TAIWANESE JOURNAL OF MATHEMATICS
Vol. 28, No. 2, pp. 377–396, April 2024
DOI: 10.11650/tjm/231101

A Modified Iterative Method for Solving the Non-symmetric Coupled Algebraic Riccati Equation

Li Wang* and Yibo Wang

Abstract. In this paper, a modified alternately linear implicit (MALI) iteration method is derived for solving the non-symmetric coupled algebraic Riccati equation (NCARE). In the MALI iteration algorithm, the coefficient matrices of the linear matrix equations are fixed at each iteration step. In addition, the MALI iteration method utilizes a weighted average of the estimates in both the last step and current step to update the estimates in the next iteration step. Further, we give the convergence theory of the modified algorithm. Last, numerical examples demonstrate the effectiveness and feasibility of the derived algorithm.

1. Introduction

In this paper, we study the minimal non-negative solution of the non-symmetric coupled algebraic Riccati equation (NCARE)

$$R_i(X_1, X_2, \ldots, X_s) = X_i C_i X_i - X_i D_i - A_i X_i + B_i + \sum_{j \neq i} e_{ij} X_j = 0, \tag{1.1}$$

where $X_i \in R^{m \times n}$ is the solution of the NCARE (1.1), $i \in S$, $S = \{1, 2, \ldots, s\}$ is a finite set, $A_i \in R^{m \times m}$, $B_i \in R^{m \times n}$, $C_i \in R^{n \times m}$, $D_i \in R^{n \times n}$, and e_{ij} is non-negative constant. When $s = 1$, the NCARE (1.1) changes to the non-symmetric algebraic Riccati equation (NARE)

$$R(X) = XCX - XD - AX + B = 0.$$

It is important to research the minimal non-negative solutions of the NCARE because of their broad applications in many fields. For example, for the optimal control of jump linear system, the feedback control law to minimize the quadratic performance index is obtained by solving the NCARE with some constraints [1,7,12,14]. In addition, in particle transport theory, the problem of particle transport scattering function can be transformed to get the minimal non-negative solution of the NARE [2,4,13]. Moreover, introducing

Received June 5, 2023; Accepted November 3, 2023.
Communicated by Jenn-Nan Wang.
2020 *Mathematics Subject Classification.* 15A24, 15A06, 65F10, 65F45.
Key words and phrases. non-symmetric couple algebraic Riccati equation, non-negative solution, iterative algorithm, convergence.
*Corresponding author.

Wiener–Hopf decomposition to the Markov chain is a significative step in traffic flow, and the problem of Wiener–Hopf decomposition can be converted to find the mininum non-negative solution of the NARE [19].

In order to obtain the minimal non-negative solutions of the NARE (1.1), various iteration schemes have been developed [3,8,10,11,15,20–22], such as classic Newton iteration method, alternately implicit iteration method, structure-preserving doubling algorithm and fixed point iteration method. In addition, Lu and Ma [17] proposed the linearized implicit iteration method for solving the algebraic Riccati equations. Benner and Kuerschner [5] presented low-rank Newton-ADI methods for solving large non-symmetric algebraic Riccati equations. Later, Guan [9] derived modified alternately linearized implicit iteration method for M-matrix algebraic Riccati equations. But there are very little results about the NCARE (1.1). In 2011, Luo [16] presented Newton iteration and fixed point iteration to solve the NCARE (1.1), where the two methods required solving a Sylvester equation at each step of the iterations. Recently, Zhang and Tan [23] proposed the INewton iteration method and the alternately linear implicit method for solving the NCARE (1.1), which avoided directly solving Sylvester equation. Motivated by above work, we propose a modified iterative algorithm to find the minimal non-negative solutions of the NCARE (1.1). Compared with some existing iterative algorithms, the modified iterative algorithm has better numerical effectiveness.

The rest of the paper is organized as follows. In Section 2, we present the MALI iteration algorithm to solve the NCARE (1.1). In Section 3, we show the convergence of the MALI iteration algorithm. In Section 4, we use numerical examples to show the feasibility and effectiveness of the modified iterative algorithm.

Throughout the paper, let $A = [a_{ij}] \in R^{m\times n}$ and $B = [b_{ij}] \in R^{m\times n}$. We write $A > 0$ ($A \geq 0$) if all $a_{ij} > 0$ ($a_{ij} \geq 0$) for all i, j. If $A > 0$ ($A \geq 0$), we say that A is a positive (non-negative) matrix. $A > B$ ($A \geq B$) means $A - B > 0$ ($A - B \geq 0$). A matrix $A \in R^{n\times n}$ is called a Z-matrix if its off-diagonal elements are non-positive. Any Z-matrix can be written as $A = sI - B$, where s is a positive constant and B a non-negative matrix. Z-matrix is called a non-singular M-matrix if $s > \rho(B)$ and a singular M-matrix if $s = \rho(B)$, where $\rho(B)$ is the spectral radius. A^T and $\|A\|$ denote the transpose and the spectral norm of matrix A, respectively.

The following are an assumption and some necessary lemmas.

Assumption 1.1. [23] *For the NCARE* (1.1) *we can find non-negative matrices* $Y_1, Y_2, \ldots, Y_s$ *such that* $R_i(Y_1, \ldots, Y_s) \leq 0$, *and*

$$K_i = \begin{pmatrix} D_i & -C_i \\ -B_i - \sum_{j\neq i} e_{ij} Y_j & A_i \end{pmatrix}$$

is a non-singular M-matrix or an irreducible singular M-matrix.

Lemma 1.2. [6] *For a Z-matrix $A \in R^{n\times n}$, the following statements are equivalent:*

(1) *A is a non-singular M-matrix;*

(2) *$A^{-1} \geq 0$;*

(3) *$Av > 0$ for some positive vector $v \in R^n$;*

(4) *All eigenvalues of A have positive real parts.*

Lemma 1.3. [18] *Let $A = (A_{ij}) \in R^{n\times n}$ be an M-matrix and $B = (b_{ij}) \in R^{n\times n}$ be a Z-matrix. If the element of B satisfies*

$$b_{ii} \geq a_{ii}, \quad a_{ij} \leq b_{ij} \leq 0, \quad i \neq j,\ 1 \leq i, j \leq n,$$

then B is also an M-matrix. Particularly, for any positive real θ, $B = \theta I + A$ is an M-matrix.

Lemma 1.4. [23] *If Assumption 1.1 is met, then $B_i \geq 0$, $C_i \geq 0$, and the NCARE (1.1) has a minimal non-negative solution $S = (S_1, \dots, S_s)$. Further, $D_i - C_iS_i$ and $A_i - S_iC_i$ are M-matrices.*

2. The MALI iteration method

In [23], Zhang presented an alternately linearized implicit (ALI) iteration method for NCARE (1.1) as follows.

