AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 5(3) (2024) 195-215
https://doi.org/10.22060/AJMC.2024.22444.1159

Original Article

An iterative scheme for a class of generalized Sylvester matrix equations

Mehdi Dehghan*[a], Gholamreza Karamali[b], Akbar Shirilord[a]

[a] *Department of Mathematics and Computer Science, Amirkabir University of Technology (Tehran Polytechnic), Iran*
[b] *Faculty of Basic Sciences, Shahid Sattari Aeronautical University of Sciences and Technology, South Mehrabad, Tehran, Iran*

ABSTRACT: In this study based on the accelerated over relaxation (AOR) method we make an iterative scheme for solving generalized Lyapunov matrix equation

$$AX + XB + \sum_{j=1}^{m} N_j X M_j = C,$$

over complex or real matrices. Then we analyze the convergence of the new iterative method in detail. There have been discussions for the calculation of optimal parameters. Finally a numerical example is given to demonstrate the capability of the new method.

Review History:
Received:04 June 2023
Revised:01 February 2024
Accepted:13 February 2024
Available Online:01 July 2024

Keywords:
Optimal parameters
Complex matrices
Generalized Sylvester matrix equation
Accelerated over relaxation (AOR) method
Control theory

MSC (2020):
15A24; 39B42; 65F10

1. Introduction

The matrix equation

$$AX + XB + \sum_{j=1}^{m} N_j X M_j = C, \tag{1}$$

is a generalized Sylvester equation that involves known matrices A, B, N_j, M_j $(j = 1, \ldots, m), C \in \mathbb{C}^{n\times n}$ and the unknown matrix $X \in \mathbb{C}^{n\times n}$. This equation is commonly encountered in computational science and engineering applications, such as in evaluating implicit numerical schemes for partial differential equations and decoupling techniques for ordinary differential equations, among others. Previous works, such as [1, 2, 9, 10, 13, 30, 35, 36, 37], have addressed this problem. Specifically, in [14], an iterative algorithm was proposed to solve matrix equation (1).

*Corresponding author.
E-mail addresses: mdehghan@aut.ac.ir, mdehghan.aut@gmail.com, rezakaramali918@gmail.com, akbar.shirilord@aut.ac.ir

Previous researchers have investigated various methods for solving generalized Sylvester matrix equations. For instance, Dehghan and Hajarian [16] focused on the reflexive solutions of the coupled Sylvester matrix equations and proposed an iterative algorithm. Mukaidani et al. [32] developed a numerical algorithm for solving cross-coupled algebraic Riccati equations. Additionally, Zhou et al. [39, 40] investigated generalized Sylvester matrix equation (1) through the use of explicit solutions.

The focus of this paper is to present a novel iterative method for solving the generalized Sylvester matrix equation (1). Specifically, we introduce a new approach that can lead to faster convergence rates by carefully selecting appropriate parameters in the algorithm.

The linear Sylvester matrix equation (1) can be transformed into an equivalent system of linear equations as follows:

$$(I \otimes A + B^T \otimes I + \sum_{j=1}^{m} M_j^T \otimes N_j)\mathbf{vec}(X) = \mathbf{vec}(C),$$

where

$$\mathbf{vec}: \ X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \mapsto \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix},$$

where $\mathbf{x}_k$ is the $k-th$ column of X.

The uniqueness of the solution of linear Sylvester matrix equation (1) can be determined by verifying the following condition:

$$det(I \otimes A + B^T \otimes I + \sum_{j=1}^{m} M_j^T \otimes N_j) \neq 0,$$

where $det(A)$ denotes the determinant of the matrix A. This condition ensures that the system of linear equations associated with (1) has a unique solution.

Bouhamidi et al. proposed an iterative method to solve generalized Sylvester matrix equation (1) in [14]. This is just one of the many techniques that have been used to develop iterative methods for solving matrix equations. Some of the other methods that have been utilized include those proposed by Smith et al. [34], Soleymani et al. [5, 19, 27], as well as those by Ding et al. [22, 23, 24], for example. Additionally, iterative methods have also been designed for solving other types of matrix equations such as algebraic Riccati equations [21], coupled Sylvester matrix equations [18], and more [4, 8, 11, 15, 17, 25, 28, 29, 33].

1.1. The structure of the paper

The outline of this paper is as follows:

1. A new iterative method is introduced for Eq. (1) in Section 2 and also the convergence analysis of this method in details is presented. There have been discussions for the calculation of optimal parameters.
2. Section 3 includes a test problem to examine the new scheme and a comparison is made with the existing results.
3. At the end, we give some conclusions in Section 4.

1.2. Primitive definitions and notations

To ensure a clear understanding of the symbols and notation used in this paper, we will provide a brief explanation of them. The real vector sets with dimension n will be denoted by $\mathbb{R}^n$ ($\mathbb{C}^n$ for the complex vector), and the real matrices with m rows and n columns will be denoted by $\mathbb{R}^{m\times n}$ ($\mathbb{C}^{m\times n}$ for the complex matrix). Additionally, we will use the Hermitian of A denoted as A^H, the Frobenius norm for matrix A denoted as $\|A\|_F = \sqrt{\text{tr}(AA^H)}$, and the spectral radius of A denoted as $\rho(A)$. The Kronecker product (or tensor product) will be denoted by $\otimes$, the zero matrix by O, the identity matrix of size $n \times n$ by I_n, and A^T will refer to the transpose of A.

2. Main results

Drawing inspiration from the AOR technique proposed by Hadjimos [26], we have developed a novel approach for solving Eq. (1). To accomplish this, we introduce a splitting procedure for the matrices $A, C, N_1, \ldots, N_m$, and $M_1, \ldots, M_m$ in Eq. (1):

$$A = W_1 + iT_1, \quad B = W_2 + iT_2, \quad C = C_1 + iC_2, \quad N_j = U_j + iV_j, \quad M_j = F_j + iG_j, \ \ j = 1, \ldots, m,$$

where W_1, W_2, T_1, T_2, C_1, C_2 and U_j, V_j, F_j, G_j $(j = 1, \ldots, m)$ are real matrices in $\mathbb{R}^{n\times n}$.

Furthermore, assuming that the solution of Eq. (1) can be expressed in the form of $X = Z + iY$, where Z and $Y \in \mathbb{R}^{n\times n}$, we can substitute this assumption into the generalized Sylvester matrix equation (1), yielding the following result:

$$\begin{cases} W_1Z - T_1Y + ZW_2 - YT_2 + \sum_{j=1}^{m} [U_jZF_j - U_jYG_j - V_jZG_j - V_jYF_j] = C_1, \\ W_1Y + T_1Z + ZT_2 + YW_2 + \sum_{j=1}^{m} [U_jZG_j + U_jYF_j + V_jZF_j - V_jYG_j] = C_2. \end{cases}$$

Drawing upon the AOR technique proposed by Hadjimos [26], we derive the new method as follows:

$$\begin{cases} WZ + ZW^T = \ \omega\left(TY + YT^T\right) + (1-\omega)\left(WZ + ZW^T\right) \\ \qquad\qquad + \omega\sum_{j=1}^{m} [U_jYG_j + V_jZG_j + V_jYF_j - U_jZF_j] + \omega C_1, \\ WY + YW^T = -r\left(TZ + ZT^T\right) + (1-r)\left(WY + YW^T\right) \\ \qquad\qquad + r\sum_{j=1}^{m} [V_jYG_j - U_jZG_j - U_jYF_j - V_jZF_j] + rC_2. \end{cases} \tag{2}$$

We now consider the following iterative method based on (2), with the introduction of auxiliary real parameters $\omega \neq 0$ and $r \neq 0$:

$$\begin{cases} W_1Z^{(k+1)} + Z^{(k+1)}W_2 = \ \omega\left(T_1Y^{(k)} + Y^{(k)}T_2\right) + (1-\omega)\left(W_1Z^{(k)} + Z^{(k)}W_2\right) \\ \qquad\qquad + \omega\sum_{j=1}^{m}\left[U_jY^{(k)}G_j + V_jZ^{(k)}G_j + V_jY^{(k)}F_j - U_jZ^{(k)}F_j\right] + \omega C_1, \\ W_1Y^{(k+1)} + Y^{(k+1)}W_2 = -r\left(T_1Z^{(k+1)} + Z^{(k+1)}T_2\right) + (1-r)\left(W_1Y^{(k)} + Y^{(k)}W_2\right) \\ \qquad\qquad + r\sum_{j=1}^{m}\left[V_jY^{(k)}G_j - U_jZ^{(k+1)}G_j - U_jY^{(k)}F_j - V_jZ^{(k+1)}F_j\right] + rC_2. \end{cases} \tag{3}$$

By appropriately selecting values for ω and r, we can ensure that the resulting AOR-like iterative method exhibits a faster convergence rate and higher computational efficiency.

We initialize the iterative method with an initial approximation of $X^{(0)} = Z^{(0)} + iY^{(0)}$.

Remark 2.1. *If we set $\omega = r$ in the AOR-Like method given by (3), the resulting method reduces to the SOR-Like method described in [20].*

Remark 2.2. *If the generalized Sylvester matrix equation (1) is defined on real matrices, such that*

$$Y = O, \qquad T_1 = T_2 = O, \qquad C_2 = O, \qquad V_j = O, \qquad G_j = O, \ j = 1, \ldots, m,$$

then method given by (3) can be expressed as:

$$W_1Z^{(k+1)} + Z^{(k+1)}W_2 = (1-\omega)\left(W_1Z^{(k)} + Z^{(k)}W_2\right) - \omega\sum_{j=1}^{m}\left[U_jZ^{(k)}F_j\right] + \omega C_1.$$

If we further assume that $W_2 = W_1^T$ and $F_j = U_j^T (j = 1, \ldots, m)$, then AOR-Like method (3) reduces to the method proposed in [20] for solving the real generalized Lyapunov matrix equation.

2.1. Convergence analysis of AOR-Like method

In this section, we will discuss the convergence properties of the new iteration method. To begin with, let us set

$$\tilde{W} = W_2^T \otimes I_n + I_n \otimes W_1, \qquad \tilde{T} = T_2^T \otimes I_n + I_n \otimes T_1,$$
$$P = \sum_{j=1}^{m} \left[F_j^T \otimes V_j + G_j^T \otimes U_j\right], \qquad Q = \sum_{j=1}^{m} \left[G_j^T \otimes V_j - F_j^T \otimes U_j\right],$$

and

$$\Omega_1(\alpha) = (1-\alpha)I_{n^2} + \alpha\hat{Q}, \qquad \Omega_2(\alpha) = \alpha\hat{T} + \alpha\hat{P}, \qquad (\alpha \in \mathbb{R}), \tag{4}$$

where $\hat{Q} = \tilde{W}^{-1}Q$, $\hat{P} = \tilde{W}^{-1}P$ and $\hat{T} = \tilde{W}^{-1}\tilde{T}$.

For the remainder of this paper, we will require the following lemmas.

Lemma 2.3. *[20] Suppose that the matrices P, Q, $\tilde{W}$ and $\tilde{T}$ satisfy*

$$Q\hat{T} = \tilde{T}\hat{Q}, \qquad \text{and} \qquad Q\hat{P} = P\hat{Q}, \tag{5}$$

then the matrices $\Omega_1(\omega)$ and $\Omega_2(r)$ are commute, i.e., $\Omega_1(\omega)\Omega_2(r) = \Omega_2(r)\Omega_1(\omega)$.

