Journal of Applied Analysis and Computation
Volume 9, Number 2, April 2019, 526–546

Website:http://jaac.ijournal.cn
DOI:10.11948/2156-907X.20170324

SOME ITERATIVE ALGORITHMS FOR POSITIVE DEFINITE SOLUTION TO NONLINEAR MATRIX EQUATIONS*

Baohua Huang[1]and Changfeng Ma[1,†]

Abstract This paper is concerned with the unique positive definite solution to a system of nonlinear matrix equations $X - A^*\bar{Y}^{-1}A = I_n$ and $Y - B^*\bar{X}^{-1}B = I_n$, where $A, B \in \mathbb{C}^{n\times n}$ are given matrices. Based on the special structure of the system of nonlinear matrix equations, the system can be equivalently reformulated as $V - C^*\bar{V}^{-1}C = I_{2n}$. Moreover, by means of Sherman-Moorison-Woodbury formula, we derive the relationship between the solutions of $V - C^*\bar{V}^{-1}C = I_{2n}$ and the well studied standard nonlinear matrix equation $Z + D^*Z^{-1}D = Q$, where D, Q are uniquely determined by C. Then, we present a structure-preserving doubling algorithm and two modified structure-preserving doubling algorithms to compute the positive definite solution of the system. Furthermore, cyclic reduction algorithm and two modified cyclic reduction algorithms for the positive definite solution of the system are proposed. Finally, some numerical examples are presented to illustrate the efficiency of the theoretical results and the behavior of the considered algorithms.

Keywords Nonlinear matrix equation, structure-preserving algorithm, cyclic reduction algorithm, positive definite solution, convergence theory.

MSC(2010) 65F10, 65F30, 65H10, 15A24.

1. Introduction

Matrix equations appear frequently in many areas of applied mathematics and play important roles in many applications, such as control theory, system theory [8–10, 18, 20, 27, 28, 30, 35–39]. Various kinds of matrix equations have received much attention in the literature (see, for example, [6, 7, 11, 30, 37, 43–45] and the references therein). Especially, Zhou et al. [45] considered the solution of matrix equation $X = Af(X)B + C$ with $f(X) = X^T$, $f(X) = \bar{X}$ and $f(X) = X^*$, where X^T, $\bar{X}$ and X^* represent the transpose, the conjugate and the conjugate transpose of the matrix X, respectively. It is proven that the solvability of these equations is equivalent to the solvability of some auxiliary standard Stein equations in the form

[†]the corresponding author.
Email address: macf@fjnu.edu.cn (C.-F. Ma), baohuahuang@126.com(B.-H. Huang)
[1]College of Mathematics and Informatics & FJKLMAA, Fujian Normal University, Fuzhou, 350117, China
*The authors were supported by National Key Research and Development Program of China (No. 2018YFC0603500) and the CAS/CAFEA international partnership Program for creative research teams (No. KZZD-EW-TZ-19 and KZZD-EW-TZ-15).

of $W = \mathcal{A}W\mathcal{B} + \mathcal{C}$, where the dimensions of the coefficient matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are the same as those of the original equation. A new iterative method for obtaining an approximation solution of Stein matrix equation $X = AXB + C$ is established by Zhou et al. [44]. In 2012, Al-Dubiban [1] studied the following system

$$\begin{cases} X - A^*Y^{-n}A = I_n, \\ Y - B^*X^{-m}B = I_n, \end{cases}$$

where n, m are two positive integers. And Al-Dubiban derived the necessary and sufficient conditions for the existence of positive definite solutions and presented an iterative algorithm for obtaining positive definite solutions of the system. Later, Al-Dubiban [2] further discussed the following problem

$$\begin{cases} X + A^*Y^{-n}A = I_n, \\ Y + B^*X^{-m}B = I_n \end{cases}$$

and proposed an iterative algorithm for finding the positive definite solutions of the system. Very recently, Huang and Ma [22] established the structure-preserving doubling algorithms for positive definite solution to the following system of nonlinear matrix equations

$$\begin{cases} X + A^*Y^{-1}A = I_n, \\ Y + B^*X^{-1}B = I_n. \end{cases} \tag{1.1}$$

On the other hand, consimilarity of complex matrices arises as a result of studying an antilinear operator referred to different bases in complex vector spaces and it plays an important role in modern quantum theory [25]. Based on the theory of consimilarity, linear matrix equations $AX - XB = C$ and $X - AXB = C$ which are generally derived by the similarity of square matrices have been respectively extended to $AX - \bar{X}B = C$ and $X - A\bar{X}B = C$ [4, 5, 26]. Similarly, the system (1.1) can be generalized to the following system of nonlinear matrix equations

$$\begin{cases} X + A^*\bar{Y}^{-1}A = I_n, \\ Y + B^*\bar{X}^{-1}B = I_n. \end{cases}$$

In this paper, we consider the following nonlinear matrix equations

$$\begin{cases} X - A^*\bar{Y}^{-1}A = I_n, \\ Y - B^*\bar{X}^{-1}B = I_n, \end{cases} \tag{1.2}$$

where $X, Y \in \mathbb{C}^{n\times n}$ are unknown matrices, $A, B \in \mathbb{C}^{n\times n}$ are given matrices and I_n is the identity matrix of order n. By introducing the notations

$$V = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}, \quad C = \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix}, \tag{1.3}$$

the system (1.2) can be equivalently reformulated as

$$V - C^*\bar{V}^{-1}C = I_{2n}. \tag{1.4}$$

Just as the linear case does, it is expected that nonlinear matrix equation (1.4) will find the possible applications in the design of modern quantum control theory. Hence, nonlinear matrix equation (1.4) has received much attention in recent years [23, 42]. By introducing and studying a matrix operator on complex matrices, Zhou et al. [42] showed that the existence of positive definite solutions to this class of nonlinear matrix equations is equivalent to the existence of positive definite solutions to another nonlinear matrix equation. They also studied the bounds of the positive definite solutions and proposed some sufficient conditions and necessary conditions for the existence of positive definite solutions. Huang and Ma [23] proposed the structure-preserving-doubling like algorithm for obtaining the positive definite solution of the nonlinear matrix equation (1.4). Li et al. [31] showed that the nonlinear matrix equation $X + A^*\bar{X}^{-1}A = I_n$ has a positive definite solution if and only if an auxiliary standard nonlinear matrix equation in the form of $Y + (\bar{A}A)^*Y^{-1}(\bar{A}A) = Q$ has a positive definite solution. They also proposed three iterative algorithm for finding the positive definite solution.