ALI iteration scheme. Take a positive constant ζ_i such that

$$\zeta_i = \max\left\{\max_{1\leq j\leq m}[A_i]_{jj}, \max_{1\leq j\leq n}[D_i]_{jj}\right\},$$

then the iteration scheme is

$$\begin{aligned} X_i^{k+1/2}\big(\zeta_i I + (D_i - C_iX_i^k)\big) &= (\zeta_i I - A_i)X_i^k + B_i + \sum_{j\neq i} e_{ij}X_j^k, \\ \big(\zeta_i I + (A_i - X_i^{k+1/2}C_i)\big)X_i^{k+1} &= X_i^{k+1/2}(\zeta_i I - D_i) + B_i + \sum_{j\neq i} e_{ij}X_j^{k+1/2}. \end{aligned}$$

In this section, we propose a modified alternately linearized implicit method for the NCARE (1.1).

The MALI iteration scheme. Take a positive constant γ_i and β_i such that

$$\gamma_i = \max_{1\leq j\leq m}[A_i]_{jj}, \quad \beta_i = \max_{1\leq j\leq n}[D_i]_{jj}, \tag{2.1}$$

then the modified iteration scheme is

$$
\begin{aligned}
X_i^{k+1/2}(\gamma_i I + D_i) &= (\gamma_i I - A_i + X_i^k C_i)X_i^k + B_i \\
&\quad + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1/2} + (1-\omega)X_j^k\big) + \sum_{j=i+1}^{s} e_{ij} X_j^k, \\
(\beta_i I + A_i)X_i^{k+1} &= X_i^{k+1/2}(\beta_i I - D_i + C_i X_i^{k+1/2}) + B_i \\
&\quad + \sum_{j=i}^{i-1} e_{ij}\big(\omega X_j^{k+1} + (1-\omega)X_j^{k+1/2}\big) + \sum_{j=i+1}^{s} e_{ij} X_j^{k+1/2},
\end{aligned}
\tag{2.2}
$$

where $0 \le \omega \le 1$ is a given parameter.

Compared with ALI iteration method, the modified method is more efficient since the coefficient matrices of the modified iteration scheme (2.2) are fixed at each iteration step and different parameters γ_i and β_i are chosen based on different matrices D_i and A_i. Then less computational time is required for solving the NCARE (1.1). In addition, the modified method utilizes a weighted average of the estimates in both the last step and current step to update the estimates $X_i^{k+1/2}$ and X_i^{k+1}. Therefore, the convergence performance of the new method can be improved.

The algorithm is described as follows.

Algorithm 2.1.

Step 1: *Input matrices* A_i, B_i, C_i, D_i, $i = 1, 2, \ldots, s$, $E = (e_{ij})$ *and* $\omega > 0$.

Step 2: *Set* $X_i^0 = 0$, *a tolerance* ε_{out}, $k = 0$, *and compute*

$$
\mathrm{RES}_i^0 = \Big\| X_i^0 C_i X_i^0 - A_i X_i^0 - X_i^0 D_i + B_i + \sum_{j \ne i} e_{ij} X_j^0 \Big\|.
$$

Step 3: *Set* γ_i, β_i *as* (2.1).

Step 4: *Compute*

$$
\mathrm{RES}_i^k = \frac{\| X_i^k C_i X_i^k - A_i X_i^k - X_i^k D_i + B_i + \sum_{j \ne i} e_{ij} X_j^k \|}{\mathrm{RES}_i^0}.
$$

Step 5: *Stop if* $\max_{1 \le i \le s}(\mathrm{RES}_i^k) \le \varepsilon_{\text{out}}$. *Otherwise, go to Step* 6.

Step 6: *Compute*

$$
\begin{aligned}
X_i^{k+1/2} = &\Big((\gamma_i I - A_i + X_i^k C_i)X_i^k + B_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1/2} + (1-\omega)X_j^k\big) + \sum_{j=i+1}^{s} e_{ij} X_j^k \Big) \\
&\times (\gamma_i I + D_i)^{-1},
\end{aligned}
$$

$$X_i^{k+1} = (\beta_i I + A_i)^{-1}\Big(X_i^{k+1/2}(\beta_i I - D_i + C_i X_i^{k+1/2}) + B_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1} + (1-\omega)X_j^{k+1/2}\big) + \sum_{j=i+1}^{s} e_{ij} X_j^{k+1/2}\Big).$$

Step 7: *Set $k = k+1$ and go to Step* 4.

3. Convergence analysis

In this section, we analyze the convergence of Algorithm 2.1.

Theorem 3.1. *Let $S = (S_1, \ldots, S_s)$ be the minimal non-negative solution of the NCARE* (1.1)*. If Assumption* 1.1 *is met, then the matrix sequence $\{X_i^k\}$, $i = 1, 2, \ldots, s$, generated by Algorithm* 2.1*, satisfies*

(1)
$$(X_i^{k+1/2} - S_i)(\gamma_i I + D_i) = (\gamma_i I - A_i + X_i^k C_i)(X_i^k - S_i) + (X_i^k - S_i)C_i S_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega(X_j^{k+1/2} - S_i) + (1-\omega)(X_j^k - S_j)\big) + \sum_{j=i+1}^{s} e_{ij}(X_j^k - S_j);$$

(2)
$$(X_i^{k+1/2} - X_i^k)(\gamma_i I + D_i) = R_i(X_1^k, \ldots, X_i^k, \ldots, X_s^k) + \sum_{j=1}^{i-1} e_{ij}\omega(X_j^{k+1/2} - X_j^k);$$

(3)
$$R_i(X_1^{k+1/2}, \ldots, X_i^{k+1/2}, \ldots, X_s^{k+1/2}) = (\gamma_i I - A_i + X_i^{k+1/2} C_i)(X_i^{k+1/2} - X_i^k) + (X_i^{k+1/2} - X_i^k)C_i X_i^k + \sum_{j=1}^{i-1} e_{ij}(1-\omega)(X_j^{k+1/2} - X_j^k) + \sum_{j=i+1}^{s} e_{ij}(X_j^{k+1/2} - X_j^k);$$

(4)
$$(\beta_i I + A_i)(X_i^{k+1} - S_i) = (X_i^{k+1/2} - S_i)(\beta_i I - D_i + C_i X_i^{k+1/2}) + C_i S_i(X_i^{k+1/2} - S_i) + \sum_{j=1}^{i-1} e_{ij}\big(\omega(X_j^{k+1} - S_j) + (1-\omega)(X_j^{k+1/2} - S_j)\big) + \sum_{j=i+1}^{s} e_{ij}(X_j^{k+1/2} - S_j);$$

(5)
$$(\beta_i I + A_i)(X_i^{k+1} - X_i^{k+1/2}) = R_i(X_1^{k+1/2}, \ldots, X_i^{k+1/2}, \ldots, X_s^{k+1/2}) + \sum_{j=1}^{i-1} e_{ij}\omega(X_j^{k+1} - X_j^{k+1/2});$$

$$
\begin{aligned}
(6)\qquad & R_i(X_1^{k+1},\ldots,X_i^{k+1},\ldots,X_s^{k+1}) \\
&= (X_i^{k+1}-X_i^{k+1/2})(\beta_i I - D_i + C_i X_i^{k+1}) + X_i^{k+1/2}C_i(X_i^{k+1}-X_i^{k+1/2}) \\
&\quad + \sum_{j=1}^{i-1} e_{ij}(1-\omega)(X_j^{k+1}-X_j^{k+1/2}) + \sum_{j=i+1}^{s} e_{ij}(X_j^{k+1}-X_j^{k+1/2}).
\end{aligned}
$$

Proof. We prove (1)–(3), and omit (4)–(6) here since the proof process for (4)–(6) is similar to (1)–(3).