Proof. Assumptions (5) yield

$$r(1-\omega)\hat{T} + r(1-\omega)\hat{P} + \omega r\hat{Q}\hat{T} + \omega r\hat{Q}\hat{P} = r(1-\omega)\hat{T} + \omega r\hat{T}\hat{Q} + r(1-\omega)\hat{P} + \omega r\hat{P}\hat{Q},$$

or

$$\left[(1-\omega)I + \omega\hat{Q}\right]\left[r\hat{T} + r\hat{P}\right] = \left[r\hat{T} + r\hat{P}\right]\left[(1-\omega)I + \omega\hat{Q}\right],$$

which becomes $\Omega_1(\omega)\Omega_2(r) = \Omega_2(r)\Omega_1(\omega)$.

□

We also have the following lemma.

Lemma 2.4. *[20] Let the matrices U_j, V_j, F_j, G_j, $(j = 1, \ldots, m)$ be symmetric and W be symmetric positive definite. Then the eigenvalues of matrix $\hat{Q}$ are real.*

Proof. Since the matrices U_j, V_j, $(j = 1, \ldots, m)$ are symmetric, thus Q is symmetric. On the other hand W is symmetric positive definite matrix, that yields the matrix $\tilde{W} = W \otimes I + I \otimes W$ is symmetric positive definite. Hence $\tilde{W}$ is invertible. Therefore we can write

$$\hat{Q} = \tilde{W}^{-\frac{1}{2}}\left(\tilde{W}^{-\frac{1}{2}}Q\tilde{W}^{-\frac{1}{2}}\right)\tilde{W}^{\frac{1}{2}}.$$

This relation concludes that $\hat{Q}$ is similar to $\tilde{W}^{-\frac{1}{2}}Q\tilde{W}^{-\frac{1}{2}}$. Since Q is symmetric then $\tilde{W}^{-\frac{1}{2}}Q\tilde{W}^{-\frac{1}{2}}$ is symmetric matrix. This completes the proof. □

We will also need the following lemma in the following.

Lemma 2.5. *[3] Both roots of the real quadratic equation $\lambda^2 - p\lambda + s = 0$ are less than one in modulus if and only if $|s| < 1$ and $|p| < 1 + s$.*

We are now ready to discuss the convergence analysis of the AOR-Like method. By taking the operator **vec** from both sides of equation (3), we obtain:

$$\begin{cases} (W_2^T \otimes I_n + I_n \otimes W_1)\mathbf{z}^{(k+1)} = & \omega(T_2^T \otimes I_n + I_n \otimes T_1)\mathbf{y}^{(k)} + (1-\omega)(W_2^T \otimes I_n + I_n \otimes W_1)\mathbf{z}^{(k)} \\ & + \omega\sum_{j=1}^{m}\left[F_j^T \otimes V_j + G_j^T \otimes U_j\right]\mathbf{y}^{(k)} \\ & + \omega\sum_{j=1}^{m}\left[G_j^T \otimes V_j - F_j^T \otimes U_j\right]\mathbf{z}^{(k)} + \omega\mathbf{vec}(C_1), \\ (W_2^T \otimes I_n + I_n \otimes W_1)\mathbf{y}^{(k+1)} = & -r(T_2^T \otimes I_n + I_n \otimes T_1)\mathbf{z}^{(k+1)} + (1-r)(W_2^T \otimes I_n + I_n \otimes W_1)\mathbf{y}^{(k)} \\ & - r\sum_{j=1}^{m}\left[F_j^T \otimes V_j + G_j^T \otimes U_j\right]\mathbf{z}^{(k+1)} \\ & + r\sum_{j=1}^{m}\left[G_j^T \otimes V_j - F_j^T \otimes U_j\right]\mathbf{y}^{(k)} + r\mathbf{vec}(C_2), \end{cases} \tag{6}$$

where $\mathbf{y}^{(k)} = \mathbf{vec}(Y^{(k)})$ and $\mathbf{z}^{(k)} = \mathbf{vec}(Z^{(k)})$. Eq. (6) takes the simple form

$$\begin{cases} \tilde{W}\mathbf{z}^{(k+1)} = ((1-\omega)\tilde{W} + \omega Q)\mathbf{z}^{(k)} + (\omega\tilde{T} + \omega P)\mathbf{y}^{(k)} + \omega\mathbf{vec}(C_1), \\ \tilde{W}\mathbf{y}^{(k+1)} = -(r\tilde{T} + rP)\mathbf{z}^{(k+1)} + ((1-r)\tilde{W} + rQ)\mathbf{y}^{(k)} + r\mathbf{vec}(C_2). \end{cases} \tag{7}$$

Now (7) becomes

$$\begin{cases} \mathbf{z}^{(k+1)} = \Omega_1(\omega)\mathbf{z}^{(k)} + \Omega_2(\omega)\mathbf{y}^{(k)} + \omega\mathbf{r}, \\ \mathbf{y}^{(k+1)} = -\Omega_2(r)\mathbf{z}^{(k+1)} + \Omega_1(r)\mathbf{y}^{(k)} + r\mathbf{s}, \end{cases} \tag{8}$$

where $\mathbf{r} = \tilde{W}^{-1}\mathbf{vec}(C_1)$, $\mathbf{s} = \tilde{W}^{-1}\mathbf{vec}(C_2)$, and Ω_1, Ω_2 are defined in (4).

Substituting the first equation of (8) into the second equation yields:

$$\begin{aligned} \mathbf{y}^{(k+1)} &= -\Omega_2(r)\left[\Omega_1(\omega)\mathbf{z}^{(k)} + \Omega_2(\omega)\mathbf{y}^{(k)} + \omega\mathbf{r}\right] + \Omega_1(r)\mathbf{y}^{(k)} + r\mathbf{s} \\ &= -\Omega_2(r)\Omega_1(\omega)\mathbf{z}^{(k)} + (\Omega_1(r) - \Omega_2(r)\Omega_2(\omega))\mathbf{y}^{(k)} + r\mathbf{s} - \omega\Omega_2(r)\mathbf{r}. \end{aligned} \tag{9}$$

Using the relationship given in equation (9), it is possible to express the coupled vector form given in equation (8) in the form of a matrix-vector equation:

$$\begin{bmatrix} \mathbf{z}^{(k+1)} \\ \mathbf{y}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \Omega_1(\omega) & \Omega_2(\omega) \\ -\Omega_1(\omega)\Omega_2(r) & \Omega_1(r) - \Omega_2(\omega)\Omega_2(r) \end{bmatrix} \begin{bmatrix} \mathbf{z}^{(k)} \\ \mathbf{y}^{(k)} \end{bmatrix} + \begin{bmatrix} \omega I_{n^2} & O \\ -\omega\Omega_2(r) & rI_{n^2} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}. \tag{10}$$

The iterative form of relation (10) is commonly used and can be expressed as follows:

$$\mathbf{u}^{(k+1)} = \Phi(\omega, r)\mathbf{u}^{(k)} + \Upsilon(\omega, r)\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}, \tag{11}$$

where

$$\Phi(\omega, r) = \begin{bmatrix} \Omega_1(\omega) & \Omega_2(\omega) \\ -\Omega_2(r)\Omega_1(\omega) & \Omega_1(r) - \Omega_2(r)\Omega_2(\omega) \end{bmatrix},$$

is called iteration matrix of AOR-Like method and

$$\Upsilon(\omega, r) = \begin{bmatrix} \omega I_{n^2} & O \\ -\omega\Omega_2(r) & rI_{n^2} \end{bmatrix}.$$

Also the vector $\mathbf{u}^{(k)}$ is defined by $\mathbf{u}^{(k)} = \left[\mathbf{z}^{(k)^T}, \mathbf{y}^{(k)^T}\right]^T$.

Iteration matrix $\Phi(\omega, r)$ can be decomposed as (See [6])

$$\begin{bmatrix} \Omega_1(\omega) & \Omega_2(\omega) \\ -\Omega_1(\omega)\Omega_2(r) & \Omega_1(r) - \Omega_2(\omega)\Omega_2(r) \end{bmatrix} = \begin{bmatrix} I_{n^2} & O \\ -\Omega_2(r) & I_{n^2} \end{bmatrix} \begin{bmatrix} \Omega_1(\omega) & \Omega_2(\omega) \\ O & \Omega_1(r) \end{bmatrix},$$

where I_{n^2} is identity matrix of size $n^2 \times n^2$.

We will now examine the convergence properties of the new iterative method.

The next theorem presents sufficient conditions for the convergence of the new method. According to relation (11), it is clear that AOR-Like method is convergent if and only if $\rho(\Phi(\omega, r)) < 1$.

Theorem 2.6. *Assuming that the conditions of Lemmas 2.3 and 2.4 are satisfied, and there exists a real constant σ such that $\hat{T} + \hat{P} = \sigma I_{n^2}$, and let $\eta < 1$ be any eigenvalue of matrix $\hat{Q}$.*
(a): *Suppose*

$$0 < \sigma < 1 - \eta, \quad 0 < \omega \leq \frac{2}{1 + \sigma - \eta}, \quad 0 < r < \frac{-4 + 2\omega - 2\eta\omega}{-2 + 2\eta + \omega + \sigma^2\omega - 2\eta\omega + \eta^2\omega}, \tag{12}$$

then $\rho(\Phi(\omega, r)) < 1$.
(b): *Suppose*

$$\eta - 1 < \sigma < 0, \quad 0 < \omega \leq \frac{2}{1 - \sigma - \eta}, \quad 0 < r < \frac{-4 + 2\omega - 2\eta\omega}{-2 + 2\eta + \omega + \sigma^2\omega - 2\eta\omega + \eta^2\omega}, \tag{13}$$

then $\rho(\Phi(\omega, r)) < 1$.