Motivated and inspired by the work mentioned above, in this paper, we consider the system (1.2). First, we transform the system (1.2) to the nonlinear matrix equation $V - C^*\bar{V}^{-1}C = I_{2n}$. Then, by means of Sherman-Moorison-Woodbury formula, we derive the relationship between the solutions of $V - C^*\bar{V}^{-1}C = I_{2n}$ and the well studied standard nonlinear matrix equation $Z + D^*Z^{-1}D = Q$, where D, Q are uniquely determined by C. The standard nonlinear matrix equation $Z + D^*Z^{-1}D = Q$ has been widely studied [3, 12, 17, 21, 24, 41]. There are several kinds of results for the analysis on this nonlinear matrix equation, such as, the fixed point iteration [14], structure-preserving doubling algorithm [16, 32] and some inversion free iterations [13, 34]. Based on reformulation equation $Z + D^*Z^{-1}D = Q$, we present a structure-preserving doubling algorithm and two modified structure-preserving doubling algorithms to compute the positive definite solution of the system. Also, cyclic reduction algorithm and two modified cyclic reduction algorithms for the positive definite solution of the system are proposed. In addition, some numerical examples are presented to illustrate the efficiency of the theoretical results and the behavior of the considered algorithms.

For convenience, we use the following notations throughout this paper. Let $\mathbb{C}^{n\times n}$ be the sets $n \times n$ complex matrices. For $A \in \mathbb{C}^{n\times n}$, we write $A^T, \bar{A}, A^*$, $A^{-1}, \|A\|_F$ and $\|A\|_2$ to denote transpose, the conjugate, the conjugate transpose, the inverse, Frobenius norm and the spectral norm of the matrix A, respectively. Saying A is positive definite matrix means A is a Hermitian positive definite matrix. For any $A = (a_{ij})$, $B = (b_{ij}) \in \mathbb{C}^{n\times n}$, we write $A \geq B$ (or $B \leq A$) if $A - B$ is a Hermitian positive semidefinite matrix. And we write $A > B$ (or $B < A$) if $A - B$ is a Hermitian positive definite matrix. Obviously, if $0 \leq A \leq B$, $\|A\|_2 \leq \|B\|_2$. In addition, 0 denotes the zero matrix of size implied by context. I_n denotes the n-order identity matrix and I_{2n} denotes the $2n$-order identity matrix.

The reminder of this paper is organized as follows. In Section 2, we give some preliminaries and related lemmas which will be used in this paper. Then we derive the relationship between the reformulation equation (1.4) and the standard nonlinear matrix equation $Z + D^*Z^{-1}D = Q$. Section 3 presents the structure-preserving doubling algorithms to compute the positive definite solution of the system (1.2). Cyclic reduction algorithm and two modified cyclic reduction algorithms for the positive definite solution of the system (1.2) are proposed in Section 4. The numerical examples are given to show the efficiency of the proposed algorithms in Section

5. The paper ends up with some conclusions in Section 6.

2. Reformulation of nonlinear matrix equation (1.4)

First, we recall the following Sherman-Moorison-Woodbury formula and some existing results about the positive definite solution of $V - C^*\bar{V}^{-1}C = Q$ and $V + C^*\bar{V}^{-1}C = Q$.

Lemma 2.1 (Theorem 1.8.1 [40] (Sherman-Morrison-Woodbury formula))**.** *Let A, B, C and D be some matrices of appropriate dimensions. Assume that A, C, $A+BCD$ and $C^{-1}+DA^{-1}B$ are all nonsingular matrices. Then*

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}.$$

Lemma 2.2 (Proposition 5.1 [15])**.** *The set of solutions to $V - C^*V^{-1}C = Q$ is nonempty, and admits a maximal element V_+ and a minimal element V_-, where V_+ satisfies $\rho(V_+^{-1}C) < 1$. Moreover, V_+ is the unique positive definite solution and it can be approximated by the fixed point iteration*

$$\begin{cases} V_0 = Q, \\ V_{k+1} = Q + C^*V_k^{-1}C, \ k = 0, 1, 2, \cdots. \end{cases}$$

Concerning matrix equation $V + C^*V^{-1}C = Q$, let us introduce the rational matrix function

$$\psi(\lambda) = \lambda C + Q + \lambda^{-1}C^*$$

defined on the unit circle $\mathcal{C}$ of the complex plane, which is Hermitian for any $\lambda \in \mathcal{C}$. This function is said to be regular if there exists at least a $\lambda \in \mathcal{C}$ such that $\psi(\lambda) \neq 0$. Then, we have the following fundamental results.

Lemma 2.3 (Theorem 2.1, Algorithm 4.1 [14])**.** *Matrix equation $V + C^*V^{-1}C = Q$ has a positive definite solution V if and only if $\psi(\lambda)$ is regular and $\psi(\lambda) \geq 0$ for all $\lambda \in \mathcal{C}$. Moreover, if matrix equation $V + C^*V^{-1}C = Q$ has a positive definite solution, then it has a maximal solution V_+ and a minimal solution V_- such that $0 < V_- \leq V \leq V_+$ for any positive definite solution V. In addition, the maximal positive definite solution V_+ satisfies $\rho(V_+^{-1}C) < 1$ and it can be approximated by the following basic fixed point iteration*

$$\begin{cases} V_0 = Q, \\ V_{k+1} = Q - C^*V_k^{-1}C, \ k = 0, 1, 2, \cdots. \end{cases}$$

Lemma 2.4 (Theorem 2.2 [33])**.** *Let C be a nonsingular matrix. Then V solves $V + C^*V^{-1}C = Q$ if and only if $U = Q - V$ solves $U + CU^{-1}C^* = Q$. In particular, if U_+ is the maximal positive definite solution of $U + CU^{-1}C^* = Q$, then $V_- = Q - U_+$ is the minimal positive definite solution of $V + C^*V^{-1}C = Q$.*

Following the idea of [42], it is easy to see that the nonlinear matrix equation (1.4) can be transformed to the following nonlinear matrix equation

$$V^\bigstar - (C^\blacktriangledown)^T(V^\bigstar)^{-1}C^\blacktriangledown = I_{4n}, \tag{2.1}$$

where the operators $(\cdot)^\bigstar$ and $(\cdot)^\blacktriangledown$ are defined as follows. For a complex matrix $A = A_1 + \mathrm{i}A_2 \in \mathbb{C}^{n\times n}$ with $A_1, A_2 \in \mathbb{R}^{n\times n}$, we have

$$A^\bigstar = \begin{pmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{pmatrix}, \quad A^\blacktriangledown = \begin{pmatrix} A_2 & A_1 \\ A_1 & -A_2 \end{pmatrix}.$$

Based on the properties of the operator $(\cdot)^\bigstar$, $(\cdot)^\blacktriangledown$ and Lemma 2.2, we have the following theorem.