(1) From (2.2) and

$$
B_i - S_iD_i = A_iS_i - S_iC_iS_i - \sum_{j=1}^{i-1} e_{ij}\big(\omega S_j + (1-\omega)S_j\big) - \sum_{j=i+1}^{s} e_{ij}S_j,
$$

we get

$$
\begin{aligned}
&(X_i^{k+1/2}-S_i)(\gamma_i I + D_i) \\
&= (\gamma_i I - A_i + X_i^kC_i)X_i^k + B_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1/2} + (1-\omega)X_j^k\big) \\
&\quad + \sum_{j=i+1}^{s} e_{ij}X_j^k - S_i(\gamma_i I + D_i) \\
&= (\gamma_i I - A_i + X_i^kC_i)X_i^k + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1/2} + (1-\omega)X_j^k\big) + \sum_{j=i+1}^{s} e_{ij}X_j^k \\
&\quad + A_iS_i - S_iC_iS_i - \sum_{j=1}^{i-1} e_{ij}\big(\omega S_j + (1-\omega)S_j\big) - \sum_{j=i+1}^{s} e_{ij}S_j - \gamma_iS_i \\
&= (\gamma_i I - A_i)X_i^k - (\gamma_i I - A_i)S_i + X_i^kC_iX_i^k - X_i^kC_iS_i + X_i^kC_iS_i - S_iC_iS_i \\
&\quad + \sum_{j=1}^{i-1} e_{ij}\big(\omega(X_j^{k+1/2}-S_j) + (1-\omega)(X_j^k - S_j)\big) + \sum_{j=i+1}^{s} e_{ij}(X_j^k - S_j) \\
&= (\gamma_i I - A_i + X_i^kC_i)(X_i^k - S_i) + (X_i^k - S_i)C_iS_i \\
&\quad + \sum_{j=1}^{i-1} e_{ij}\big(\omega(X_j^{k+1/2}-S_i) + (1-\omega)(X_j^k - S_j)\big) + \sum_{j=i+1}^{s} e_{ij}(X_j^k - S_j).
\end{aligned}
$$

(2) Using (2.2) again, it is easy to verify that

$$
\begin{aligned}
&(X_i^{k+1/2}-X_i^k)(\gamma_i I + D_i) \\
&= (\gamma_i I - A_i + X_i^kC_i)X_i^k + B_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega X_j^{k+1/2} + (1-\omega)X_j^k\big)
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{j=i+1}^{s} e_{ij}X_j^k-\gamma_iX_i^k-X_i^kD_i\\
&=X_i^kC_iX_i^k-A_iX_i^k+B_i-X_i^kD_i+\sum_{j=1}^{i-1}e_{ij}\big(\omega X_j^{k+1/2}+(1-\omega)X_j^k\big)+\sum_{j=i+1}^{s}e_{ij}X_j^k\\
&=X_i^kC_iX_i^k-A_iX_i^k+B_i-X_i^kD_i+\sum_{j=1}^{i-1}e_{ij}X_j^k+\sum_{j=1}^{i-1}e_{ij}\omega(X_j^{k+1/2}-X_j^k)+\sum_{j=i+1}^{s}e_{ij}X_j^k\\
&=R_i(X_1^k,\dots,X_i^k,\dots,X_s^k)+\sum_{j=1}^{i-1}e_{ij}\omega(X_j^{k+1/2}-X_j^k).
\end{aligned}
$$

(3) From the first equation of (2.2), we have

$$
\begin{aligned}
&B_i-X_i^{k+1/2}D_i\\
&=\gamma_iX_i^{k+1/2}-(\gamma_iI-A_i+X_i^kC_i)X_i^k-\sum_{j=1}^{i-1}e_{ij}\big(\omega X_j^{k+1/2}+(1-\omega)X_j^k\big)-\sum_{j=i+1}^{s}e_{ij}X_j^k.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&R_i(X_1^{k+1/2},\dots,X_i^{k+1/2},\dots,X_s^{k+1/2})\\
&=X_i^{k+1/2}C_iX_i^{k+1/2}-A_iX_i^{k+1/2}-X_i^{k+1/2}D_i+B_i\\
&\quad+\sum_{j=1}^{i-1}e_{ij}\big(\omega X_j^{k+1/2}+(1-\omega)X_j^{k+1/2}\big)+\sum_{j=i+1}^{s}e_{ij}X_j^{k+1/2}\\
&=X_i^{k+1/2}C_iX_i^{k+1/2}-A_iX_i^{k+1/2}+\gamma_iX_i^{k+1/2}-(\gamma_iI-A_i+X_i^kC_i)X_i^k\\
&\quad-\sum_{j=1}^{i-1}e_{ij}\big(\omega X_j^{k+1/2}+(1-\omega)X_j^k\big)-\sum_{j=i+1}^{s}e_{ij}X_j^k\\
&\quad+\sum_{j=1}^{i-1}e_{ij}\big(\omega X_j^{k+1/2}+(1-\omega)X_j^{k+1/2}\big)+\sum_{j=i+1}^{s}e_{ij}X_j^{k+1/2}\\
&=X_i^{k+1/2}C_iX_i^{k+1/2}+(\gamma_iI-A_i)(X_i^{k+1/2}-X_i^k)-X_i^kC_iX_i^k\\
&\quad+\sum_{j=1}^{i-1}e_{ij}(1-\omega)(X_j^{k+1/2}-X_j^k)+\sum_{j=i+1}^{s}e_{ij}(X_j^{k+1/2}-X_j^k)\\
&=(\gamma_iI-A_i)(X_i^{k+1/2}-X_i^k)+X_i^{k+1/2}C_iX_i^{k+1/2}-X_i^{k+1/2}C_iX_i^k+X_i^{k+1/2}C_iX_i^k\\
&\quad-X_i^kC_iX_i^k+\sum_{j=1}^{i-1}e_{ij}(1-\omega)(X_j^{k+1/2}-X_j^k)+\sum_{j=i+1}^{s}e_{ij}(X_j^{k+1/2}-X_j^k)\\
&=(\gamma_iI-A_i+X_i^{k+1/2}C_i)(X_i^{k+1/2}-X_i^k)+(X_i^{k+1/2}-X_i^k)C_iX_i^k\\
&\quad+\sum_{j=1}^{i-1}e_{ij}(1-\omega)(X_j^{k+1/2}-X_j^k)+\sum_{j=i+1}^{s}e_{ij}(X_j^{k+1/2}-X_j^k).
\end{aligned}
$$