Proof. Suppose $(\lambda, \mathbf{v})$ is an eigenpair of matrix $\Phi(\omega, r)$, where $\mathbf{v} = [v^T,\ w^T]^T$, with $v,\ w \in \mathbb{C}^{n^2}$. The definition of eigenvalue and eigenvector can be used to derive the following expression $\Phi(\omega, r)\begin{bmatrix} v \\ w \end{bmatrix} = \lambda \begin{bmatrix} v \\ w \end{bmatrix}$ or (see [6]):

$$\begin{bmatrix} \Omega_1(\omega) & \Omega_2(\omega) \\ O & \Omega_1(r) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \lambda \begin{bmatrix} I_{n^2} & O \\ -\Omega_2(r) & I_{n^2} \end{bmatrix}^{-1} \begin{bmatrix} v \\ w \end{bmatrix} = \lambda \begin{bmatrix} I_{n^2} & O \\ \Omega_2(r) & I_{n^2} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix},$$

that gives

$$\begin{cases} \Omega_2(\omega) w = (\lambda I_{n^2} - \Omega_1(\omega))\, v, \\ \\ -(\lambda I_{n^2} - \Omega_1(r))\, w = \lambda \Omega_2(r) v. \end{cases} \tag{14}$$

Lemma 2.3 concludes $\Omega_1(\omega)\Omega_2(r) = \Omega_2(r)\Omega_1(\omega)$ and from (14) we have

$$-(\lambda I_{n^2} - \Omega_1(\omega))\,(\lambda I_{n^2} - \Omega_1(r))\, w = \lambda \Omega_2(r)\Omega_2(\omega) w.$$

Based on these results, it is straightforward to observe that

$$-\left(\lambda I_{n^2} - [(1-\omega) I_{n^2} + \omega \hat{Q}]\right)\left(\lambda I_{n^2} - [(1-r) I_{n^2} + r\hat{Q}]\right) w = \lambda [r\hat{T} + r\hat{P}][\omega \hat{T} + \omega \hat{P}] w = \omega r \lambda [\hat{T} + \hat{P}]^2 w,$$

and

$$\begin{aligned} \left(\lambda I_{n^2} - [(1-\omega) I_{n^2} + \omega \hat{Q}]\right)\left(\lambda I_{n^2} - [(1-r) I_{n^2} + r\hat{Q}]\right) = \quad & (\lambda^2 + (\omega + r - 2)\lambda - \omega - r + \omega r + 1) I_{n^2} \\ & + (\omega + r - (\omega + r)\lambda - 2\omega r)\hat{Q} + \omega r \hat{Q}^2. \end{aligned}$$

Obviously, from the above equation we can write

$$[(\lambda^2 + (\omega + r - 2)\lambda - \omega - r + \omega r + 1) I_{n^2} + (\omega + r - (\omega + r)\lambda - 2\omega r)\hat{Q} + \omega r \hat{Q}^2] w = \omega r \lambda [\hat{T} + \hat{P}]^2 w.$$

Since η is any eigenvalue of matrix $\hat{Q}$, then

$$\lambda^2 + (\omega + r - 2)\lambda - \omega - r + \omega r + 1 + (\omega + r - (\omega + r)\lambda - 2\omega r)\eta + \omega r \eta^2 = \omega r \sigma^2 \lambda,$$

or

$$\lambda^2 + \xi_1(\omega, r)\lambda + \xi_2(\omega, r) = 0, \tag{15}$$

where

$$\xi_1(\omega, r) = -\omega\, r\, \sigma^2 - \omega\, \eta - r\, \eta + \omega + r - 2,$$

and

$$\xi_2(\omega, r) = (r\, \eta - r + 1)\,(\omega\, \eta - \omega + 1)\,.$$

To establish convergence of the new method, it is necessary to demonstrate that the roots of polynomial (15) are contained within the unit circle of the complex plane. Lemma 2.5 indicates that in order to accomplish this, it is essential to satisfy two conditions: $|\xi_2(\omega, r)| < 1$ and $|\xi_1(\omega, r)| < 1 + \xi_2(\omega, r)$.

$$\begin{cases} |\,(r\, \eta - r + 1)\,(\omega\, \eta - \omega + 1)\,| < 1, \\ \\ |-\omega\, r\, \sigma^2 - \omega\, \eta - r\, \eta + \omega + r - 2| < 1 + (r\, \eta - r + 1)\,(\omega\, \eta - \omega + 1)\,. \end{cases}$$

Now it is easy to check that if the parameters ω and r satisfy in relations (12) or (13), then $\rho\,(\Phi(\omega, r)) < 1$. □

It should be noted that determining the optimal parameters for the proposed method is a highly challenging task. In the following we will minimize the spectral radius of iteration matrix $\Phi(\omega, r)$.

Theorem 2.7. *Let the conditions ((a) or (b)) of Theorem 2.6 be satisfied and $\eta \neq 1$. Then*

$$r_{opt}(\omega) = \frac{\omega(-\eta\omega\sigma^2 + \omega\sigma^2 + \eta^2 - 2\sigma^2 - 2\eta + 1 + 2\sigma\sqrt{(\eta - 1 + \sigma)(\eta - 1 - \sigma)(-\eta\omega + \omega - 1)})}{(\omega\sigma^2 + \eta - 1)^2}, \tag{16}$$

$$r_{opt}(\omega)=\frac{\omega(-\eta\omega\sigma^2+\omega\sigma^2+\eta^2-2\sigma^2-2\eta+1-2\sigma\sqrt{(\eta-1+\sigma)(\eta-1-\sigma)(-\eta\omega+\omega-1)})}{(\omega\sigma^2+\eta-1)^2}, \tag{17}$$

where ω is a solution of the following minimization problem

$$\min_{\omega}\left|\omega\, r_{opt}(\omega)\,\sigma^2+\omega\,\eta+r_{opt}(\omega)\,\eta-\omega-r_{opt}(\omega)+2\right|, \tag{18}$$

or

$$r_{opt}(\omega_{opt})=\frac{\omega_{opt}\,\eta-\omega_{opt}+2}{-\omega_{opt}\,\sigma^2-\eta+1}, \tag{19}$$

where ω is a solution of the following minimization problem

$$\min_{\omega}\left|\sqrt{(r_{opt}(\omega)\,\eta-r_{opt}(\omega)+1)\,(\omega\,\eta-\omega+1)}\right|. \tag{20}$$

Proof. To minimize the modulus of the roots in Eq. (15), it is necessary to choose appropriate values for the two parameters ω and r. Specifically, the maximum value between $|\lambda_1(\omega,r)|$ and $|\lambda_2(\omega,r)|$, where $\lambda_1(\omega,r)$ and $\lambda_2(\omega,r)$ are the roots of quadratic Eq. (15), should be minimized. For doing this it is enough to put $|\lambda_1(\omega,r)|=|\lambda_2(\omega,r)|$.

Case I: According to the discussion in [20], setting the discriminant of Eq. (15) equal to zero is sufficient to $\max\{|\lambda_1(\omega,r)|,|\lambda_2(\omega,r)|\}$ be minimized. So ω and r_{opt} satisfy

$$\xi_1(\omega,r_{opt})^2-4\xi_2(\omega,r_{opt})=0,$$

or

$$(-\omega\,r_{opt}\,\sigma^2-\omega\,\eta-r_{opt}\,\eta+\omega+r_{opt}-2)^2-4((r_{opt}\,\eta-r_{opt}+1)\,(\omega\,\eta-\omega+1))=0,$$

or

$$\begin{aligned}\omega^2r_{opt}^2\sigma^4+2\eta\omega^2r_{opt}\sigma^2+2\eta\omega r_{opt}^2\sigma^2-2\omega^2r_{opt}\sigma^2-2\omega r_{opt}^2\sigma^2+\eta^2\omega^2-2\eta^2\omega r_{opt}\\+\eta^2r_{opt}^2+4\omega r_{opt}\sigma^2-2\eta\omega^2+4\eta\omega r_{opt}-2\eta r_{opt}^2+\omega^2-2\omega r_{opt}+r_{opt}^2=0.\end{aligned}$$

It can easily be seen that the above equality is equivalent by

$$\left(\omega\sigma^2+\eta-1\right)^2r_{opt}^2+(2\eta\omega^2\sigma^2-2\omega^2\sigma^2-2\eta^2\omega+4\omega\sigma^2+4\eta\omega-2\omega)r_{opt}+\omega^2(\eta-1)^2=0. \tag{21}$$

Now solving (21) for $r_{opt}(\omega)$ gives (16) and (17). In this case $|\lambda(\omega,r_{opt})|$ can be determined by

$$|\lambda(\omega,r_{opt}(\omega))|=\left|\frac{\xi_1(\omega,r_{opt}(\omega))}{2}\right|.$$

On the other hand $|\lambda(\omega,r_{opt}(\omega))|$ is minimum if $\left|\omega\, r_{opt}(\omega)\,\sigma^2+\omega\,\eta+r_{opt}(\omega)\,\eta-\omega-r_{opt}(\omega)+2\right|$, be minimized.

Case II: It is enough to put

$$\xi_1(\omega,r_{opt})=0\Rightarrow-\omega\,r_{opt}\,\sigma^2-\omega\,\eta-r_{opt}\,\eta+\omega+r_{opt}-2=0,$$

hence

$$r_{opt}(\omega)=\frac{\omega\,\eta-\omega+2}{-\omega\,\sigma^2-\eta+1}.$$

In this case $|\lambda(\omega,r_{opt}(\omega))|$ can be determined by

$$|\lambda(\omega,r_{opt})|=\left|\pm\sqrt{-\xi_2(\omega,r_{opt})}\right|.$$

On the other hand $|\lambda(\omega,r_{opt}(\omega))|$ is minimum if

$$\left|\pm\sqrt{-\left(r_{opt}(\omega)\,\eta-r_{opt}(\omega)+1\right)\left(\omega\,\eta-\omega+1\right)}\right|,$$

be minimized. This completes the proof of the theorem. □

Example 2.1. *Suppose* $\eta = \frac{1}{2}$ *and* $\sigma = \frac{1}{4}$. *By Theorem 2.7 we can obtain the optimal parameters of AOR-Like method.*

Case I*: From (16) and (17) we have*

$$r_{opt}(\omega_{opt}) = \frac{8(\omega_{opt} + 4 + 2\sqrt{6\omega_{opt} - 12})\omega_{opt}}{\omega_{opt}^2 - 16\omega_{opt} + 64}, \tag{22}$$

or

$$r_{opt}(\omega_{opt}) = -\frac{8(-\omega_{opt} - 4 + 2\sqrt{6\omega_{opt} - 12})\omega_{opt}}{\omega_{opt}^2 - 16\omega_{opt} + 64}. \tag{23}$$

For (22), minimization problem (18) is as follows

$$\min_{\omega} \left| \frac{\omega\sqrt{6(\omega - 2)} + 8(\omega - 2)}{\omega - 8} \right|.$$

The global minimum of this problem occurs at $\omega_{opt} = 2$. *Then (22) yields* $r_{opt} = \frac{8}{3}$.
For (23), minimization problem (18) is as follows

$$\min_{\omega} \left| \frac{\omega\sqrt{6(\omega - 2)} + 8(2 - \omega)}{\omega - 8} \right|.$$

The global minimum of this problem occurs at $\omega_{opt} = 2$ *and* $\omega_{opt} = \frac{8}{3}$. *Then (23) yields* $r_{opt} = \frac{8}{3}$ *and* $r_{opt} = 2$. *In this case we obtain*

$$(\omega_{opt}, r_{opt}, \lambda(\omega_{opt}, r_{opt})) = (2, \frac{8}{3}, 0) \quad \text{and} \quad (\omega_{opt}, r_{opt}, \lambda(\omega_{opt}, r_{opt})) = (\frac{8}{3}, 2, 0).$$

Case II*: From (19) we have*

$$r_{opt} = \frac{8(\omega - 4)}{\omega - 8} \tag{24}$$

From (24), minimization problem (18) is as follows

$$\min_{\omega} \frac{\sqrt{2}}{2} \sqrt{\left| \frac{(3\,\omega - 8)\,(\omega - 2)}{\omega - 8} \right|}. \tag{25}$$

The global minimum of this problem occurs at $\omega_{opt} = 2$ *and* $\omega_{opt} = \frac{8}{3}$. *Then (25) yields* $r_{opt} = \frac{8}{3}$ *and* $r_{opt} = 2$.

In the following, we will see that the minimum value of optimization problems (20) and (18) is zero.

Theorem 2.8. *Let the conditions ((a) or (b)) of Theorem 2.6 be satisfied and* $\eta \neq 1$. *Then the optimal parameters of AOR-Like method are given in the following*

$$\omega_{opt} = \frac{1}{1 - \eta}, \qquad r_{opt} = \frac{1 - \eta}{(1 - \eta)^2 - \sigma^2},$$

$$\omega_{opt} = \frac{1 - \eta}{(1 - \eta)^2 - \sigma^2}, \qquad r_{opt} = \frac{1}{1 - \eta},$$

and in both cases we have $\rho\left(\Phi(\omega_{opt}, r_{opt})\right) = 0$.