Theorem 2.1. *The nonlinear matrix equation (1.4) always has a unique positive definite solution V_+, and the matrix sequence $\{V_k\}$ generated by the following iteration*

$$\begin{cases} V_0 = I_{2n}, \\ V_{k+1} = I_{2n} + C^*\bar{V}_k^{-1}C,\ k = 0, 1, 2, \cdots \end{cases} \tag{2.2}$$

converges to V_+.

Proof. The proof is similar to Proposition 5.2 in [15], so we omit here. □

In order to derive effective algorithms for the unique positive definite solution to nonlinear matrix equation (1.4), following the idea of Theorem 2 in [31], we introduce the precondition technique and transform (1.4) to the extensively studied nonlinear matrix equation $Z + D^*Z^{-1}D = Q$, where $D = \bar{C}C$ and $Q = I_{2n} + C^*C + \bar{C}\bar{C}^*$.

Theorem 2.2. *Let $V_+ > 0$ be the unique positive definite solution of (1.4). Then $V_+ = Z_+ - \bar{C}C^T$, where Z_+ is the maximal positive definite solution of*

$$Z + D^*Z^{-1}D = Q, \tag{2.3}$$

where $D = \bar{C}C$ and $Q = I_{2n} + C^*C + \bar{C}\bar{C}^*$.

Proof. First, we consider the nonlinear matrix equation (2.3). Since for all $\lambda \in \mathcal{C}$ with $|\lambda| = 1$, we have

$$\begin{aligned} \Psi(\lambda) &= Q + \lambda\bar{C}C + \lambda^{-1}\bar{C}C \\ &= I_{2n} + C^*C + \bar{C}\bar{C}^* + \lambda\bar{C}C + \lambda^{-1}(\bar{C}C)^* \\ &= I_{2n} + \bar{C}\bar{C}^* + |\lambda|^2C^*C + \lambda\bar{C}C + \bar{\lambda}(\bar{C}C)^* \\ &= I_{2n} + (\bar{C}^* + \lambda C)^*(\bar{C}^* + \lambda C) \\ &> 0. \end{aligned}$$

It then follows from Lemma 2.3 that nonlinear matrix equation (2.3) always has a positive definite solution, and hence has the maximal positive definite solution Z_+.

On the other hand, considering nonlinear matrix equation (1.4), from Theorem 2.1, we know that (1.4) always has a unique positive definite solution V_+, and the sequence $\{V_k\}$ generated by (2.2) converges to V_+.

Now we consider the subsequence $\{V_{2r+1}\}_{r=0}^{\infty}$ consisting of odd elements of $\{V_k\}$ in (2.2). It is obvious that $V_1 = I_{2n} + C^*C$ and for all $r \geq 1$,

$$\begin{aligned} V_{2r+1} &= I_{2n} + C^*\bar{V}_{2r}^{-1}C \\ &= I_{2n} + C^*(I_{2n} + \bar{C}^*V_{2r-1}^{-1}\bar{C})^{-1}C \end{aligned}$$

$$= I_{2n} + C^*[I_{2n} - \bar{C}^*(V_{2r-1} + \bar{C}\bar{C}^*)^{-1}\bar{C}]C$$
$$= I_{2n} + C^*C - (\bar{C}\bar{C}^*)(V_{2r-1} + \bar{C}\bar{C}^*)^{-1}(\bar{C}C), \tag{2.4}$$

where the Sherman-Moorison-Woodbury formula is used in the third equality. Hence

$$V_{2r+1} + \bar{C}\bar{C}^* = I_{2n} + C^*C + \bar{C}\bar{C}^* - (\bar{C}\bar{C}^*)(V_{2r-1} + \bar{C}\bar{C}^*)^{-1}(\bar{C}C). \tag{2.5}$$

Let

$$Z_r = V_{2r+1} + \bar{C}\bar{C}^*,\ r = 0, 1, 2, \cdots . \tag{2.6}$$

Then $Z_r \geq \bar{C}\bar{C}^*$. Moreover, from the relation (2.5), we conclude that Z_r satisfies

$$\begin{cases} Z_0 = V_1 + \bar{C}\bar{C}^* = I_{2n} + C^*C + \bar{C}\bar{C}^* = Q, \\ Z_r = Q - (\bar{C}C)^* Z_{r-1}^{-1}(\bar{C}C), \quad r = 1, 2, \cdots . \end{cases}$$

So, by Lemma 2.3, we know that $\{V_r\}$ is monotonically decreasing and converges to V_+, which is the maximal positive definite solution to the nonlinear matrix equation (2.3).

At the same time, $\{V_r\}$ converges to the unique positive definite solution V_+ of the nonlinear matrix equation (1.4), so does the odd sequence $\{V_{2r+1}\}_{r=0}^{\infty}$. Taking limits on both sides of (2.6), we have

$$Z_+ = \lim_{r\to\infty} Z_r = \lim_{r\to\infty} V_{2r+1} + \bar{C}\bar{C}^* = V_+ + \bar{C}\bar{C}^*,$$

which completes the proof. □

3. Structure-preserving doubling algorithm for the system (1.2)

In this section, we present the structure-preserving doubling algorithm for finding the unique positive definite solution of the system (1.2). Firstly, we give the structure-preserving doubling algorithm for nonlinear matrix equation (2.3), which has been studied in [32].

