, 90), A(:, 514) = D(:, 325), A(:, 515) = D(:, 100) + D(:, 222), A(:, 516) = D(:, 331) + D(:, 342), A(:, 517) = D(:, 445), A(:, 518) = D(:, 441) + D(:, 462), A(:, 519) = D(:, 451) + D(:, 382), A(:, 520) = D(:, 461) + D(:, 472)$, $b = (b1, b2)^{\top}$, where b1=round(500*rand(480,1)), b2=round(500*rand(2520,1)).

For Example 4.3, the USSOR method and MUSSOR method for solving (1) are tested. It is easy to compute $||B|| = 1.7204 \times 10^3$. When using the USSOR method and MUSSOR method, we use Theorem 3.3 to obtain optimal parameter $\tau_{opt} = \omega_{opt} + \hat{\omega}_{opt} - \omega_{opt}\hat{\omega}_{opt} = 0.0012$. The detail computational results are listed in Table 4-3, where $k1$ and $k2$ denote the iteration times, and the $CPU1$ and $CPU2$ denote the computation time of the MUSSOR and USSOR method, respectively. From the Table 4-3, we can see that the CPU1 is less than CPU2. That is to say that the MUSSOR method is superior to USSOR method, and the corresponding improvement of CPU time is about 46%.

Table 4-3 The numerical results for Example 4.3

ω	$\hat{\omega}$	τ	$k1$	CPU1	$k2$	CPU2
5.7108×10^{-4}	5.9108×10^{-4}	0.0012	24739	1977.1465	22806	3626.1143
5.6108×10^{-4}	6.0108×10^{-4}	0.0012	24506	1965.3093	22424	3572.4748
5.5108×10^{-4}	6.1108×10^{-4}	0.0012	24100	1932.2157	22448	3574.0562

5. Conclusions

The USSOR method has been presented for solving the rank deficient linear least squares problem by J. Song and Y. Song (CALCOLO, 54(2017) 95-115). However, the convergence rate is relatively slow. In order to improve the convergence rate, the MUSSOR method is proposed, and the convergence theorems are proved in this paper. Furthermore, the optimal parameter and optimal convergence factor are given. Numerical examples demonstrate that the MUSSOR method is far superior to the USSOR method, and the corresponding improvement of CPU time is up to 63% under some conditions. With reference to [4, 15], we can construct MAOR method for solving the linear system (1) and give out the convergence theorems and optimal convergence factor.

6. Acknowledgements

This work was supported by Wuxi University Research Start-up Fund for Introduced Talents. The authors would like to thank the referees for their many valuable suggestions and comments which led us to improve this paper.

References

[1] R. H. AL-Obaidi, M. T. Darvishi, *A Comparative Study on Qualification Criteria of Nonlinear Solvers with Introducing Some New Ones*, J. Math-UK. **4327913** (2022), 2314–4629.

[2] A. Berman, R. J. Plemmons, *Nonnegative Matrices in the Mathematics Sciences*, Society for Industrial Mathematics, Philadelphia,1994.

[3] X. Chen, Y. Chen, *A necessary and sufficient condition for semiconvergent and optimal parameter of the SSOR method for solving the rank deficient linear least squares problem*, Appl. Math. Comput. **2** (2006), 1108–1126.

[4] Y. Chen, *Semiconvergence of AOR iterative methods for solving the rank deficient least squares problems*, J. Nanjing Normal Univ. **4** (2005),1–7.

[5] Y. T. Chen, *Iterative methods for linear least squares problems*, University of Waterloo, Canada, 1975.

[6] R. E. Cline, *Inverses of rank invariant powers of a matrix*, SIAM J. Numer Anal. **1** (1968), 182–197.

[7] M. T. Darvishi, P. Hessari, *Symmetric SOR method for augmented systems*, Appl. Math. Comput. **1** (2006), 409–415.

[8] M. T. Darvishi, R. Khosro-Aghdam, *Determination of the optimal value of relaxation parameter in symmetric SOR method for rectangular coefficient matrices*, Appl. Math. Comput. **2** (2006), 1018–1025.

[9] M. T. Darvishi, R. Khosro-Aghdam, *Symmetric successive overrelaxation methods for rank deficient linear systems*, Appl. Math. Comput. **1** (2006), 404–420.

[10] M. T. Darvishi, F. Khani, S. Hamedi-Nezhad, B. Zheng, *Symmetric block-SOR methods for rank-deficient least squares problems*, J. Comput. Appl. Math. **1** (2008), 14–27.

[11] M. T. Darvishi, P. Hessari, *A modified symmetric successive overrelaxation method for augmented systems*, Comput. Math. Appl. **10** (2011), 3128–3135.

[12] X. Dong, X. H. Shao, H. L. Shen, *A new SOR-like method for solving absolute value equations*, Appl. Numer. Math. **156** (2020), 410–421.

[13] G. H. Golub, C. F. Van, *Matrix computations*, Johns Hopkins University Press, Baltimore City, 2014.

[14] P. Guo, S. L. Wu, C. X. Li, *On the SOR-like method for solving absolute value equations*, Appl. Math. Lett. **97** (2019), 107–113.

[15] Y. Huang, Y. Song, *AOR iterative methods for rank deficient least squares problems*, J. Appl. Math. Comput. **1** (2008), 105–124.

[16] T. L. Markham, M. Neumann, R. J. Plemmons, *Convergence of a direct-iterative method for large-scale least squares problems*, Linear Algebra Appl. **69** (1985), 155–167.
[17] V. A. Miller, M. Neumann, *Successive overrelaxation methods for solving the rank deficient linear least squares problem*, Linear Algebra Appl. **88-89** (1987), 533–557.
[18] W. Niethammer, J. D. Pihis, R. S. Varga, *Convergence of block iterative methods applied to sparse least-squares problems*, Linear Algebra Appl. **58** (1984), 327–341.
[19] R. J. Plemmons, *Adjustment by least squares in Geodesy using block iterative methods for sparse matrices*, in: Proceedings of the Army Conference on Numerical Analysis and Computers, White Sands, NM, USA, 1979.
[20] J. Song, Y. Song, *USSOR methods for solving the rank deficient linear least squares problem*, CALCOLO. **54** (2017), 95–115.
[21] H. Tian, *Accelerate overrelaxation methods for rank deficient linear systems*, Appl. Math. Comput. **2-3** (2003), 485–499.
[22] L. Wang, *On the convergence of the USSOR iterative methods for solving least squares problems*, J. Nanjing Normal Univ. **23** (2000), 8–13.
[23] P. B. Xua, N. M. Zhang, *A Note on the optimal parameters of USSOR method for solving linear least squares problems*, FILOMAT. **17** (2019), 5441-5450.
[24] B. Zheng, K. Wang, *Symmetric successive overrelaxation methods for solving the rank deficient linear least squares problem*, Appl. Math. Comput. **2** (2005), 1305-1323.