Proof. By putting (16) in $\xi_1(\omega, r)$ we have

$$\xi_1(\omega, r_{opt}(\omega)) = -\frac{2\left(\omega\sigma\sqrt{-\eta^3\omega + \eta\omega\sigma^2 + 3\eta^2\omega - \omega\sigma^2 - \eta^2 - 3\eta\omega + \sigma^2 + 2\eta + \omega - 1} + \eta^2\omega - 2\omega\eta + \eta + \omega - 1\right)}{\omega\sigma^2 + \eta - 1}.$$

Then solving $\xi_1(\omega, r_{opt}(\omega)) = 0$ yields:

$$\omega_{opt} = \frac{1}{1 - \eta}.$$

Substituting the above term in (16) gives:

$$r_{opt} = \frac{1-\eta}{(1-\eta)^2 - \sigma^2}.$$

Then the equation $\xi_1(\omega_{opt}, r_{opt})^2 - 4\xi_2(\omega_{opt}, r_{opt}) = 0$ concludes $\xi_2(\omega_{opt}, r_{opt}) = 0$. Therefore from (15) we get $\lambda_1(\omega_{opt}, r_{opt}) = \lambda_2(\omega_{opt}, r_{opt}) = 0$. Hence $\rho(\Phi(\omega_{opt}, r_{opt})) = 0$. By putting (17) in $\xi_1(\omega, r)$ we obtain the same result. Putting (16) in $\xi_2(\omega, r)$ yields

$$\xi_2(\omega, r) = -\frac{\left(\eta^2\omega - \omega\sigma^2 - 2\eta\omega + \eta + \omega - 1\right)(\eta\omega - \omega + 1)}{\omega\sigma^2 + \eta - 1}.$$

Then solving $\xi_2(\omega, r_{opt}(\omega)) = 0$ yields:

$$\omega_{opt} = \frac{1}{1-\eta}, \quad and \quad \omega_{opt} = \frac{1-\eta}{(1-\eta)^2 - \sigma^2}.$$

Then by (16) and the above equation we get

$$r_{opt} = \frac{1-\eta}{(1-\eta)^2 - \sigma^2}, \quad and \quad r_{opt} = \frac{1}{1-\eta}.$$

Similarly we obtain $\rho(\Phi(\omega_{opt}, r_{opt})) = 0$. □

2.2. Application in solving Helmholtz differential equation

Consider the following Helmholtz equation by Dirichlet boundary conditions [12, 31]:

$$\begin{cases} -\dfrac{\partial^2 u}{\partial x^2} - \dfrac{\partial^2 u}{\partial y^2} + \sigma_1 u + i\sigma_2 u = f, & \text{on } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where Ω is a unique square $\Omega = \{(x, y) \mid 0 < x, y < 1\}$, $\partial\Omega$ is the boundary of this area, σ_1 and σ_2 are real coefficient functions, and $i = \sqrt{-1}$. In addition, we consider f to be a continuous function on Ω. Suppose n is an integer positive number that is already known. Consider the following set of points

$$\overline{\Omega_h} := \{(jh,\ kh) \mid j,\ k = 0,\ 1, \ldots,\ n+1\},$$

$$\Omega_h := \{(jh,\ kh) \mid j,\ k = 0,\ 1,\ 2, \ldots,\ n\},$$

where $h = \frac{1}{n+1}$ indicates the length of the step. The set Ω_h is the inner of this area and $\overline{\Omega_h} - \Omega_h$ will be boundary points. Let $u_{j,k} = u(jh,\ kh)$ be the exact answer of the equation at the point $(j,\ k)$ and $U_{j,k}$ be an approximation of it. To approximate the second derivative u rather than x and y, we get help from the following central approximations

$$\frac{\partial^2 u(x,y)}{\partial x^2}|_{(x=jh, y=kh)} \simeq \frac{U_{j-1,k} - 2U_{jk} + U_{j+1,k}}{h^2},$$

$$\frac{\partial^2 u(x,y)}{\partial y^2}|_{(x=jh, y=kh)} \simeq \frac{U_{j,k-1} - 2U_{jk} + U_{j,k+1}}{h^2}.$$

Therefore, the given problem is discretized as follows

$$-\frac{U_{j-1,k} - 2U_{jk} + U_{j+1,k}}{h^2} - \frac{U_{j,k-1} - 2U_{jk} + U_{j,k+1}}{h^2} + \sigma_1 U_{jk} + i\sigma_2 U_{jk} = f_{jk}, \quad j,\ k = 1, 2, \ldots, n.$$

This implies

$$(4 + h^2(\sigma_1 + i\sigma_2))U_{jk} - U_{j-1,k} - U_{j+1,k} - U_{j,k-1} - U_{j,k+1} = h^2 f_{jk}, \quad j,\ k = 1, 2, \ldots, n.$$

These equations can be written as follows

$$j = 1, \quad \begin{cases} k = 1 : (4 + h^2(\sigma_1 + i\sigma_2))U_{11} - U_{01} - U_{21} - U_{10} - U_{12} = h^2 f_{11}, \\ k = 2 : (4 + h^2(\sigma_1 + i\sigma_2))U_{12} - U_{02} - U_{22} - U_{11} - U_{13} = h^2 f_{12}, \\ \quad \vdots \\ k = n : (4 + h^2(\sigma_1 + i\sigma_2))U_{1n} - U_{0n} - U_{2n} - U_{1,k-1} - U_{1,n+1} = h^2 f_{1n}, \end{cases}$$

$$j=2,\quad \begin{cases} k=1:(4+h^2(\sigma_1+i\sigma_2))U_{21}-U_{11}-U_{31}-U_{20}-U_{22}=h^2f_{21},\\ k=2:(4+h^2(\sigma_1+i\sigma_2))U_{22}-U_{12}-U_{32}-U_{21}-U_{23}=h^2f_{22},\\ \quad\vdots\\ k=n:(4+h^2(\sigma_1+i\sigma_2))U_{2n}-U_{1n}-U_{3n}-U_{2,n-1}-U_{2,n+1}=h^2f_{2n}, \end{cases}$$

$$\vdots$$

$$j=n,\quad \begin{cases} k=1:(4+h^2(\sigma_1+i\sigma_2))U_{n1}-U_{n-1,1}-U_{n+1,1}-U_{n0}-U_{n2}=h^2f_{n1},\\ k=2:(4+h^2(\sigma_1+i\sigma_2))U_{n2}-U_{n-1,2}-U_{n+1,2}-U_{n1}-U_{n3}=h^2f_{n2},\\ \quad\vdots\\ k=n:(4+h^2(\sigma_1+i\sigma_2))U_{nn}-U_{n-1,n}-U_{n+1,n}-U_{n,n-1}-U_{n,n+1}=h^2f_{nn}. \end{cases}$$

We obtain by using boundary conditions

$$j=1,\quad \begin{cases} k=1:(4+h^2(\sigma_1+i\sigma_2))U_{11}-0-U_{21}-0-U_{12}=h^2f_{11},\\ k=2:(4+h^2(\sigma_1+i\sigma_2))U_{12}-0-U_{22}-U_{11}-U_{13}=h^2f_{12},\\ \quad\vdots\\ k=n:(4+h^2(\sigma_1+i\sigma_2))U_{1n}-0-U_{2n}-U_{1,k-1}-0=h^2f_{1n}, \end{cases}$$

$$j=2,\quad \begin{cases} k=1:(4+h^2(\sigma_1+i\sigma_2))U_{21}-U_{11}-U_{31}-0-U_{22}=h^2f_{21},\\ k=2:(4+h^2(\sigma_1+i\sigma_2))U_{22}-U_{12}-U_{32}-U_{21}-U_{23}=h^2f_{22},\\ \quad\vdots\\ k=n:(4+h^2(\sigma_1+i\sigma_2))U_{2n}-U_{1n}-U_{3n}-U_{2,n-1}-0=h^2f_{2n}, \end{cases}$$

$$\vdots$$

$$j=n,\quad \begin{cases} k=1:(4+h^2(\sigma_1+i\sigma_2))U_{n1}-U_{n-1,1}-0-0-U_{n2}=h^2f_{n1},\\ k=2:(4+h^2(\sigma_1+i\sigma_2))U_{n2}-U_{n-1,2}-0-U_{n1}-U_{n3}=h^2f_{n2},\\ \quad\vdots\\ k=n:(4+h^2(\sigma_1+i\sigma_2))U_{nn}-U_{n-1,n}-0-U_{n,n-1}-0=h^2f_{nn}. \end{cases}$$

Now these equations can be represented in the following matrix form ($\gamma:=2+h^2(\frac{1}{2}\sigma_1+\frac{i}{2}\sigma_2)$)

$$\begin{bmatrix} \gamma U_{11}-U_{21} & \gamma U_{12}-U_{22} & \cdots & \gamma U_{1n}-U_{2n}\\ -U_{11}+\beta U_{21}-U_{31} & -U_{12}+\gamma U_{22}-U_{32} & \cdots & -U_{1n}+\gamma U_{2n}-U_{3n}\\ \vdots & \vdots & \ddots & \vdots\\ -U_{n-1,1}+\gamma U_{n1} & -U_{n-1,2}+\gamma U_{n2} & \cdots & -U_{n-1,n}+\gamma U_{nn} \end{bmatrix}$$

$$+\begin{bmatrix} \gamma U_{11}-U_{12} & -U_{11}+\gamma U_{12}-U_{13} & \cdots & -U_{1,n-1}+\gamma U_{1n}\\ \gamma U_{21}-U_{22} & -U_{21}+\gamma U_{22}-U_{23} & \cdots & -U_{2,n-1}+\gamma U_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \gamma U_{n1}-U_{n2} & -U_{n1}+\gamma U_{n2}-U_{n3} & \cdots & -U_{n,n-1}+\gamma U_{nn} \end{bmatrix}=h^2\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n}\\ f_{21} & f_{22} & \cdots & f_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix},$$

or

$$\begin{bmatrix} \gamma & -1 & 0 & \dots & 0\\ -1 & \gamma & -1 & \dots & 0\\ 0 & \ddots & \ddots & \ddots & \vdots\\ \vdots & & \ddots & \ddots & -1\\ 0 & \dots & \dots & -1 & \gamma \end{bmatrix}\begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n}\\ U_{21} & U_{22} & \dots & U_{2n}\\ \vdots & \ddots & & \vdots\\ U_{n1} & U_{n2} & \dots & U_{nn} \end{bmatrix}+\begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n}\\ U_{21} & U_{22} & \dots & U_{2n}\\ \vdots & \ddots & & \vdots\\ U_{n1} & U_{n2} & \dots & U_{nn} \end{bmatrix}\begin{bmatrix} \gamma & -1 & 0 & \dots & 0\\ -1 & \gamma & -1 & \dots & 0\\ 0 & \ddots & \ddots & \ddots & \vdots\\ \vdots & & \ddots & \ddots & -1\\ 0 & \dots & \dots & -1 & \gamma \end{bmatrix}$$

$$=h^2\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n}\\ f_{21} & f_{22} & \cdots & f_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix},$$

or

$$AU + UA = h^2 F, \tag{26}$$

where

$$U = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ U_{21} & U_{22} & \dots & U_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ U_{n1} & U_{n2} & \dots & U_{nn} \end{bmatrix}_{n\times n}, \quad F = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{n1} & f_{n2} & \dots & f_{nn} \end{bmatrix}_{n\times n}, \quad A = \begin{bmatrix} \gamma & -1 & 0 & \dots & 0 \\ -1 & \gamma & -1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \dots & \dots & -1 & \gamma \end{bmatrix}_{n\times n}.$$

It is clear that solving the Helmholtz equation by finite difference method has led to solving a Lyapunov matrix equation with a complex coefficient matrix. Moreover equation (26) is a special case of (1) studied here. Note that the matrix of coefficients of this Lyapunov equation can be split as follows:

$$A = W + iT,$$

where

$$W = \begin{bmatrix} 2+\frac{h^2}{2}\sigma_1 & -1 & 0 & \dots & 0 \\ -1 & 2+\frac{h^2}{2}\sigma_1 & -1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \dots & \dots & -1 & 2+\frac{h^2}{2}\sigma_1 \end{bmatrix}, \quad T = \begin{bmatrix} \frac{h^2}{2}\sigma_2 & 0 & 0 & \dots & 0 \\ 0 & \frac{h^2}{2}\sigma_2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \frac{h^2}{2}\sigma_2 \end{bmatrix}.$$

3. Test problems

All numerical experiments in this section were carried out using MATLAB software, running on a computer equipped with an Intel (R) Pentium (R) CPU N3700 (1.60 GHz) and 4 GB RAM.