Lin and Xu [32] established the following recursive formulas:

$$\begin{cases} D_{k+1} = D_k(Q_k - P_k)^{-1}D_k, \\ Q_{k+1} = Q_k - D_k^*(Q_k - P_k)^{-1}D_k, \\ P_{k+1} = P_k + D_k(Q_k - P_k)^{-1}D_k^*, \quad k = 0, 1, 2, \cdots . \end{cases} \tag{3.1}$$

Algorithm 3.1 outlines the structure-preserving doubling algorithm for computing positive definite solution to the nonlinear matrix equation (2.3) in [32].

Algorithm 3.1. [32] (Structure-preserving doubling algorithm for nonlinear matrix equation (2.3)).

Step 1. Input the matrix $D \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $D_0 = D$, $Q_0 = Q$, $P_0 = 0$. Set $k = 0$.

Step 2. Obtain D_{k+1}, Q_{k+1}, P_{k+1} by the iterative scheme (3.1).

Step 3. If $\|Q_{k+1} - Q_k\|_F \leq \varepsilon$, stop. Otherwise, $k = k + 1$, go to Step 2.

The convergence theory of Algorithm 3.1 has been studied in [32], that is, we have the following lemma.

Lemma 3.1 (Theorem 4.1 [32]). *Assume that $Z > 0$ satisfies (2.3) and $S = Z^{-1}D$. If $\rho(S) < 1$, then*

(i) $\limsup_{k\to\infty} \sqrt[2^k]{\|D_k\|_2} \leq \rho(S)$;

(ii) $\limsup_{k\to\infty} \sqrt[2^k]{\|Q_k - Z\|_2} \leq \rho(S)^2$.

Based on Algorithm 3.1, we can derive the structure-preserving doubling algorithm for nonlinear matrix equation (1.4).

Algorithm 3.2. (Structure-preserving doubling algorithm for nonlinear matrix equation (1.4)).

Step 1. Input the matrix $C \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $C_0 = \bar{C}C$, $Q_0 = Q = I_{2n} + C^*C + \bar{C}\bar{C}^*$, $P_0 = 0$. Set $k = 0$.

Step 2. Obtain C_{k+1}, Q_{k+1}, P_{k+1} by the following iterative scheme

$$\begin{cases} C_{k+1} = C_k(Q_k - P_k)^{-1}C_k, \\ Q_{k+1} = Q_k - C_k^*(Q_k - P_k)^{-1}C_k, \\ P_{k+1} = P_k + C_k(Q_k - P_k)^{-1}C_k^*, \quad k = 0, 1, 2, \cdots . \end{cases} \tag{3.2}$$

Step 3. If $\|Q_{k+1} - Q_k\|_F \leq \varepsilon$, then $Z_+ = Q_{k+1}$, compute $V_+ = Z_+ - \bar{C}\bar{C}^*$, stop. Otherwise, $k = k + 1$, go to Step 2.

From the iterative scheme (3.2), we find that Algorithm 3.2 needs to compute a inverse of $2n\times 2n$ matrix and five product of $2n\times 2n$ matrices. However, the matrices in the original problem (1.2) are just n-order. Therefore, applying Algorithm 3.2 to compute the positive definite solution of the system (1.2) is unpractical even through its fast convergence rate. In order to reduce the calculation of Algorithm 3.2, we present the following theorem, which will be important to the studies of the structure-preserving doubling algorithm for the system (1.2).

Theorem 3.1. *Let C_k, Q_k, P_k be generated by Algorithm 3.2. Then, for all $k \geq 0$, C_k, Q_k, P_k have the forms of*

$$C_k = \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}, \quad Q_k = \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}, \quad P_k = \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix}, \tag{3.3}$$

respectively.

Proof. We prove this theorem by mathematical induction. Firstly, from Step 1 of Algorithm 3.2, we have

$$C_0 = \bar{C}C = \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix}\begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} = \begin{pmatrix} \bar{B}A & 0 \\ 0 & \bar{A}B \end{pmatrix}, \quad P_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$\begin{aligned}Q_0 &= I_{2n} + C^*C + \bar{C}\bar{C}^* \\ &= \begin{pmatrix} I_n & 0 \\ 0 & I_n \end{pmatrix} + \begin{pmatrix} 0 & A^* \\ B^* & 0 \end{pmatrix}\begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} + \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix}\begin{pmatrix} 0 & A^T \\ B^T & 0 \end{pmatrix} \\ &= \begin{pmatrix} I_n + A^*A + \bar{B}\bar{B}^* & 0 \\ 0 & I_n + B^*B + \bar{A}\bar{A}^* \end{pmatrix}.\end{aligned}$$

Suppose that C_k, P_k, Q_k have the forms of (3.3). Then, we immediately have

$$Q_k - P_k = \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix} - \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix} = \begin{pmatrix} E_k - G_k & 0 \\ 0 & F_k - H_k \end{pmatrix}.$$

It then follows from iterative scheme (3.2) that

$$\begin{aligned}C_{k+1} &= C_k(Q_k - P_k)^{-1}C_k \\ &= \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}\begin{pmatrix} E_k - G_k & 0 \\ 0 & F_k - H_k \end{pmatrix}^{-1}\begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \\ &= \begin{pmatrix} A_k(E_k - G_k)^{-1}A_k & 0 \\ 0 & B_k(F_k - H_k)^{-1}B_k \end{pmatrix};\end{aligned}$$

$$\begin{aligned}Q_{k+1} &= Q_k - C_k^*(Q_k - P_k)^{-1}C_k \\ &= \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix} - \begin{pmatrix} A_k^* & 0 \\ 0 & B_k^* \end{pmatrix}\begin{pmatrix} E_k - G_k & 0 \\ 0 & F_k - H_k \end{pmatrix}^{-1}\begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \\ &= \begin{pmatrix} E_k - A_k^*(E_k - G_k)^{-1}A_k & 0 \\ 0 & F_k - B_k^*(F_k - H_k)^{-1}B_k \end{pmatrix};\end{aligned}$$

$$\begin{aligned}P_{k+1} &= P_k + C_k(Q_k - P_k)^{-1}C_k^* \\ &= \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix} - \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}\begin{pmatrix} E_k - G_k & 0 \\ 0 & F_k - H_k \end{pmatrix}^{-1}\begin{pmatrix} A_k^* & 0 \\ 0 & B_k^* \end{pmatrix} \\ &= \begin{pmatrix} G_k + A_k(E_k - G_k)^{-1}A_k^* & 0 \\ 0 & H_k + B_k(F_k - H_k)^{-1}B_k^* \end{pmatrix}.\end{aligned}$$