Therefore, we have proven the conclusions (1)–(3). □

Theorem 3.2. *Let $S = (S_1, \ldots, S_s)$ be the minimal non-negative solution of the NCARE* (1.1)*. If Assumption* 1.1 *is met, then the matrix sequence $\{X_i^k\}$, $i = 1, 2, \ldots, s$, generated by Algorithm* 2.1 *is well defined and is monotonically increasing and bounded,*

$$R_i(X_1^{k+1}, \ldots, X_i^{k+1}, \ldots, X_s^{k+1}) \geq 0, \quad 0 \leq X_i^0 \leq \cdots \leq X_i^k \leq X_i^{k+1} \leq \cdots \leq S_i. \tag{3.1}$$

Moreover, $\{X_i^k\}$ is convergent to the minimal non-negative solution S_i of the NCARE (1.1)*.*

Proof. (1) Because K_i is a non-singular M-matrix or an irreducible singular M-matrix, then $B_i \geq 0$, $C_i \geq 0$ and $K_i \leq \operatorname{diag}(D_i, A_i)$, thus A_i and D_i are M-matrices by Lemma 1.3. Therefore,

$$\gamma_i I - A_i \geq 0, \quad \beta_i I - D_i \geq 0, \quad i = 1, 2, \ldots, s, \tag{3.2}$$

where γ_i, β_i are from (2.1), and the matrices $\beta_i I + A_i$ and $\gamma_i I + D_i$ are also M-matrices according to Lemma 1.3, then we get

$$(\gamma_i I + D_i)^{-1} \geq 0, \quad (\beta_i I + A_i)^{-1} \geq 0, \quad i = 1, 2, \ldots, s \tag{3.3}$$

with Lemma 1.2.

Next we demonstrate (3.1) by mathematical induction.

(i) When $k = 0$, we will prove (3.1) is true.

(a) Let's prove $X_i^{1/2} \leq S_i$, $i = 1, 2, \ldots, s$. Putting the initial matrix $X_i^0 = 0$ into the Theorem 3.1(1), we get

$$\begin{aligned} &X_i^{1/2} - S_i \\ &= \Big(-(\gamma_i I - A_i)S_i - S_i C_i S_i + \sum_{j=1}^{i-1} e_{ij}\big(\omega(X_j^{1/2} - S_i) + (1-\omega)(-S_j)\big) - \sum_{j=i+1}^{s} e_{ij} S_j \Big) \\ &\quad \times (\gamma_i I + D_i)^{-1}. \end{aligned} \tag{3.4}$$

Considering the above equation (3.4) with $i = 1$, we have

$$X_1^{1/2} - S_1 = \Big(-(\gamma_1 I - A_1)S_1 - S_1 C_1 S_1 - \sum_{j=2}^{s} e_{1j} S_j \Big)(\gamma_1 I + D_1)^{-1}. \tag{3.5}$$

Thus from (3.5), (3.3) and (3.2), it follows that $X_1^{1/2} - S_1 \leq 0$.

Assume that $X_i^{1/2} - S_i \le 0$, $i \le l$. Considering the equation (3.4) with $i = l+1$, we get

$$X_{l+1}^{1/2} - S_{l+1} = -\Big((\gamma_{l+1} I - A_{l+1}) S_{l+1} + \sum_{j=1}^{l} e_{l+1,j}\big(\omega(S_j - X_j^{1/2}) + (1-\omega) S_j\big) + S_{l+1} C_{l+1} S_{l+1} + \sum_{j=l+2}^{s} e_{l+1,j} S_j\Big)(\gamma_{l+1} I + D_{l+1})^{-1}.$$

By (3.2), (3.3) and induction assumption $X_i^{1/2} - S_i \le 0$, $i \le l$, we get $X_{l+1}^{1/2} - S_{l+1} \le 0$. Thus we have

$$X_i^{1/2} \le S_i, \quad i = 1, 2, \ldots, s \tag{3.6}$$

by principle of mathematical induction.

(b) Let's prove $X_i^{1/2} \ge 0$, $i = 1, 2, \ldots, s$. Putting the initial matrix $X_i^0 = 0$ into the iteration format (2.2), we get

$$X_i^{1/2}(\gamma_i I + D_i) = B_i + \sum_{j=1}^{i-1} \omega e_{ij} X_j^{1/2}. \tag{3.7}$$

Considering the equation (3.7) with $i = 1$, we have $X_1^{1/2}(\gamma_1 I + D_1) = B_1$. Hence, by (3.3) we get $X_1^{1/2} = B_1(\gamma_1 I + D_1)^{-1} \ge 0$.

Assumed that $X_i^{1/2} \ge 0$, $i \le l$. Considering the equation (3.7) with $i = l+1$, we have

$$X_{l+1}^{1/2} = \Big(B_{l+1} + \sum_{j=1}^{l} \omega e_{l+1,j} X_j^{1/2}\Big)(\gamma_{l+1} I + D_{l+1})^{-1}.$$

By (3.3), we get $X_{l+1}^{1/2} \ge 0$. Hence, it has shown that $X_i^{1/2} \ge 0$ for all i.

(c) Let's prove $X_i^1 \ge X_i^{1/2}$, $i = 1, 2, \ldots, s$. Using the conclusion (3) of Theorem 3.1, we can easily get

$$\begin{aligned} &R_i(X_1^{1/2}, \ldots, X_i^{1/2}, \ldots, X_s^{1/2}) \\ &= (\gamma_i I - A_i + X_i^{1/2} C_i) X_i^{1/2} + \sum_{j=1}^{i-1} e_{ij}(1-\omega) X_j^{1/2} + \sum_{j=i+1}^{s} e_{ij} X_j^{1/2} \ge 0 \end{aligned} \tag{3.8}$$

with (3.2). And from the conclusion (5) of Theorem 3.1, we have

$$X_i^1 - X_i^{1/2} = (\beta_i I + A_i)^{-1}\Big(R_i(X_1^{1/2}, \ldots, X_i^{1/2}, \ldots, X_s^{1/2}) + \sum_{j=1}^{i-1} e_{ij}\omega(X_j^1 - X_j^{1/2})\Big). \tag{3.9}$$

Then considering the equation (3.9) with $i = 1$, according to (3.3) and (3.8), we get

$$X_1^1 - X_1^{1/2} = (\beta_1 I + A_1)^{-1} R_1(X_1^{1/2}, \ldots, X_i^{1/2}, \ldots, X_s^{1/2}) \ge 0.$$