Example 3.1. *Here we will solve matrix equation*

$$AX + XB + \underbrace{((15+0.5i)I_n)}_{N_1} X (\underbrace{(15+0.5i)I_n}_{M_1})^T + \underbrace{((6-0.2i)I_n)}_{N_2} X (\underbrace{(6-0.2i)I_n}_{M_2})^T = C, \tag{27}$$

where $A = W + iT$, $B = A^T = W^T + iT^T$ *with*

$$W = I_m \otimes V_m + V_m \otimes I_m - \pi^2 I_n, \quad (n = m^2),$$
$$T = 10\pi I_n + \mu \left(I_m \otimes V_m + V_m \otimes I_m \right),$$

and

$$V_m = \frac{1}{h^2} Tridiagonal(-1, 2, -1)_{m\times m},$$

with $h = 1/(m+1)$ *and* μ *is a positive parameter with values* $\mu = 1$ *and* $\mu = 10$. *Also consider right hand side matrix* C *such that* $X^\star = (x^\star_{i,j})$ *with*

$$x^\star_{i,j} = \sin(x(i)) + \sin(y(j)), \qquad i, j = 1, 2, \dots, m^2, \tag{28}$$

is exact solution of matrix equation (27), *where* $x_i = -2 + 4(i-1)/(m^2-1)$ *and* $y_j = -2 + 4(j-1)/(m^2-1)$, $i, j = 1, 2, \dots, m^2$. *The stopping criteria for iterations is*

$$S(k) \le 5 \times 10^{-5}, \tag{29}$$

where

$$S(k) := \frac{\left\| AX^{(k)} + X^{(k)}B + N_1 X^{(k)} M_1 + N_2 X^{(k)} M_2 - C \right\|_F}{\left\| AX^{(0)} + X^{(0)}B + N_1 X^{(0)} M_1 + N_2 X^{(0)} M_2 - C \right\|_F},$$

and $X^{(0)}$ *is an initial guess that here is zero matrix. In each step of new method we solve two standard real Sylvester matrix equations via Bartels-Stewart algorithm [7].*

Table 1: The comparison of iteration number (IT) and CPU time for Example 3.1.

Sizes of coefficient matrices	64×64	100×100	225×225	400×400
$\mu = 1$				
SOR-Like				
α_{exp}	0.1143	0.1143	0.1143	0.1143
IT	43	49	59	65
CPU(s)	0.830457	2.699985	34.983288	197.890202
AOR-Like				
ω_{exp}	0.09	0.09	0.09	0.09
r_{exp}	0.07	0.07	0.07	0.07
IT	21	22	24	35
CPU(s)	0.466266	1.153980	10.788853	103.171439
MHSS [38]				
$\alpha_{exp} = \beta_{exp}$	260	260	260	260
IT	16	16	25	37
CPU(s)	0.632790	1.298898	24.332186	217.143481
GMHSS [38]				
$\alpha_{exp} = \beta_{exp}$	260	260	260	260
$\tau_{exp} = \theta_{exp}$	270	270	270	270
IT	14	15	23	35
CPU(s)	0.439721	1.313612	21.689607	203.071587

Table 2: The comparison of iteration number (IT) and CPU time for Example 3.1.

Sizes of coefficient matrices	64×64	100×100	225×225	400×400
$\mu = 10$				
SOR-Like				
α_{exp}	0.05	0.05	0.05	0.05
IT	61	123	145	158
CPU(s)	1.215940	6.307189	68.495753	513.966249
AOR-Like				
ω_{exp}	0.02	0.02	0.02	0.02
r_{exp}	0.09	0.09	0.09	0.09
IT	20	21	25	36
CPU(s)	0.546207	1.024469	10.495669	107.481687
MHSS [38]				
$\alpha_{exp} = \beta_{exp}$	730	730	730	730
IT	23	24	26	30
CPU(s)	0.712993	2.032919	23.640414	186.367522
GMHSS [38]				
$\alpha_{exp} = \beta_{exp}$	1060	1060	1060	1060
$\tau_{exp} = \theta_{exp}$	70	70	70	70
IT	9	9	12	17
CPU(s)	0.276629	0.759973	11.922190	99.080614

Table 1 presents the numerical results obtained from our experiments, including the running time and the residual error for AOR-Like method, SOR-Like method [20], MHSS and the GMHSS methods [38]. As shown in Table 1, the method under consideration exhibits a shorter time to reach the desired level of accuracy compared to SOR-Like, MHSS and the GMHSS methods. This indicates the effectiveness and efficiency of the new method. In fact the AOR-Like method was found to be faster than the method in [20].

Tables 1 and 2 present the numerical results of MHSS and GMHSS methods. The AOR-Like method consistently achieves faster convergence to the exact solution of the matrix equation compared to other methods, as demonstrated by the data presented. The validity of the assumption can be reinforced by examining the data presented in these tables.

Figures 1 and 2 illustrate the approximation of the optimal parameters for both methods when the problem size is 225×225. *The optimal parameter for the method in [20] was determined to be* $\alpha_{opt} \approx 0.1$ *for* $\mu = 1$ *and* $\alpha_{opt} \approx 0.05$ *for* $\mu = 10$, *while for the AOR-Like method, the optimal parameters were found to be* $\omega_{opt} \approx 0.09, ; r_{opt} \approx 0.08$ *for* $\mu = 0.01, 0.1, 1$ *and* $\omega_{opt} \approx 0.02; r_{opt} \approx 0.09$ *for* $\mu = 10$. *It should be noted that the optimal parameters were obtained experimentally by minimizing the number of iterations in Figs. 1 and 2. Moreover, the optimal parameters for both methods were observed to be fixed with increasing problem size, as can be seen in Table 1.*

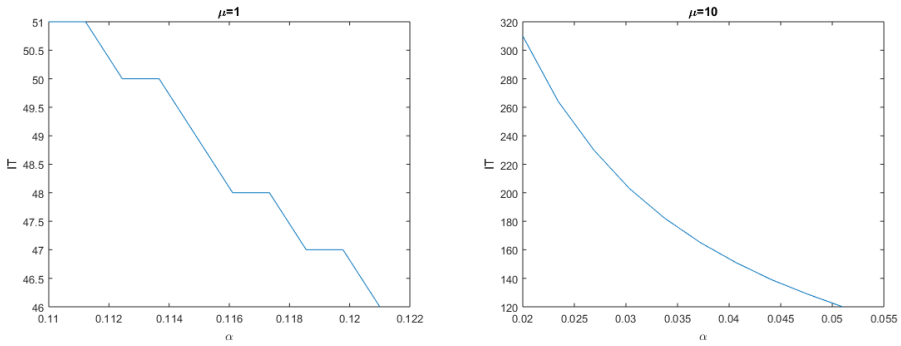


Figure 1: The parameter α versus iteration numbers for FRSI (SOR-Like) method for Example 3.1.

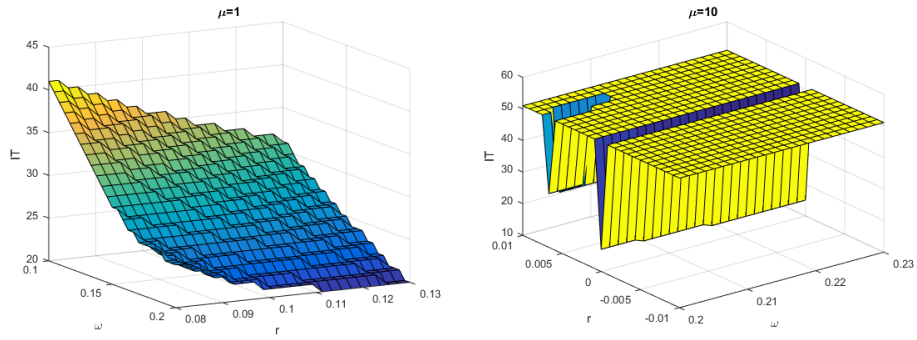


Figure 2: The parameters ω and r versus iteration numbers for AOR-Like for Example 3.1.

Figure 3 shows the relationship between the parameters ω *and* r *and the spectral radius of the iteration matrix for AOR-Like method.*

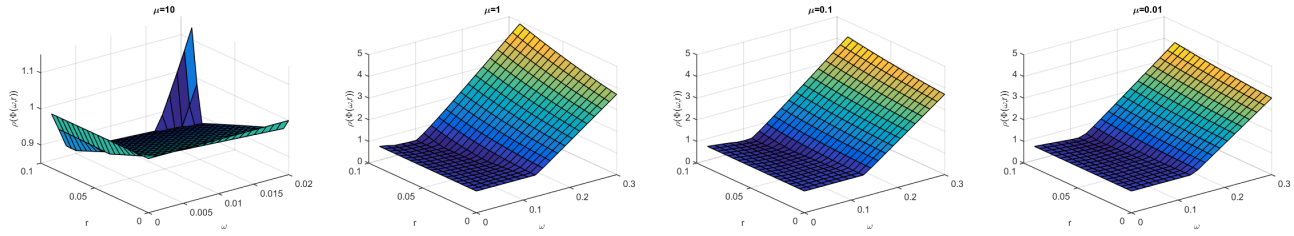


Figure 3: The parameters ω and r versus spectral radius of iteration matrix for AOR-Like for Example 3.1.