Hence, $C_{k+1}, P_{k+1}, Q_{k+1}$ also have the forms of (3.3). The proof is completed. □

Remark 3.1. From the proof of Theorem 3.1, we have the following formulas

$$\begin{cases} A_{k+1} = A_k(E_k - G_k)^{-1}A_k, \\ E_{k+1} = E_k - A_k^*(E_k - G_k)^{-1}A_k, \\ G_{k+1} = G_k + A_k(E_k - G_k)^{-1}A_k^*, \quad k = 0, 1, 2, \cdots, \end{cases} \tag{3.4}$$

$$\begin{cases} B_{k+1} = B_k(F_k - H_k)^{-1}B_k, \\ F_{k+1} = F_k - B_k^*(F_k - H_k)^{-1}B_k, \\ H_{k+1} = H_k + B_k(F_k - H_k)^{-1}B_k^*, \quad k = 0, 1, 2, \cdots . \end{cases} \tag{3.5}$$

Now we are in a position to give the structure-preserving doubling algorithm for solving the system (1.2).

Algorithm 3.3. (Structure-preserving doubling algorithm for the system (1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $A_0 = \bar{B}A$, $G_0 = 0$, $E_0 = I_n + A^*A + \bar{B}\bar{B}^*$, $B_0 = \bar{A}B$, $F_0 = I_n + B^*B + \bar{A}\bar{A}^*$, $H_0 = 0$. Set $k = 0$.

Step 2. Compute A_{k+1}, E_{k+1}, G_{k+1} by the iterative scheme (3.4) and B_{k+1}, F_{k+1}, H_{k+1} by the iterative scheme (3.5).

Step 3. If $\|E_{k+1} - E_k\|_F + \|F_{k+1} - F_k\|_F \leq \varepsilon$, then $X_+^\diamond = E_{k+1}$, $Y_+^\diamond = F_{k+1}$, compute $X_+ = X_+^\diamond - \bar{B}\bar{B}^*$, $Y_+ = Y_+^\diamond - \bar{A}\bar{A}^*$, stop . Otherwise, $k = k + 1$, go to Step 2.

Next, we establish the convergence theory for Algorithm 3.3 based on Lemma 3.1.

Theorem 3.2. *Assume that $X > 0$ and $Y > 0$ satisfies the system (1.2), let $X^\diamond = X + \bar{B}\bar{B}^*$, $Y^\diamond = Y + \bar{A}\bar{A}^*$ and $S_1 = (X^\diamond)^{-1}\bar{B}A$, $S_2 = (Y^\diamond)^{-1}\bar{A}B$. If $\rho(S_1) < 1$ and $\rho(S_2) < 1$, then*

(i) $\limsup_{k\to\infty} \sqrt[2^k]{\|A_k\|_2} \leq \max\{\rho(S_1), \rho(S_2)\}$,
$\limsup_{k\to\infty} \sqrt[2^k]{\|B_k\|_2} \leq \max\{\rho(S_1), \rho(S_2)\}$;

(ii) $\limsup_{k\to\infty} \sqrt[2^k]{\|E_k - X^\diamond\|_2} \leq \max\{\rho(S_1), \rho(S_2)\}^2$,
$\limsup_{k\to\infty} \sqrt[2^k]{\|F_k - Y^\diamond\|_2} \leq \max\ \{\rho(S_1),\ \rho(S_2)\}^2$.

Proof. If $X > 0$ and $Y > 0$ satisfies the system (1.2), then

$$V = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}$$

satisfies the nonlinear matrix equation (1.4).

From Theorem 2.2, we know that

$$V + \bar{C}\bar{C}^* = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} + \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix}\begin{pmatrix} 0 & \bar{A}^* \\ \bar{B}^* & 0 \end{pmatrix} = \begin{pmatrix} X + \bar{B}\bar{B}^* & 0 \\ 0 & Y + \bar{A}\bar{A}^* \end{pmatrix} = \begin{pmatrix} X^\diamond & 0 \\ 0 & Y^\diamond \end{pmatrix}$$

is the solution of nonlinear matrix equation (2.3). By the definition of S in Lemma 3.1, we have

$$S = \begin{pmatrix} X^\diamond & 0 \\ 0 & Y^\diamond \end{pmatrix}^{-1}\begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix}\begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} = \begin{pmatrix} (X^\diamond)^{-1}\bar{B}A & 0 \\ 0 & (Y^\diamond)^{-1}\bar{A}B \end{pmatrix} = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}. \tag{3.6}$$

On the other hand, from Algorithm 3.3 and Theorem 3.1, we get

$$D_k = C_k = \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}, \quad Q_k = \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}. \tag{3.7}$$

Together (3.6) with (3.7) yields

$$\limsup_{k\to\infty} \sqrt[2^k]{\|A_k\|_2} \le \limsup_{k\to\infty} \sqrt[2^k]{\|D_k\|_2} \le \rho(S) = \max\{\rho(S_1), \rho(S_2)\}$$

and

$$\begin{aligned}\limsup_{k\to\infty} \sqrt[2^k]{\|E_k - X^\diamond\|_2} &\le \limsup_{k\to\infty} \sqrt[2^k]{\|Q_k - (V + \bar{C}\bar{C}^*)\|_2} \\ &\le \rho(S)^2 = \max\{\rho(S_1), \rho(S_2)\}^2,\end{aligned}$$

which completes the proof. □

From Algorithm 3.3, we find that the iterative scheme (3.4) and (3.5) are independent of each other. So, we have the following two independent algorithms.

Algorithm 3.4. (Modified structure-preserving doubling algorithm for the system(1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \ge 0$. Compute $A_0 = \bar{B}A$, $G_0 = 0$ and $E_0 = I_n + A^*A + \bar{B}\bar{B}^*$. Set $k = 0$.

Step 2. Compute A_{k+1}, E_{k+1}, G_{k+1} by the iterative scheme (3.4).