Assume that $X_i^1 - X_i^{1/2} \geq 0$, $i \leq l$. Considering the equation (3.9) with $i = l+1$, in light of (3.3), (3.8) and the induction assumption $X_i^1 - X_i^{1/2} \geq 0$, $i \leq l$, we get

$$\begin{aligned}
&X_{l+1}^1 - X_{l+1}^{1/2} \\
&= (\beta_{l+1} I + A_{l+1})^{-1} \left(R_{l+1}(X_1^{1/2}, \ldots, X_i^{1/2}, \ldots, X_s^{1/2}) + \sum_{j=1}^{l} e_{l+1,j} \omega (X_j^1 - X_j^{1/2}) \right) \geq 0.
\end{aligned}$$

Thus, by induction we have

$$X_i^1 \geq X_i^{1/2}, \quad i = 1, 2, \ldots, s. \tag{3.10}$$

(d) Let's prove $X_i^1 \leq S_i$, $i = 1, 2, \ldots, s$. Utilizing the conclusion (4) of Theorem 3.1, we get

$$\begin{aligned}
&(\beta_i I + A_i)(X_i^1 - S_i) \\
&= (X_i^{1/2} - S_i)(\beta_i I - D_i + C_i X_i^{1/2}) + C_i S_i (X_i^{1/2} - S_i) \\
&\quad + \sum_{j=1}^{i-1} e_{ij} \big(\omega (X_j^1 - S_j) + (1-\omega)(X_j^{1/2} - S_j) \big) + \sum_{j=i+1}^{s} e_{ij} (X_j^{1/2} - S_j).
\end{aligned} \tag{3.11}$$

Considering the equation (3.11) with $i = 1$, according to (3.3), (3.2) and (3.6), we have

$$\begin{aligned}
X_1^1 - S_1 = (\beta_1 I + A_1)^{-1} \Big(&(X_1^{1/2} - S_1)(\beta_1 I - D_1 + C_1 X_1^{1/2}) \\
&+ C_1 S_1 (X_1^{1/2} - S_1) + \sum_{j=2}^{s} e_{1j} (X_j^{1/2} - S_j) \Big) \leq 0,
\end{aligned}$$

that is, $X_1^1 \leq S_1$.

Assume that $X_i^1 \leq S_i$, $i \leq l$. For the equation (3.11) with $i = l+1$, according to (3.3), (3.2), (3.6) and the induction assumption $X_i^1 \leq S_i$, $i \leq l$, we have

$$\begin{aligned}
&X_{l+1}^1 - S_{l+1} \\
&= (\beta_{l+1} I + A_{l+1})^{-1} \\
&\quad \times \Big((X_{l+1}^{1/2} - S_{l+1})(\beta_{l+1} I - D_{l+1} + C_{l+1} X_{l+1}^{1/2}) + \sum_{j=l+2}^{s} e_{l+1,j} (X_j^{1/2} - S_j) \\
&\quad + \sum_{j=1}^{l} e_{l+1,j} \big(\omega (X_j^1 - S_j) + (1-\omega)(X_j^{1/2} - S_j) \big) + C_{l+1} S_{l+1} (X_{l+1}^{1/2} - S_{l+1}) \Big) \leq 0.
\end{aligned}$$

Thus, by induction, we know that $X_i^1 \leq S_i$ holds for all i.

Moreover, for the conclusion (6) of Theorem 3.1 with $k = 0$, by (3.10) and (3.2) we get

$$R_i(X_1^1, \ldots, X_i^1, \ldots, X_s^1) = (X_i^1 - X_i^{1/2})(\beta_i I - D_i + C_i X_i^1) + X_i^{1/2} C_i (X_i^1 - X_i^{1/2}) + \sum_{j=1}^{i-1} e_{ij}(1-\omega)(X_j^1 - X_j^{1/2}) + \sum_{j=i+1}^{s} e_{ij}(X_j^1 - X_j^{1/2}) \geq 0.$$

By now, we have proven

$$0 \leq X_i^0 \leq X_i^1 \leq S_i, \quad R_i(X_1^1, \ldots, X_i^1, \ldots, X_s^1) \geq 0, \quad i = 1, 2, \ldots, s.$$

(ii) Assume that (3.1) is true for $k \geq 1$, i.e.,

$$0 \leq X_i^{k-1} \leq X_i^k \leq S_i, \quad R_i(X_1^k, \ldots, X_i^k, \ldots, X_s^k) \geq 0, \quad i = 1, 2, \ldots, s. \tag{3.12}$$

(iii) Next we will prove (3.1) is true for $k + 1$.

(a') Using conclusion (1) of Theorem 3.1 with $i = 1$, by (3.2), (3.12) and (3.3), we get

$$\begin{aligned} X_1^{k+1/2} - S_1 &= \Big((\gamma_1 I - A_1 + X_1^k C_1)(X_1^k - S_1) + (X_1^k - S_1) C_1 S_1 + \sum_{j=2}^{s} e_{ij}(X_j^k - S_j) \Big) \\ &\quad \times (\gamma_1 I + D_1)^{-1} \\ &\leq 0, \end{aligned}$$

that is, $X_1^{k+1/2} \leq S_1$.

Assume that $X_i^{k+1/2} \leq S_i$, $i \leq l$. Using conclusion (1) of Theorem 3.1 with $i = l + 1$, by (3.2), (3.12), (3.3) and the induction assumption $X_i^{k+1/2} \leq S_i$, $i \leq l$, we have

$$\begin{aligned} X_{l+1}^{k+1/2} - S_{l+1} = \Big(&(\gamma_{l+1} I - A_{l+1} + X_{l+1}^k C_{l+1})(X_{l+1}^k - S_{l+1}) \\ &+ (X_{l+1}^k - S_{l+1}) C_{l+1} S_{l+1} + \sum_{j=l+2}^{s} e_{l+1,j}(X_j^k - S_j) \\ &+ \sum_{j=1}^{l} e_{l+1,j}\big(\omega(X_j^{k+1/2} - S_j) + (1-\omega)(X_j^k - S_j)\big) \Big) (\gamma_{l+1} I + D_{l+1})^{-1} \leq 0. \end{aligned}$$

Therefore,

$$X_i^{k+1/2} \leq S_i, \quad i = 1, 2, \ldots, s. \tag{3.13}$$

(b') Utilizing the conclusion (2) of Theorem 3.1, we get

$$X_i^{k+1/2} - X_i^k = \Big(R_i(X_1^k, \ldots, X_i^k, \ldots, X_s^k) + \sum_{j=1}^{i-1} e_{ij}\omega(X_j^{k+1/2} - X_j^k) \Big) (\gamma_i I + D_i)^{-1}. \tag{3.14}$$

Considering (3.14) with $i = 1$, according to (3.12) and (3.3), we have

$$X_1^{k+1/2} - X_1^k = R_1(X_1^k, \dots, X_i^k, \dots, X_s^k)(\gamma_1 I + D_1)^{-1} \geq 0,$$

that is, $X_1^{k+1/2} \geq X_1^k$.