In Figure 4, we can see the approximate solutions for the imaginary and real parts obtained using the AOR-Like method after 2, 5, and 20 iterations. It can be observed that by increasing the number of iterations, the sequence of matrices $\{X^{(k)}\}_{k=0}^{\infty}$ *converges to the real solution (28).*

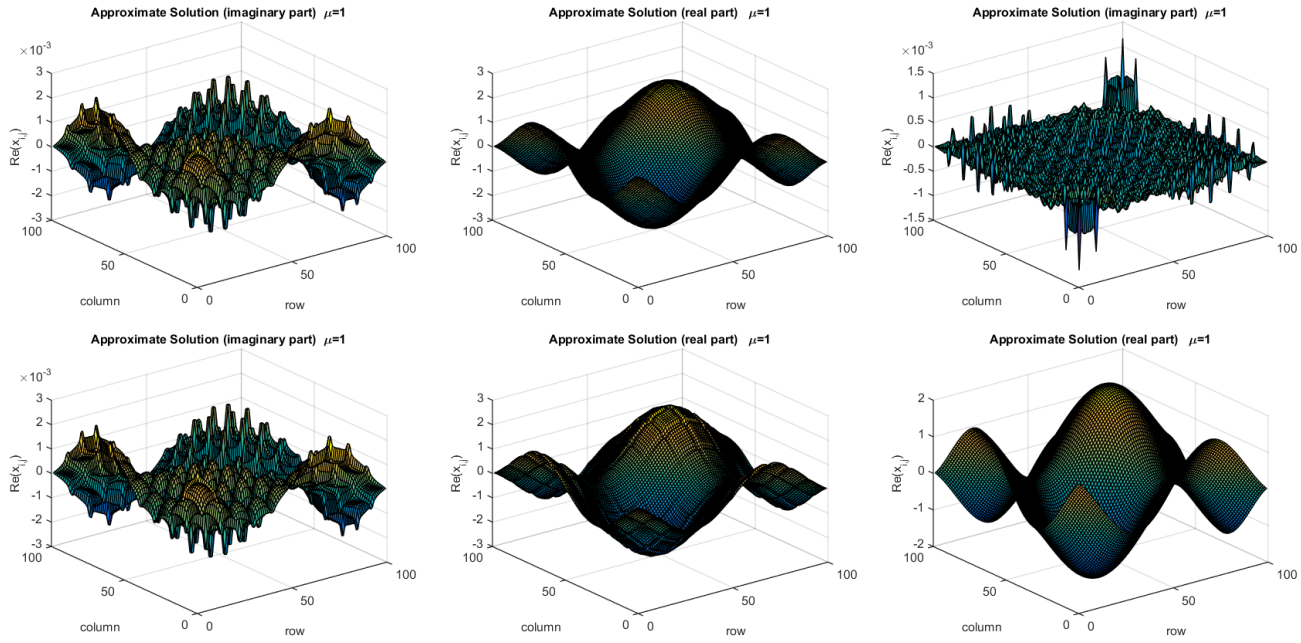


Figure 4: Approximate solutions for imaginary and real parts with size $n \times n = 100 \times 100$; Top (after 20 iterations); Middle (after 40 iterations); Bottom (after 80 iterations) for Example 3.1.

On the other hand, Figure 5 displays the eigenvalue distribution of SOR-like iteration matrix for problem sizes of 500×500 *and different values of the parameter* μ*. Based on the information presented in the figure, it can be concluded that for* $\mu = 0.01$*, the eigenvalues of the iteration matrix of the AOR-Like method approach zero. This suggests that in this specific scenario, the method exhibits a high convergence speed.*

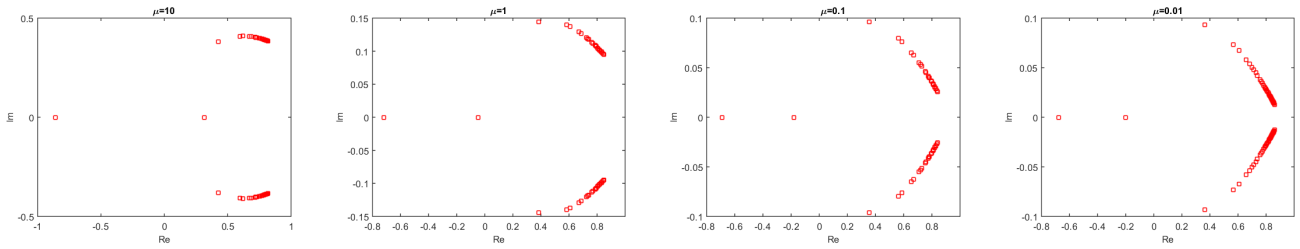


Figure 5: The eigenvalue distribution of the iteration matrix for Example 3.1.

The results clearly demonstrate that the method being evaluated outperforms the SOR-Like, MHSS, and GMHSS methods in terms of the time required to achieve the desired level of accuracy. This evidence supports the notion that the new method is effective and efficient.

Example 3.2. *Consider the equation*

$$(W + iT)X + X(W + iT)^T + v\left(N_1 X N_1^T + N_2 X N_2^T\right) = C,$$

where

$$T = \begin{bmatrix} 10 & 1 & 1 & -1 & 2 & & & & & \boldsymbol{O} \\ 1 & 10 & 1 & 1 & -1 & 2 & & & & \\ 1 & 1 & 10 & 1 & 1 & -1 & 2 & & & \\ -1 & 1 & 1 & 10 & 1 & 1 & -1 & 2 & & \\ 2 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 2 \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & -1 \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 \\ & & & & 2 & -1 & 1 & 1 & 10 & 1 \\ \boldsymbol{O} & & & & & 2 & -1 & 1 & 1 & 10 \end{bmatrix}_{n\times n}, \quad W = \begin{bmatrix} 2 & 0.1 & 0.2 & 0.3 & 0.1 & & & & & \boldsymbol{O} \\ 0.1 & 2 & 0.1 & 0.2 & 0.3 & 0.1 & & & & \\ 0.2 & 0.1 & 2 & 0.1 & 0.2 & 0.3 & 0.1 & & & \\ 0.3 & 0.2 & 0.1 & 2 & 0.1 & 0.2 & 0.3 & 0.1 & & \\ 0.1 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & 0.1 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0.1 \\ & & 0.1 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0.3 \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0.2 \\ & & & & 0.1 & 0.3 & 0.2 & 0.1 & 2 & 0.1 \\ \boldsymbol{O} & & & & & 0.1 & 0.3 & 0.2 & 0.1 & 2 \end{bmatrix}_{n\times n},$$

$$N_1 = \begin{bmatrix} 2 & -0.1 & 0.3 & & & & & & & \boldsymbol{O} \\ 0.3 & 2 & -0.1 & 0.3 & & & & & & \\ -0.2 & 0.3 & 2 & -0.1 & 0.3 & & & & & \\ & -0.2 & 0.3 & 2 & -0.1 & 0.3 & & & & \\ & & -0.2 & 0.3 & 2 & -0.1 & 0.3 & & & \\ & & & -0.2 & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & & \ddots & \ddots & \ddots & \ddots & 0.3 \\ & & & & & & \ddots & \ddots & \ddots & -0.1 \\ \boldsymbol{O} & & & & & & & -0.2 & 0.3 & 2 \end{bmatrix}_{n\times n},$$

and

$$N_2 = \begin{bmatrix} 1 & 0.2 & 0.1 & & & & & & & \boldsymbol{O} \\ 0.3 & 1 & 0.2 & 0.1 & & & & & & \\ -0.1 & 0.3 & 1 & 0.2 & 0.1 & & & & & \\ & -0.1 & 0.3 & 1 & 0.2 & 0.1 & & & & \\ & & -0.1 & 0.3 & 1 & 0.2 & 0.1 & & & \\ & & & -0.1 & \ddots & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & & \ddots & \ddots & \ddots & \ddots & 0.2 \\ & & & & & & \ddots & \ddots & \ddots & 0.1 \\ \boldsymbol{O} & & & & & & & -0.1 & 0.3 & 1 \end{bmatrix}_{n\times n}.$$

Also right hand side matrix C is such that $Z = (\boldsymbol{z}_{i,j})$ with

$$\boldsymbol{z}_{i,j} = \sin^2(x_i) + \cos^2(y_j), \qquad i,j = 1,2,\ldots,n, \tag{30}$$

can be exact solution, where $x_i = -1 + 2(i-1)/(n-1)$ and $y_j = -1 + 2(j-1)/(n-1)$, $\quad i,j = 1,2,\ldots,n$.

We performed numerical experiments to solve the matrix equation for various values of v and n. The resulting numerical results, including running time and residual error, are presented in Tables 3 and 4.

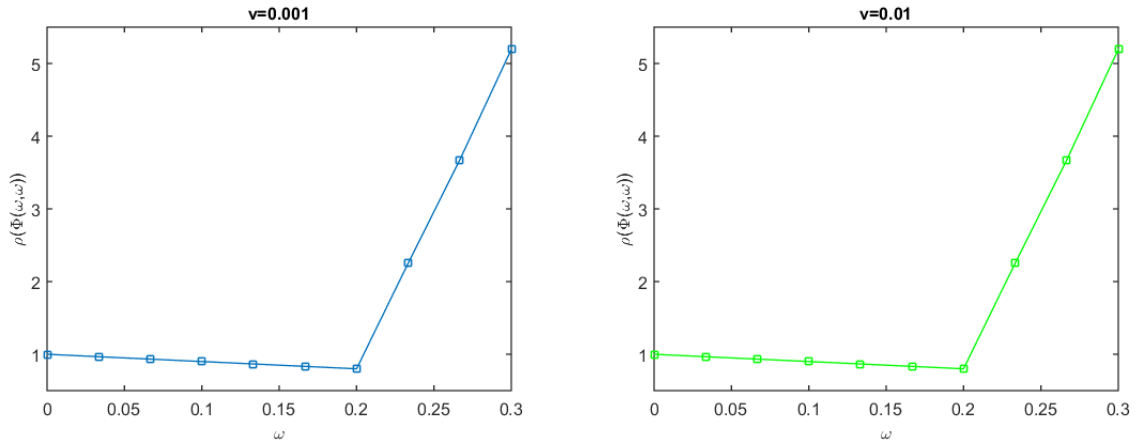
Table 3: The iteration number (IT) and CPU time for SOR-Like method for Example 3.2.

Sizes of coefficient matrices	40×40	80×80	160×160	320×320
$v=1$				
α_{exp}	0.16	0.16	0.16	0.16
IT	27	27	27	27
CPU(s)	0.294009	0.975645	5.119216	36.610712
S(.)	3.0621×10^{-5}	3.1146×10^{-5}	3.1373×10^{-5}	3.1486×10^{-5}
$v=0.1$				
α_{exp}	0.18	0.18	0.18	0.18
IT	62	62	59	59
CPU(s)	0.658270	2.195957	11.298851	72.903858
S(.)	3.9238×10^{-5}	3.9264×10^{-5}	4.9950×10^{-5}	4.9785×10^{-5}
$v=0.01$				
α_{exp}	0.18	0.18	0.18	0.18
IT	62	62	62	62
CPU(s)	0.647823	2.603903	15.073012	92.159660
S(.)	4.5062×10^{-5}	4.5127×10^{-5}	4.5114×10^{-5}	4.5108×10^{-5}
$v=0.001$				
α_{exp}	0.18	0.18	0.18	0.18
IT	62	62	62	62
CPU(s)	0.636985	2.308290	15.425496	92.797011
S(.)	4.5124×10^{-5}	4.5189×10^{-5}	4.5176×10^{-5}	4.5170×10^{-5}

Table 4: The iteration number (IT) and CPU time for AOR-Like method for Example 3.2.