Step 3. If $\|E_{k+1} - E_k\|_F \le \varepsilon$, then $X_+^\diamond = E_{k+1}$, compute $X_+ = X_+^\diamond - \bar{B}\bar{B}^*$, $Y = I_n - B^*\bar{X}_+^{-1}B$. Otherwise, $k = k + 1$, go to Step 2.

Algorithm 3.5. (Modified structure-preserving doubling algorithm for the system (1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \ge 0$. Compute $B_0 = \bar{A}B$, $F_0 = I_n + B^*B + \bar{A}\bar{A}^*$ and $H_0 = 0$. Set $k = 0$.

Step 2. Compute B_{k+1}, F_{k+1}, H_{k+1} by the iterative scheme (3.5).

Step 3. If $\|F_{k+1} - F_k\|_F \le \varepsilon$, stop. Set $Y_+^\diamond = F_{k+1}$, compute $Y_+ = Y_+^\diamond - \bar{A}\bar{A}^*$ and $X = I_n - A^*\bar{Y}_+^{-1}A$. Otherwise, $k = k + 1$, go to Step 2.

Remark 3.2. Although the variables X and Y in the system (1.2) are coupled to each other, Algorithms 3.4 and 3.5 can fast implement decoupling calculation of X and Y. Therefore, we can expect that the CPU time of Algorithms 3.4 and 3.5 is about half of that of Algorithm 3.3.

4. Cyclic reduction algorithm for the system (1.2)

In this section, we present the cyclic reduction algorithm for finding the unique positive definite solution of the system (1.2). First, we review the cyclic reduction algorithm for solving nonlinear matrix equation (2.3), which has been discussed in [33].

Algorithm 4.1. [33] (Cyclic reduction algorithm for nonlinear matrix equation (2.3)).

Step 1. Input the matrix $D \in \mathbb{C}^{2n\times 2n}$ and tolerance error $\varepsilon \ge 0$. Compute $D_0 = D$, $Q_0 = Q$, $Z_0 = Q$. Set $k = 0$.

Step 2. Obtain D_{k+1}, Q_{k+1}, Z_{k+1} by the following iterative scheme

$$\begin{cases} D_{k+1} = D_k Q_k^{-1} D_k, \\ Q_{k+1} = Q_k - D_k^* Q_k^{-1} D_k - D_k Q_k^{-1} D_k^*, \\ Z_{k+1} = Z_k - D_k^* Q_k^{-1} D_k, \quad k = 0, 1, 2, \cdots . \end{cases} \tag{4.1}$$

Step 3. If $\|Z_{k+1} - Z_k\|_F \leq \varepsilon$, stop. Otherwise, $k = k + 1$, go to Step 2.

The convergence theory of Algorithm 4.1 has been studied in [33], that is, we have the following lemma.

Lemma 4.1 (Theorem 4.1 [33]). *For the sequences of matrices $\{Z_k\}$, $\{Q_k\}$ defined in Algorithm 4.1, the following results hold:*

(i) $0 < Z_{k+1} \leq Z_k$, $0 < Q_{k+1} \leq Q_k$, $k = 0, 1, 2, \cdots$;

(ii) $\{Q_k\}$ *and* $\{Q_k^{-1}\}$ *are bounded in norm.*

Moreover, assume that $Z > 0$ *satisfies (2.3) and* $S = Z^{-1}D$. *If* $\sigma = \rho(S) < 1$, *then for any* $\varepsilon > 0$ *and for any matrix norm* $\|\cdot\|$, *it holds that*

(iii) $\|D_k\| = O((\sigma + \varepsilon)^{2^k})$;

(iiii) $\|I_{2n} - Z_k Z^{-1}\| = O((\sigma + \varepsilon)^{2^{k+1}})$.

By means of Theorem 2.2, we give the cyclic reduction algorithm for nonlinear matrix equation (1.4).

Algorithm 4.2. (Cyclic reduction algorithm for nonlinear matrix equationv(1.4)).

Step 1. Input the matrix $C \in \mathbb{C}^{2n \times 2n}$ and tolerance error $\varepsilon \geq 0$. Compute $C_0 = \bar{C}C$, $Q_0 = Q = I_{2n} + C^*C + \bar{C}\bar{C}^*$, $Z_0 = Q = I_{2n} + C^*C + \bar{C}\bar{C}^*$. Set $k = 0$.

Step 2. Obtain C_{k+1}, Q_{k+1}, Z_{k+1} by the following iterative scheme

$$\begin{cases} C_{k+1} = C_k Q_k^{-1} C_k, \\ Q_{k+1} = Q_k - C_k^* Q_k^{-1} C_k - C_k Q_k^{-1} C_k^*, \\ Z_{k+1} = Z_k - C_k^* Q_k^{-1} C_k, \quad k = 0, 1, 2, \cdots \end{cases} \tag{4.2}$$

Step 3. If $\|Z_{k+1} - Z_k\|_F \leq \varepsilon$, then $Z_+ = Z_{k+1}$, compute $V_+ = Z_+ - \bar{C}\bar{C}^*$, stop. Otherwise, $k = k + 1$, go to Step 2.

From the iterative scheme (4.2), we find that Algorithm 4.2 needs to compute a inverse of $2n \times 2n$ matrix and seven product of $2n \times 2n$ matrices. However, the matrices in the original problem (1.2) are just n-order. Therefore, applying Algorithm 4.2 to compute the positive definite solution of the system (1.2) is unpractical even through its fast convergence rate. In order to reduce the calculation of Algorithm 4.2, we present the following theorem, which will be important to the studies of the structure-preserving doubling algorithm for the system (1.2).

Theorem 4.1. *Let C_k, Q_k, Z_k be generated by Algorithm 4.2. Then, for all $k \geq 0$, C_k, Q_k, Z_k have the forms of*

$$C_k = \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}, \quad Q_k = \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}, \quad Z_k = \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix}, \tag{4.3}$$

respectively.