Assume that $X_i^{k+1/2} \geq X_i^k$, $i \leq l$. Considering (3.14) with $i = l+1$, by (3.12), (3.3) and the assumption $X_i^{k+1/2} \geq X_i^k$, $i \leq l$, we get

$$\begin{aligned} &X_{l+1}^{k+1/2} - X_{l+1}^k \\ &= \left(R_{l+1}(X_1^k, \dots, X_i^k, \dots, X_s^k) + \sum_{j=1}^{l} e_{l+1,j}\omega(X_j^{k+1/2} - X_j^k) \right)(\gamma_{l+1} I + D_{l+1})^{-1} \geq 0. \end{aligned}$$

Therefore, by induction,

$$X_i^{k+1/2} \geq X_i^k, \quad i = 1, 2, \dots, s. \tag{3.15}$$

(c') For the conclusion (3) of Theorem 3.1, by (3.2) and (3.15), we get

$$\begin{aligned} &R_i(X_1^{k+1/2}, \dots, X_i^{k+1/2}, \dots, X_s^{k+1/2}) \\ &= (\gamma_i I - A_i + X_i^{k+1/2} C_i)(X_i^{k+1/2} - X_i^k) + (X_i^{k+1/2} - X_i^k) C_i X_i^k \\ &\quad + \sum_{j=1}^{i-1} e_{ij}(1-\omega)(X_j^{k+1/2} - X_j^k) + \sum_{j=i+1}^{s} e_{ij}(X_j^{k+1/2} - X_j^k) \geq 0. \end{aligned} \tag{3.16}$$

And from conclusion (5) of Theorem 3.1, we have

$$\begin{aligned} &X_i^{k+1} - X_i^{k+1/2} \\ &= (\beta_i I + A_i)^{-1} \left(R_i(X_1^{k+1/2}, \dots, X_i^{k+1/2}, \dots, X_s^{k+1/2}) + \sum_{j=1}^{i-1} e_{ij}\omega(X_j^{k+1} - X_j^{k+1/2}) \right). \end{aligned} \tag{3.17}$$

Considering the equation (3.17) with $i = 1$, by (3.3) and (3.16), we get

$$X_1^{k+1} - X_1^{k+1/2} = (\beta_1 I + A_1)^{-1} R_1(X_1^{k+1/2}, \dots, X_i^{k+1/2}, \dots, X_s^{k+1/2}) \geq 0.$$

Assume that $X_i^{k+1} \geq X_i^{k+1/2}$, $i \leq l$. Considering the equation (3.17) with $i = l+1$, by (3.3), (3.16) and the assumption $X_i^{k+1} \geq X_i^{k+1/2}$, $i \leq l$, we can easily get

$$\begin{aligned} &X_{l+1}^{k+1} - X_{l+1}^{k+1/2} \\ &= (\beta_{l+1} I + A_{l+1})^{-1} \\ &\quad \times \left(R_{l+1}(X_1^{k+1/2}, \dots, X_i^{k+1/2}, \dots, X_s^{k+1/2}) + \sum_{j=1}^{l} e_{l+1,j}\omega(X_j^{k+1} - X_j^{k+1/2}) \right) \geq 0. \end{aligned}$$

By induction,

$$X_i^{k+1} \geq X_i^{k+1/2}, \quad i = 1, 2, \ldots, s. \tag{3.18}$$

(d') By making use of the conclusion (4) of Theorem 3.1 with $i = 1$, by (3.3), (3.2) and (3.13), we know that

$$\begin{aligned} X_1^{k+1} - S_1 = (\beta_1 I + A_1)^{-1}\Big(& (X_1^{k+1/2} - S_1)(\beta_1 I - D_1 + C_1 X_1^{k+1/2}) \\ & + C_1 S_1 (X_1^{k+1/2} - S_1) + \sum_{j=2}^{s} e_{1j}(X_j^{k+1/2} - S_j)\Big) \leq 0, \end{aligned}$$

that is, $X_1^{k+1} \leq S_1$.

Assume that $X_i^{k+1} \leq S_i$, $i \leq l$. By making use of the conclusion (4) of Theorem 3.1 with $i = l+1$, by (3.3), (3.2), (3.13) and the assumption $X_i^{k+1} \leq S_i$, $i \leq l$, we can easily get

$$\begin{aligned} X_{l+1}^{k+1} - S_{l+1} = (\beta_{l+1} I + A_{l+1})^{-1}\Big(& (X_{l+1}^{k+1/2} - S_{l+1})(\beta_{l+1} I - D_{l+1} + C_{l+1} X_{l+1}^{k+1/2}) \\ & + \sum_{j=l+2}^{s} e_{l+1,j}(X_j^{k+1/2} - S_j) \\ & + \sum_{j=1}^{l} e_{l+1,j}\big(\omega(X_j^{k+1} - S_j) + (1-\omega)(X_j^{k+1/2} - S_j)\big) \\ & + C_{l+1} S_{l+1}(X_{l+1}^{k+1/2} - S_{l+1})\Big) \leq 0. \end{aligned}$$

Therefore, it holds that $X_i^{k+1} \leq S_i$ for all i by induction.

Moreover, for the conclusion (6) of Theorem 3.1, by (3.2) and (3.18), we have

$$\begin{aligned} & R_i(X_1^{k+1}, \ldots, X_i^{k+1}, \ldots, X_s^{k+1}) \\ &= (X_i^{k+1} - X_i^{k+1/2})(\beta_i I - D_i + C_i X_i^{k+1}) + X_i^{k+1/2} C_i (X_i^{k+1} - X_i^{k+1/2}) \\ &\quad + \sum_{j=1}^{i-1} e_{ij}(1-\omega)(X_j^{k+1} - X_j^{k+1/2}) + \sum_{j=i+1}^{s} e_{ij}(X_j^{k+1} - X_j^{k+1/2}) \geq 0. \end{aligned}$$

Thus, the proof of (3.1) is completed.

(2) From the above proof, we find that the matrix sequence $\{X_i^k\}$ is non-negative, monotonically increasing and bounded, so there must exist a non-negative matrix S_i^* such that $\lim_{k\to\infty} X_i^k = S_i^*$. And it also holds that $\lim_{k\to\infty} X_i^{k+1/2} = S_i^*$. Obviously, $X_i^k \leq S_i$ implies $S_i^* \leq S_i$. On the other hand, by taking limits on both sides of (2.2), we get

$$S_i^* C_i S_i^* - S_i^* D_i - A_i S_i^* + B_i + \sum_{j \neq i} e_{ij} S_j^* = 0.$$

Hence, S_i^* is also a non-negative solution of NCARE (1.1). And it holds that $S_i \leq S_i^*$ because S_i is a minimal non-negative solution of the NCARE (1.1) by Lemma 1.4. Therefore, $S_i^* = S_i$. $\square$