Sizes of coefficient matrices	40×40	80×80	160×160	320×320
$v = 1$				
ω_{exp}	0.18	0.18	0.18	0.18
r_{exp}	0.15	0.15	0.15	0.15
IT	24	24	24	24
CPU(s)	0.301654	0.912333	4.654909	30.080706
S(.)	3.6749×10^{-5}	2.7627×10^{-5}	2.6201×10^{-5}	2.5573×10^{-5}
$v = 0.1$				
ω_{exp}	0.15	0.15	0.15	0.15
r_{exp}	0.25	0.25	0.25	0.25
IT	44	44	44	44
CPU(s)	0.503273	2.160103	11.393349	63.145133
S(.)	4.3809×10^{-5}	4.2734×10^{-5}	4.2557×10^{-5}	4.2463×10^{-5}
$v = 0.01$				
ω_{exp}	0.15	0.15	0.15	0.15
r_{exp}	0.25	0.25	0.25	0.25
IT	44	44	44	44
CPU(s)	0.538805	1.801776	10.737216	58.381569
S(.)	4.8505×10^{-5}	4.6817×10^{-5}	4.6428×10^{-5}	4.6233×10^{-5}
$v = 0.001$				
ω_{exp}	0.15	0.15	0.15	0.15
r_{exp}	0.25	0.25	0.25	0.25
IT	44	44	44	44
CPU(s)	0.526946	1.504894	11.878253	49.306928
S(.)	4.8556×10^{-5}	4.6861×10^{-5}	4.6470×10^{-5}	4.6274×10^{-5}

Our experiments led us to approximate the optimal parameter for SOR-like method, which we denote as ω_{opt}. This value was obtained experimentally by minimizing the spectral radius of SOR-like iteration matrix. In Tables 3 and 4, we used ω_{opt} to obtain our numerical results, and we observed that the optimal parameter remained fixed as we increased the problem size. Furthermore, we found that SOR-like method was slower than AOR-like method, as shown in these tables.



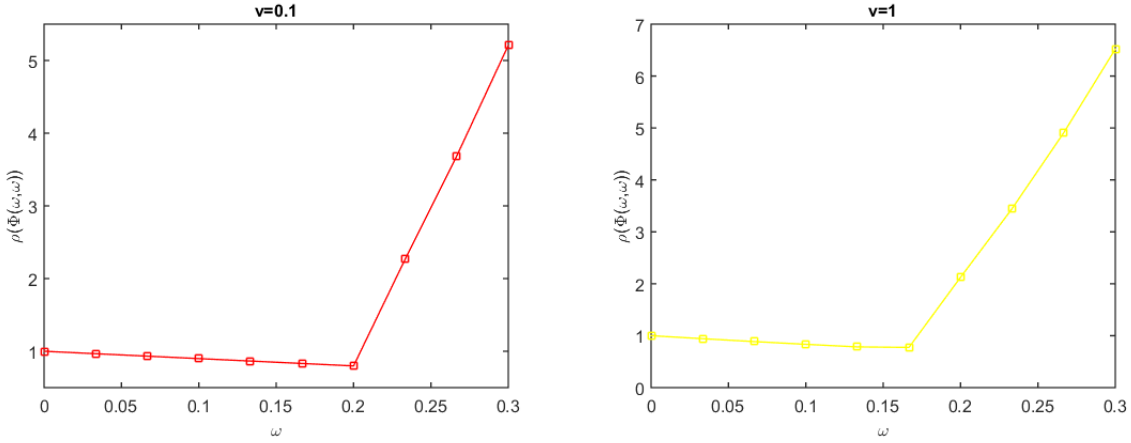


Figure 6: The parameter α versus spectral radius of iteration matrix for FRSI (SOR-Like) for Example 3.2.

Figure 6 shows the parameter α versus spectral radius of iteration matrix for FRSI (SOR-Like) for this test problem by $v = 0.001, 0.01, 0.1$ and $v = 1$.

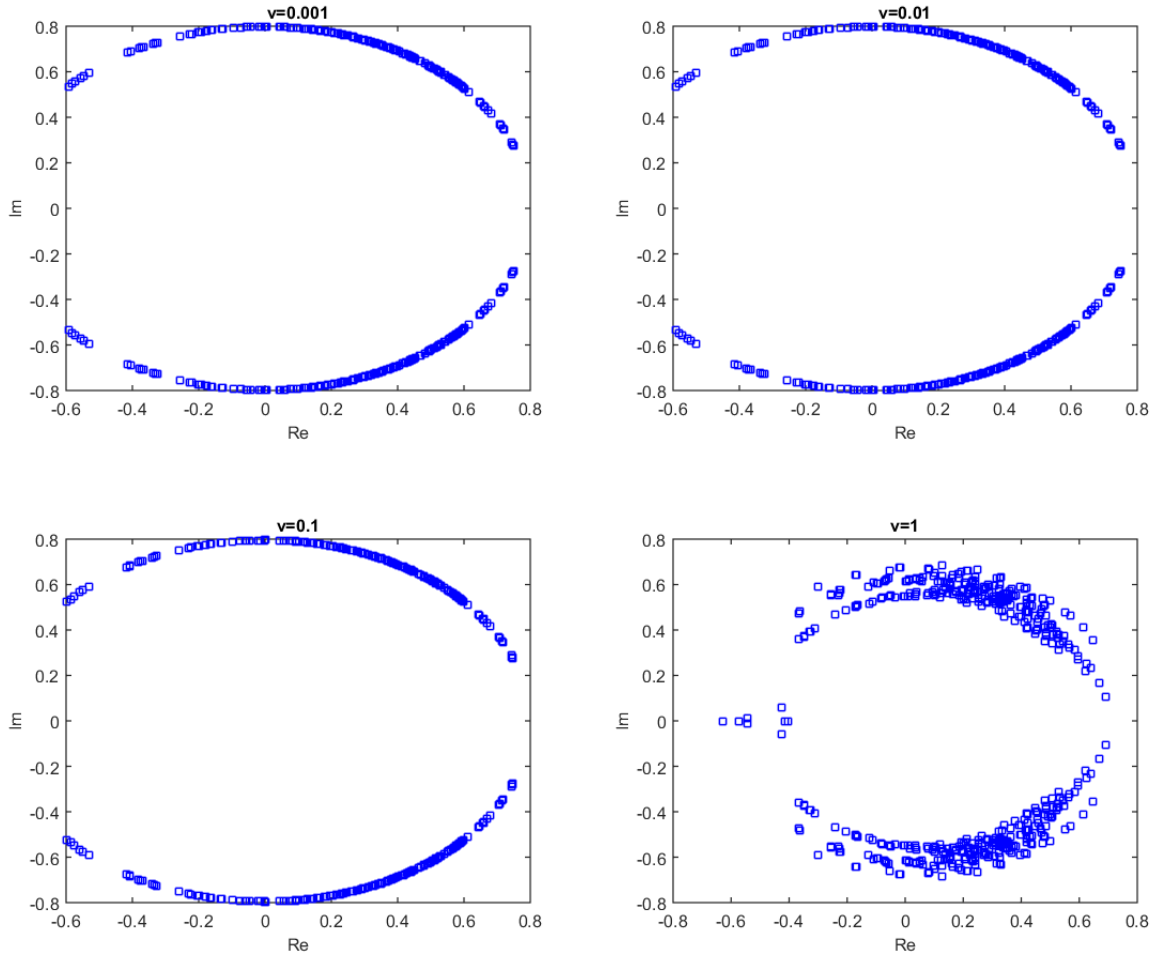


Figure 7: The eigenvalue distribution of the iteration matrix for Example 3.2.

We also investigated the eigenvalue distribution of AOR-like iteration matrix for a fixed size of 500×500 and various values of the parameter v, and the results are plotted in Figure 7. Based on the information presented in the figure, it can be concluded that for $v = 1$, the eigenvalues of the iteration matrix of the AOR-Like method approach zero. This suggests that in this specific scenario, the method exhibits a high convergence speed.

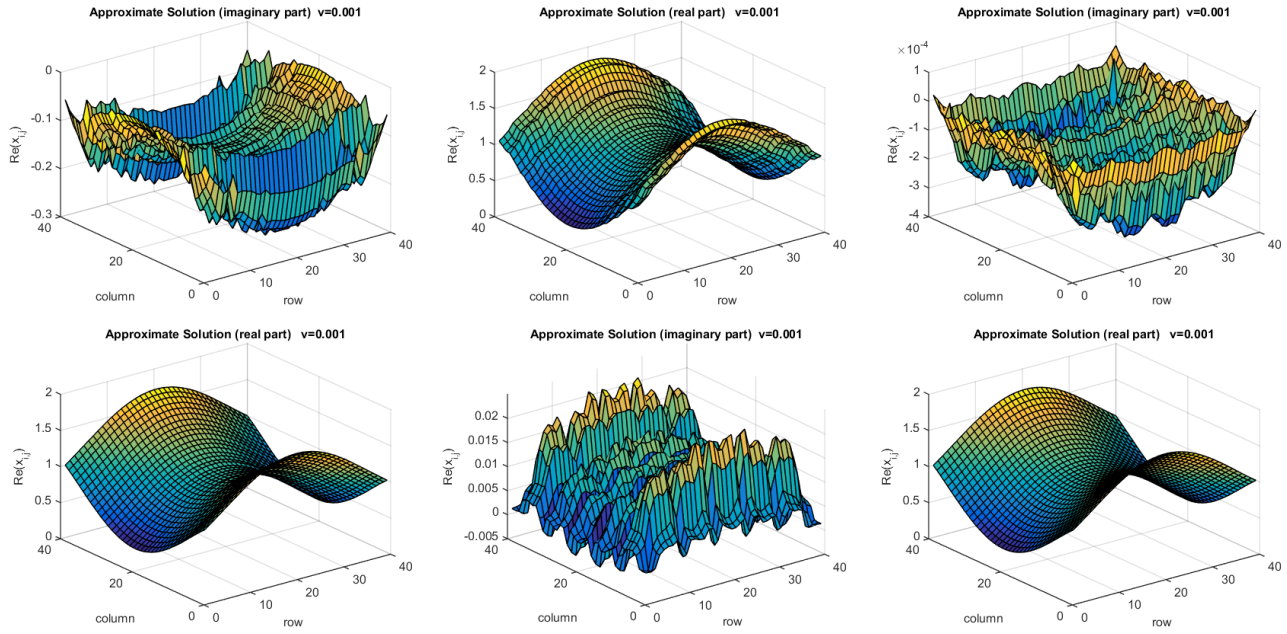


Figure 8: Approximate solutions for imaginary and real parts with size $n \times n = 100 \times 100$; Top (after 10 iterations); Middle (after 20 iterations); Bottom (after 40 iterations) for Example 3.2.

In addition, we plotted the approximate solutions for the real and imaginary parts of AOR-like method after 10, 20, and 40 iterations in Figure 8. We observed that the sequence of matrices $X^{(0)}$ converged to the real solution (30) as the number of iterations increased.

Table 5: The iteration number (IT) and CPU time for MHSS method [38] for Example 3.2.