Proof. We prove this theorem by mathematical induction. First, for $k = 0$, from Step 1 of Algorithm 4.2, we have

$$C_0 = \bar{C}C = \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix} \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} = \begin{pmatrix} \bar{B}A & 0 \\ 0 & \bar{A}B \end{pmatrix}$$

and

$$\begin{aligned} Q_0 &= I_{2n} + C^*C + \bar{C}\bar{C}^* \\ &= \begin{pmatrix} I_n & 0 \\ 0 & I_n \end{pmatrix} + \begin{pmatrix} 0 & A^* \\ B^* & 0 \end{pmatrix} \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} + \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix} \begin{pmatrix} 0 & A^T \\ B^T & 0 \end{pmatrix} \\ &= \begin{pmatrix} I_n + A^*A + \bar{B}\bar{B}^* & 0 \\ 0 & I_n + B^*B + \bar{A}\bar{A}^* \end{pmatrix}. \end{aligned}$$

Hence

$$Z_0 = Q_0 = Q = \begin{pmatrix} I_n + A^*A + \bar{B}\bar{B}^* & 0 \\ 0 & I_n + B^*B + \bar{A}\bar{A}^* \end{pmatrix}.$$

Suppose that C_k, Q_k, Z_k have the forms of (4.3). It then follows from the iteration formula (4.2) that

$$\begin{aligned} C_{k+1} = C_k Q_k^{-1} C_k &= \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}^{-1} \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \\ &= \begin{pmatrix} A_k E_k^{-1} A_k & 0 \\ 0 & B_k F_k^{-1} B_k \end{pmatrix}; \end{aligned}$$

$$\begin{aligned} Q_{k+1} &= Q_k - C_k^* Q_k^{-1} C_k - C_k Q_k^{-1} C_k^* \\ &= \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix} - \begin{pmatrix} A_k^* & 0 \\ 0 & B_k^* \end{pmatrix} \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}^{-1} \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \\ &\quad - \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}^{-1} \begin{pmatrix} A_k^* & 0 \\ 0 & B_k^* \end{pmatrix} \\ &= \begin{pmatrix} E_k - A_k^* E_k^{-1} A_k - A_k E_k^{-1} A_k^* & 0 \\ 0 & F_k - B_k^* F_k^{-1} B_k - B_k F_k^{-1} B_k^* \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} Z_{k+1} &= Z_k - C_k^* Q_k^{-1} C_k \\ &= \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix} - \begin{pmatrix} A_k^* & 0 \\ 0 & B_k^* \end{pmatrix} \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}^{-1} \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} G_k - A_k^* E_k^{-1} A_k & 0 \\ 0 & H_k - B_k^* F_k^{-1} B_k \end{pmatrix}.$$

Hence, $C_{k+1}, Q_{k+1}, Z_{k+1}$ also have the forms of (4.3). The proof is completed. □

Remark 4.1. From the proof of Theorem 4.1, we have the following formulas

$$\begin{cases} A_{k+1} = A_k E_k^{-1} A_k, \\ E_{k+1} = E_k - A_k^* E_k^{-1} A_k - A_k E_k^{-1} A_k^*, \\ G_{k+1} = G_k - A_k^* E_k^{-1} A_k, \quad k = 0, 1, 2, \cdots, \end{cases} \tag{4.4}$$

$$\begin{cases} B_{k+1} = B_k F_k^{-1} B_k, \\ F_{k+1} = F_k - B_k^* F_k^{-1} B_k - B_k F_k^{-1} B_k^*, \\ H_{k+1} = H_k - B_k^* F_k^{-1} B_k, \quad k = 0, 1, 2, \cdots. \end{cases} \tag{4.5}$$

Now we are in a position to give the cyclic reduction algorithm for solving the system (1.2).

Algorithm 4.3. (Cyclic reduction algorithm for the system (1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $A_0 = \bar{B}A$, $E_0 = G_0 = I_n + A^*A + \bar{B}\bar{B}^*$, $B_0 = \bar{A}B$, $F_0 = H_0 = I_n + B^*B + \bar{A}\bar{A}^*$. Set $k = 0$.

Step 2. Compute A_{k+1}, E_{k+1}, G_{k+1} by the iterative scheme (4.4) and B_{k+1}, F_{k+1}, H_{k+1} by the iterative scheme (4.5).

Step 3. If $\|G_{k+1} - G_k\|_F + \|H_{k+1} - H_k\|_F \leq \varepsilon$, then $X_+^\diamond = G_{k+1}$, $Y_+^\diamond = H_{k+1}$, compute $X_+ = X_+^\diamond - \bar{B}\bar{B}^*$, $Y_+ = Y_+^\diamond - \bar{A}\bar{A}^*$, stop . Otherwise, $k = k + 1$, go to Step 2.

Next, we establish the convergence theory of Algorithm 4.3 based on Lemma 4.1.

Theorem 4.2. *For the sequences of matrices $\{E_k\}$, $\{F_k\}$, $\{G_k\}$, $\{H_k\}$ defined in Algorithm 4.3, the following hold:*

(i) $0 < E_{k+1} \leq E_k$, $0 < F_{k+1} \leq F_k$, $0 < G_{k+1} \leq G_k$, $0 < H_{k+1} \leq H_k$, $k = 0, 1, 2, \cdots$;

(ii) $\{E_k\}$, $\{E_k^{-1}\}$ *and* $\{F_k\}$, $\{F_k^{-1}\}$ *are bounded in norm.*

Moreover, assume that $X > 0$, $Y > 0$ satisfies (1.2), let $X^\diamond = X + \bar{B}\bar{B}^*$, $Y^\diamond = Y + \bar{A}\bar{A}^*$ and $S_1 = (X^\diamond)^{-1}\bar{B}A$, $S_2 = (Y^\diamond)^{-1}\bar{A}B$. If $\sigma = \max\{\rho(S_1), \rho(S_2)\} < 1$, then for any $\varepsilon > 0$ and for any matrix norm $\|\cdot\|$, it holds that

(iii) $\|A_k\| = O((\sigma + \varepsilon)^{2^k})$, $\|B_k\| = O((\sigma + \varepsilon)^{2^k})$;

(iiii) $\|I_n - G_k(X^\diamond)^{-1}\| = O((\sigma + \varepsilon)^{2^{k+1}})$, $\|I_n - H_k(Y^\diamond)^{-1}\| = O((\sigma + \varepsilon)^{2^{k+1}})$.