4. Numerical examples

In this section, we use the following examples to show the feasibility and effectiveness of the modified methods for solving the minimal non-negative solution of the NCARE (1.1). We compare the MALI method with the ALI method and the INewton method about the iteration steps (IT), the computing times (CPU) and the norm of solution errors (RES). RES is defined by

$$\mathrm{RES}_i = \frac{\left\|R_i(X_1^k, X_2^k, \ldots, X_s^k)\right\|_\infty}{\left\|R_i(X_1^0, X_2^0, \ldots, X_s^0)\right\|_\infty}, \quad i = 1, 2, \ldots, s.$$

Example 4.1. Consider the NCARE (1.1) with

$$A_1 = \begin{pmatrix} 6.7 & -1.4 & -3 \\ -3.3 & 4 & -1 \\ -1 & -2 & 6 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 5 & -3.2 & -3.5 \\ -2.2 & 3 & -3 \\ -2.7 & -3.8 & 4 \end{pmatrix},$$

$$D_1 = \begin{pmatrix} 371 & -2.8 \\ 0 & 389 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 376 & -1.9 \\ -0.5 & 375 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 11 & 10 \\ 0.5 & 13 \\ 1 & 12 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1.5 & 1 \\ 1 & 2.3 \\ 1 & 1 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 1.5 & 0 & 3 \\ 2 & 0.2 & 2.8 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 2.4 & 2 & 2.2 \\ 3 & 0 & 1.4 \end{pmatrix}, \quad [e_{ij}] = \begin{pmatrix} 0.3 & 0.3 \\ 0.3 & 0.3 \end{pmatrix}.$$

We show the convergent performance of the ALI method, the INewton method and the MALI method. Obviously, from Figure 4.1 and Table 4.1, we see that MALI method is more efficient than the other two methods, and it only needs 4 steps and 0.0047s to converge to the iteration solution.

Table 4.1: Example 4.1 ($\omega = 0.3$).

Method	ALI	INewton	MALI
IT	8	5	4
CPU	0.0056	0.0107	0.0047
$\mathrm{RES}_{\max}$	4.8588e-14	4.8213e-14	6.0970e-14

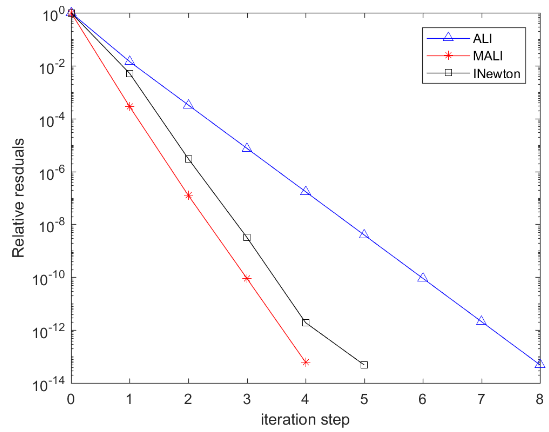

Figure 4.1: Relative residual for Example 4.1.

Example 4.2. [23] Consider the NCARE (1.1) with

$$A_i = \begin{pmatrix} i & -1 & & \\ & i & \ddots & \\ & & \ddots & -1 \\ & & & i \end{pmatrix} \in R^{n\times n}, \quad D_i = \begin{pmatrix} 2i & -1 & & \\ & 2i & \ddots & \\ & & \ddots & -1 \\ & & & 2i \end{pmatrix} \in R^{n\times n},$$

$$B_i = 0.5I_n, \quad C_i = 0.2I_n, \quad E = \text{rand}(s), \quad i = 1, 2, \ldots, s.$$

From Table 4.2, we see that the MALI method is convergent to the minimal non-negative solution S of the NCARE (1.1) under the required precision when the parameter ω is given differently. Especially when $\omega = 1.4$, it only needs 15 steps and 0.0298s to converge to the iteration solution. And the new method we presented works well in practical computation when $\omega > 1$, although we only proved its convergence when $0 \leq \omega \leq 1$.

Table 4.2: Example 4.2 with $n = 6$, $s = 18$.

ω	IT	CPU	$\text{RES}_{\max}$	ω	IT	CPU	$\text{RES}_{\max}$
0.7	18	0.0359	7.8720e-13	1.3	17	0.0257	6.5866e-13
0.8	18	0.0284	6.7561e-13	1.4	15	0.0298	9.2094e-13
0.9	23	0.0375	1.4590e-13	1.5	20	0.0393	3.9634e-13
1.0	18	0.0344	5.7115e-13	1.6	18	0.0406	1.3460e-13
1.1	16	0.0348	4.1879e-13	1.7	22	0.0422	9.1307e-13
1.2	18	0.0358	8.4378e-13	1.8	23	0.0453	1.5437e-13

Set $s = 6, 8, 10$; $n = 18$ and $\omega = 1.3$. The iteration steps, CPU times and RES residuals for the three methods are listed in Table 4.3. From Table 4.3, it can be seen that the iteration steps and computational time of the MALI method are least among all these methods. Therefore, with respect to the computing efficiency, the MALI method outperforms the ALI and the INewton methods in [23]. And Figure 4.2 shows the relative residual of Example 4.2 when $s = 10$, $n = 18$.

Table 4.3: Example 4.2 with $n = 18$.

Method		ALI	INewton	MALI
	IT	25	31	14
$s = 6$	CPU	0.0378	0.4987	0.0295
	$\mathrm{RES}_{\max}$	5.4755e-11	4.9676e-11	4.5170e-11
	IT	31	41	17
$s = 8$	CPU	0.0594	0.8202	0.0452
	$\mathrm{RES}_{\max}$	7.0171e-11	8.1980e-11	1.6864e-11
	IT	37	52	18
$s = 10$	CPU	0.0795	1.4928	0.0502
	$\mathrm{RES}_{\max}$	7.0224e-11	7.4733e-11	2.0468e-11

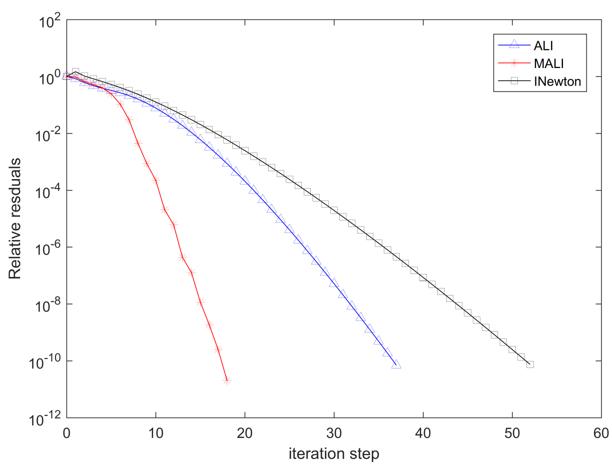

Figure 4.2: Relative residual for Example 4.2, $s = 10$, $n = 18$.