Sizes of coefficient matrices	40×40	80×80	160×160	320×320
$\underline{v = 1}$				
$\alpha_{exp} = \beta_{exp}$	12	12	13	13
IT	19	19	19	19
CPU(s)	0.245659	1.103787	5.214515	72.710017
S(.)	3.3682×10^{-5}	3.4671×10^{-5}	4.9147×10^{-5}	4.9388×10^{-5}
$\underline{v = 0.1}$				
$\alpha_{exp} = \beta_{exp}$	7	7	7.2	7.2
IT	18	18	18	18
CPU(s)	0.253777	0.866306	5.581671	71.584671
S(.)	3.9207×10^{-5}	3.9067×10^{-5}	3.8841×10^{-5}	3.8801×10^{-5}
$\underline{v = 0.01}$				
$\alpha_{exp} = \beta_{exp}$	6	6	6.3	6.3
IT	18	18	18	18
CPU(s)	0.248516	0.854497	4.997806	75.380239
S(.)	3.9981×10^{-5}	4.0067×10^{-5}	3.9212×10^{-5}	3.9217×10^{-5}
$\underline{v = 0.001}$				
$\alpha_{exp} = \beta_{exp}$	6	6	6.3	6.3
IT	18	18	18	18
CPU(s)	0.260493	0.855640	7.508921	70.927791
S(.)	3.9968×10^{-5}	4.0054×10^{-5}	3.9200×10^{-5}	3.9205×10^{-5}

Table 6: The iteration number (IT) and CPU time for GMHSS method [38] for Example 3.2.

Sizes of coefficient matrices	40×40	80×80	160×160	320×320
$\underline{v = 1}$				
$\alpha_{exp} = \beta_{exp}$	14	14	14	14
$\tau_{exp} = \theta_{exp}$	4	4	4	4
IT	13	13	13	13
CPU(s)	0.192197	0.598336	3.376144	47.733990
E(.)	2.7820×10^{-5}	2.8577×10^{-5}	2.8952×10^{-5}	2.9139×10^{-5}
$\underline{v = 0.1}$				
$\alpha_{exp} = \beta_{exp}$	10.1	10.1	10.1	10.1
$\tau_{exp} = \theta_{exp}$	1.4	1.4	1.4	1.4
IT	9	9	9	9
CPU(s)	0.145050	0.420019	2.186261	35.791883
E(.)	1.8181×10^{-5}	1.8961×10^{-5}	1.9342×10^{-5}	1.9530×10^{-5}
$\underline{v = 0.01}$				
$\alpha_{exp} = \beta_{exp}$	10.1	10.1	10.1	10.1
$\tau_{exp} = \theta_{exp}$	1.4	1.4	1.4	1.4
IT	9	9	9	9
CPU(s)	0.140525	0.413385	2.312785	35.402080
E(.)	1.6187×10^{-5}	1.6880×10^{-5}	1.7219×10^{-5}	1.7386×10^{-5}
$\underline{v = 0.001}$				
$\alpha_{exp} = \beta_{exp}$	10.1	10.1	10.1	10.1
$\tau_{exp} = \theta_{exp}$	1.4	1.4	1.4	1.4
IT	9	9	9	9
CPU(s)	0.132953	0.414135	2.338413	36.057047
E(.)	1.6175×10^{-5}	1.6867×10^{-5}	1.7206×10^{-5}	1.7373×10^{-5}

Tables 5 and 6 present numerical results for MHSS and GMHSS methods, including optimal parameters, number of iterations, time required for calculation, and relative error. AOR-Like and GMHSS methods demonstrate faster performance than the other methods analyzed in this study.

The provided evidence from Tables 3–6 strengthens the validity that AOR-Like method being evaluated surpasses the SOR-Like and MHSS methods in terms of the time needed to reach the desired accuracy level. This finding supports the claim that the new method is effective and efficient, providing additional validation for the assumption. It is worth noting that GMHSS method may outperform AOR-Like method in certain scenarios. Overall, the results from the two examples provided indicate that the new method proposed in this article is efficient.

4. Conclusion

Our work presents a novel and efficient method, called AOR-Like method, for solving the generalized Sylvester matrix equation. This equation has the form $AX + XB + \sum_{j=1}^{m} N_j X M_j = C$, where $A, B, N_j, M_j (j = 1, \ldots, m)$, $C \in \mathbb{C}^{n \times n}$ are known matrices and $X \in \mathbb{C}^{n \times n}$ is the unknown matrix to be determined. We provide a convergence theorem for AOR-Like method and analyze the procedure in detail. Additionally, we discuss how to discover the optimal parameters for the method. Finally, we test the effectiveness of our method by solving a test problem.

Acknowledgments

The authors wish to thank reviewers for careful reading and valuable comments and suggestions which improved the quality of the paper.

References

[1] R. Ali, I. Khan, A. Ali, and A. Mohamed, *Two new generalized iteration methods for solving absolute value equations using M-matrix*, AIMS Math., 7 (2022), pp. 8176–8187.

[2] R. Ali, K. Pan, and A. Ali, *Two generalized successive overrelaxation methods for solving absolute value equations*, Math. Theory Appl., 40 (2020), pp. 44–55.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[4] Z.-Z. Bai, *On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations*, J. Comput. Math., 29 (2011), pp. 185–198.

[5] Z.-Z. Bai, X.-X. Guo, and S.-F. Xu, *Alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations*, Numer. Linear Algebra Appl., 13 (2006), pp. 655–674.

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

[7] R. H. Bartels and G. W. Stewart, *Algorithm 432: Solution of the matrix equation $AX + XB = C$*, Commun. ACM, 15 (1972), pp. 820–826.

[8] F. P. A. Beik, M. Najafi-Kalyani, and L. Reichel, *Iterative Tikhonov regularization of tensor equations based on the Arnoldi process and some of its generalizations*, Appl. Numer. Math., 151 (2020), pp. 425–447.

[9] P. Benner, *Factorized solution of sylvester equations with applications in control*, in Proceedings of the 16th International Symposium on Mathematical Theory of Network and Systems, B. D. Moor, B. Motmans, J. Willems, P. V. Dooren, and V. Blondel, eds., Leuven, Belgium, 5–9 July 2004.

[10] ——, *Large-scale matrix equations of special type*, Numer. Linear Algebra Appl., 15 (2008), pp. 747–754.

[11] P. Benner and T. Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math., 124 (2013), pp. 441–470.

[12] D. Bertaccini, *Efficient preconditioning for sequences of parametric complex symmetric linear systems*, Electron. Trans. Numer. Anal., 18 (2004), pp. 49–64.

[13] A. Bouhamidi and K. Jbilou, *Sylvester Tikhonov-regularization methods in image restoration*, J. Comput. Appl. Math., 206 (2007), pp. 86–98.

[14] ——, *A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications*, Appl. Math. Comput., 206 (2008), pp. 687–694.

[15] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numer. Linear Algebra Appl., 15 (2008), pp. 853–871.

[16] M. Dehghan and M. Hajarian, *An iterative algorithm for the reflexive solutions of the generalized coupled Sylvester matrix equations and its optimal approximation*, Appl. Math. Comput., 202 (2008), pp. 571–588.

[17] ——, *An iterative method for solving the generalized coupled Sylvester matrix equations over generalized bisymmetric matrices*, Appl. Math. Model., 34 (2010), pp. 639–654.

[18] ——, *Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations*, Appl. Math. Model., 35 (2011), pp. 3285–3300.

[19] M. Dehghan and A. Shirilord, *A generalized modified Hermitian and skew-Hermitian splitting (GMHSS) method for solving complex Sylvester matrix equation*, Appl. Math. Comput., 348 (2019), pp. 632–651.

[20] ——, *A new approximation algorithm for solving generalized Lyapunov matrix equations*, J. Comput. Appl. Math., 404 (2022), Paper No. 113898, 26.

[21] A. S. A. Dilip and H. K. Pillai, *Characterization of solutions of non-symmetric algebraic Riccati equations*, Linear Algebra Appl., 507 (2016), pp. 356–372.

[22] F. Ding, P. X. Liu, and J. Ding, *Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle*, Appl. Math. Comput., 197 (2008), pp. 41–50.

[23] F. Ding and H. Zhang, *Gradient-based iterative algorithm for a class of the coupled matrix equations related to control systems*, IET Control Theory Appl., 8 (2014), pp. 1588–1595.

[24] J. Ding, Y. Liu, and F. Ding, *Iterative solutions to matrix equations of the form* $A_iXB_i = F_i$, Comput. Math. Appl., 59 (2010), pp. 3500–3507.

[25] H.-Y. Fan, P. C.-Y. Weng, and E. K.-W. Chu, *Numerical solution to generalized Lyapunov/Stein and rational Riccati equations in stochastic control*, Numer. Algorithms, 71 (2016), pp. 245–272.

[26] A. Hadjidimos, *Accelerated overrelaxation method*, Math. Comp., 32 (1978), pp. 149–157.

[27] Z.-H. He, Q.-W. Wang, and Y. Zhang, *A simultaneous decomposition for seven matrices with applications*, J. Comput. Appl. Math., 349 (2019), pp. 93–113.

[28] L. B. Iantovics and F. F. Nichita, *On the colored and the set–theoretical Yang–Baxter equations*, Axioms, 10 (2021), p. 146.

[29] G. Karamali, A. Shirilord, and M. Dehghan, *On the CRI method for solving Sylvester equation with complex symmetric positive semi-definite coefficient matrices*, Filomat, 35 (2021), pp. 3071–3090.

[30] Y. Ke and C. Ma, *The alternating direction methods for solving the Sylvester-type matrix equation* $AXB + CX^\top D = E$, J. Comput. Math., 35 (2017), pp. 620–641.

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

[32] H. Mukaidani, H. Xu, and K. Mizukami, *Numerical Algorithm for Solving Cross-Coupled Algebraic Riccati Equations of Singularly Perturbed Systems*, Birkhäuser Boston, Boston, MA, 2005, pp. 545–570.

[33] M. A. Ramadan, T. S. El-Danaf, and A. M. E. Bayoumi, *A relaxed gradient based algorithm for solving extended Sylvester-conjugate matrix equations*, Asian J. Control, 16 (2014), pp. 1334–1341.

[34] R. A. Smith, *Matrix equation* $XA + BX = C$, SIAM J. Appl. Math., 16 (1968), pp. 198–201.

[35] Q.-W. Wang, *The general solution to a system of real quaternion matrix equations*, Comput. Math. Appl., 49 (2005), pp. 665–675.

[36] Q.-W. Wang, Z.-H. He, and Y. Zhang, *Constrained two-sided coupled Sylvester-type quaternion matrix equations*, Automatica J. IFAC, 101 (2019), pp. 207–213.

[37] Y. Xie, N. Huang, and C. Ma, *Iterative method to solve the generalized coupled Sylvester-transpose linear matrix equations over reflexive or anti-reflexive matrix*, Comput. Math. Appl., 67 (2014), pp. 2071–2084.

[38] J. Zhang and H. Kang, *The generalized modified Hermitian and skew-hermitian splitting method for the generalized lyapunov equation*, Int. J. Control Autom. Syst., 19 (2021), pp. 339–349.

[39] B. Zhou and G.-R. Duan, *Solutions to generalized Sylvester matrix equation by Schur decomposition*, Internat. J. Systems Sci., 38 (2007), pp. 369–375.

[40] ——, *On the generalized Sylvester mapping and matrix equations*, Systems Control Lett., 57 (2008), pp. 200–208.

Please cite this article using:

Mehdi Dehghan, Gholamreza Karamali, Akbar Shirilord, An iterative scheme for a class of generalized Sylvester matrix equations, AUT J. Math. Comput., 5(3) (2024) 195-215
https://doi.org/10.22060/AJMC.2024.22444.1159