Example 4.3. [23] In this example, we consider the NCARE (1.1) with

$$A_i = \mathrm{tridiag}(-2iI_m, R_i, -2iI_m) \in R^{n\times n}, \quad B_i = \frac{1}{50}\mathrm{tridiag}(1,2,1) \in R^{n\times n},$$
$$C_i = \xi B_i, \quad D_i = \mathrm{tridiag}(-2iI_m, T_i, -2iI_m) \in R^{n\times n}, \quad \text{and} \quad E = \mathrm{rand}(s),$$

where $\xi > 0$ is a given constant, $i = 1, 2, \ldots, s$, $n = m^2$, and

$$R_i = \text{tridiag}\left(-1, 4i + \frac{200}{(m+1)^2}, -1\right) \in R^{m\times m},$$

$$T_i = \text{tridiag}\left(-1, 14i + \frac{200}{(m+1)^2}, -1\right) \in R^{m\times m}.$$

Set $s = 12$, $m = 5$ and $\xi = 0.2$. The iteration steps, CPU times and RES residuals for the three methods are listed in Table 4.4. From Table 4.4, it can be observed that three iteration methods can converge to the minimal non-negative solution of the NCARE (1.1). The computational time of the MALI method is smaller than the ALI and the INewton methods if the parameter ω is chosen randomly in the range 0 to 1. And the iteration steps of the MALI method is smaller than the ALI method. Hence, the MALI method that we proposed outperforms the methods in [23] with respect to the computing efficiency. Figure 4.3 shows the relative residual of Example 4.3.

Table 4.4: Example 4.3 with $s = 12$, $m = 5$, $\xi = 0.2$.

Method	ALI	INewton	MALI
IT	9	7	7
CPU	0.0612	0.4076	0.0536
$\text{RES}_{\max}$	2.8477e-7	2.7532e-7	1.4686e-7

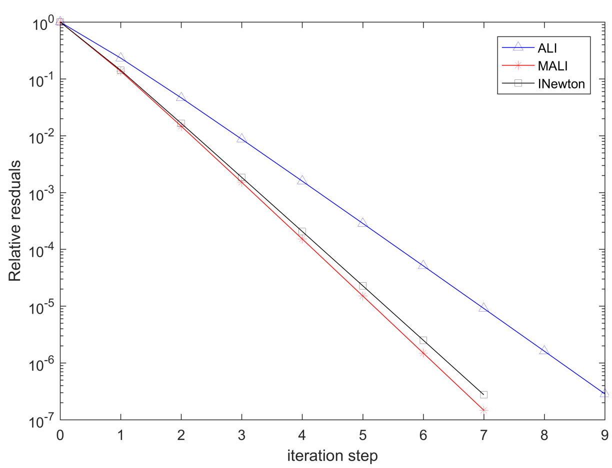

Figure 4.3: Relative residual for Example 4.3 ($\xi = 0.2$).

Acknowledgments

The authors would like to thank Yuli Zhu for providing an example of data and improving the English expression in this paper. The work was supported in part by National Natural Science Foundation for Youths of China (11801164) and the Youth Project of Hunan Provincial Education Department of China (22B0498), National Natural Science Foundation of China (11571292) and the Key Project of National Natural Science Foundation of China (91430213).

References

[1] H. Abou-Kandil, G. Freiling and G. Jank, *On the solution of discrete-time Markovian jump linear quadratic control problems*, Automatica J. IFAC **31** (1995), no. 5, 765–768.

[2] Z.-Z. Bai, Y.-H. Gao and L.-Z. Lu, *Fast iterative schemes for nonsymmetric algebraic Riccati equations arising from transport theory*, SIAM J. Sci. Comput. **30** (2008), no. 2, 804–818.

[3] 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), no. 8, 655–674.

[4] R. Bellman and G. M. Wing, *An Introduction to Invariant Imbedding*, Classics Appl. Math. **8**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[5] P. Benner, P. Kürschner and J. Saak, *Low-rank Newton-ADI methods for large nonsymmetric algebraic Riccati equations*, J. Franklin Inst. **353** (2016), no. 5, 1147–1167.

[6] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. **9**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[7] B. N. Datta, *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, San Diego, CA, 2004.

[8] Y.-H. Gao and Z.-Z. Bai, *On inexact Newton methods based on doubling iteration scheme for non-symmetric algebraic Riccati equations*, Numer. Linear Algebra Appl. **18** (2011), no. 3, 325–341.

[9] J. Guan, *Modified alternately linearized implicit iteration method for M-matrix algebraic Riccati equations*, Appl. Math. Comput. **347** (2019), 442–448.

[10] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl. **22** (2000), no. 2, 376–391.

[11] X.-X. Guo, W.-W. Lin and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math. **103** (2006), no. 3, 393–412.

[12] C. Han, H. Li, H. Zhang and M. Fu, *Optimal control and stabilization for discrete-time Markov jump linear systems with input delay*, SIAM J. Control Optim. **59** (2021), no. 5, 3524–3551.

[13] B. Huang and C. Ma, *Some accelerated iterative algorithms for solving nonsymmetric algebraic Riccati equations arising in transport theory*, Int. J. Comput. Math. **97** (2020), no. 9, 1819–1839.

[14] X. Liang, J. Xu and H. Zhang, *Solution to stochastic LQ control problem for Itô systems with state delay or input delay*, Systems Control Lett. **113** (2018), 86–92.

[15] J. Liu, L. Wang and Y. Bai, *New estimates of upper bounds for the solutions of the continuous algebraic Riccati equation and the redundant control inputs problems*, Automatica J. IFAC **116** (2020), 108936, 6 pp.

[16] J. Liu, J. Zhang and F. Luo, *Newton's method for the positive solution of the coupled algebraic Riccati equation applied to automatic control*, Comput. Appl. Math. **39** (2020), no. 2, Paper No. 113, 17 pp.

[17] H. Lu and C. Ma, *A new linearized implicit iteration method for nonsymmetric algebraic Riccati equations*, J. Appl. Math. Comput. **50** (2016), no. 1-2, 227–241.

[18] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. **30**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[19] L. C. G. Rogers, *Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains*, Ann. Appl. Probab. **4** (1994), no. 2, 390–413.

[20] L. Wang and J. Liu, *Upper solution bound of the CCARE and its application in multi-agent systems with time-delay in the state*, Internat. J. Control **96** (2023), no. 7, 1754–1764.

[21] L. Wang, L. Zhou and L. Zhang, *The new solution bounds of the discrete algebraic Riccati equation and their applications in redundant control inputs systems*, Asian J. Control **25** (2023), no. 5, 4134–4144.

[22] A.-G. Wu, H.-J. Sun and Y. Zhang, *A novel iterative algorithm for solving coupled Riccati equations*, Appl. Math. Comput. **364** (2020), 124645, 14 pp.

[23] J. Zhang and F. Tan, *Numerical methods for the minimal non-negative solution of the non-symmetric coupled algebraic Riccati equation*, Asian J. Control **23** (2021), no. 1, 374–386.

Li Wang and Yibo Wang
School of Mathematics and Computational Science, Hunan University of Science and Technology, Hunan 411201, China
E-mail addresses: wanglileigh@163.com, 1183097040@qq.com