Proof. First, from Algorithm 4.1, Algorithm 4.3 and Theorem 4.1, we get

$$D_k = C_k = \begin{pmatrix} A_k & 0 \\ 0 & B_k \end{pmatrix}, \quad Q_k = \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}, \quad Z_k = \begin{pmatrix} G_k & 0 \\ 0 & H_k \end{pmatrix}. \tag{4.6}$$

It then follows from Lemma 4.1 that

$$0 < E_{k+1} \leq E_k,\ 0 < F_{k+1} \leq F_k,\ 0 < G_{k+1} \leq G_k,\ 0 < H_{k+1} \leq H_k,\ k = 0, 1, 2, \cdots,$$

and

$$\{E_k\},\ \{E_k^{-1}\},\ \{F_k\},\ \{F_k^{-1}\}$$

are bounded in norm. If $X > 0$ and $Y > 0$ satisfies the system (1.2), then

$$V = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}$$

satisfies the nonlinear matrix equation (1.4).

From Theorem 2.2, we know that

$$V + \bar{C}\bar{C}^* = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} + \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix} \begin{pmatrix} 0 & \bar{A}^* \\ \bar{B}^* & 0 \end{pmatrix} = \begin{pmatrix} X + \bar{B}\bar{B}^* & 0 \\ 0 & Y + \bar{A}\bar{A}^* \end{pmatrix} = \begin{pmatrix} X^\diamond & 0 \\ 0 & Y^\diamond \end{pmatrix}$$

is the solution of nonlinear matrix equation (2.3). By the definition of S in Lemma 4.1, we have

$$S = \begin{pmatrix} X^\diamond & 0 \\ 0 & Y^\diamond \end{pmatrix}^{-1} \begin{pmatrix} 0 & \bar{B} \\ \bar{A} & 0 \end{pmatrix} \begin{pmatrix} 0 & B \\ A & 0 \end{pmatrix} = \begin{pmatrix} (X^\diamond)^{-1}\bar{B}A & 0 \\ 0 & (Y^\diamond)^{-1}\bar{A}B \end{pmatrix} = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}. \tag{4.7}$$

Together (4.6) with (4.7) yields

$$\|A_k\| \leq \|Z_k\| \leq O((\sigma + \varepsilon)^{2^k}),\ \|B_k\| \leq \|Z_k\| \leq O((\sigma + \varepsilon)^{2^k})$$

and

$$\|I_n - G_k(X^\diamond)^{-1}\| \leq \|I_{2n} - Z_k Z^{-1}\| = O((\sigma + \varepsilon)^{2^{k+1}}),$$
$$\|I_n - H_k(Y^\diamond)^{-1}\| \leq \|I_{2n} - Z_k Z^{-1}\| = O((\sigma + \varepsilon)^{2^{k+1}}),$$

which completes the proof. □

From Algorithm 4.3, we find that the iterative scheme (4.4) and (4.5) are independent of each other. So, we have the following two independent algorithms.

Algorithm 4.4. (Modified cyclic reduction algorithm for the system (1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $A_0 = \bar{B}A$ and $E_0 = G_0 = I_n + A^*A + \bar{B}\bar{B}^*$. Set $k = 0$.

Step 2. Compute A_{k+1}, E_{k+1}, G_{k+1} by the iterative scheme (4.4).

Step 3. If $\|G_{k+1} - G_k\|_F \leq \varepsilon$, then $X_+^\diamond = G_{k+1}$, compute $X_+ = X_+^\diamond - \bar{B}\bar{B}^*$, $Y = I_n - B^*\bar{X}_+^{-1}B$. Otherwise, $k = k + 1$, go to Step 2.

Algorithm 4.5. (Modified cyclic reduction algorithm for the system (1.2)).

Step 1. Input the matrix $A, B \in \mathbb{C}^{n\times n}$ and tolerance error $\varepsilon \geq 0$. Compute $B_0 = \bar{A}B$ and $F_0 = H_0 = I_n + B^*B + \bar{A}\bar{A}^*$. Set $k = 0$.

Step 2. Compute B_{k+1}, F_{k+1}, H_{k+1} by the iterative scheme (4.5).

Step 3. If $\|H_{k+1} - H_k\|_F \leq \varepsilon$, stop. Set $Y_+^\diamond = H_{k+1}$, compute $Y_+ = Y_+^\diamond - \bar{A}\bar{A}^*$ and $X = I_n - A^*\bar{Y}_+^{-1}A$. Otherwise, $k = k + 1$, go to Step 2.

Remark 4.2. Although the variables X and Y in the system (1.2) are coupled to each other, Algorithms 4.4 and 4.5 can fast implement decoupling calculation of X and Y. Therefore, we can expect that the CPU time of Algorithms 4.4 and 4.5 is about half of that of Algorithm 4.3.

5. Numerical experiments

In this section, we report several numerical results to examine the efficiency of all the theoretical results. All of the tests were run on the Intel (R) Core (TM), where the CPU is 2.40 GHz and the memory is 8.0 GB, the programming language is MATLAB R2015a.

Example 5.1. In this example, we consider the system (1.2) with the coefficient matrices A and B given by

$$A = \begin{pmatrix} 0.6294 - 0.1565i & 0.2647 + 0.3115i & 0.9150 + 0.3575i & 0.9143 + 0.3110i \\ 0.8116 + 0.8315i & -0.8049 - 0.9286i & 0.9298 + 0.5155i & -0.0292 - 0.6576i \\ -0.7460 + 0.5844i & -0.4430 + 0.6983i & -0.6848 + 0.4863i & 0.6006 + 0.4121i \\ 0.8268 + 0.9190i & 0.0938 + 0.8680i & 0.9412 - 0.2155i & -0.7162 - 0.9363i \end{pmatrix},$$

$$B = \begin{pmatrix} -0.4462 + 0.4187i & 0.3897 + 0.3102i & -0.1225 + 0.9195i & -0.6263 + 0.5025i \\ -0.9077 + 0.5094i & -0.3658 - 0.6748i & -0.2369 - 0.3192i & -0.0205 - 0.4898i \\ -0.8057 - 0.4479i & 0.9004 - 0.7620i & 0.5310 + 0.1705i & -0.1088 + 0.0119i \\ 0.6469 + 0.3594i & -0.9311 - 0.0033i & 0.5904 - 0.5524i & 0.2926 + 0.3982i \end{pmatrix},$$

which is generated randomly by the function $(2 * \mathrm{rand(n)} - 1) + \mathrm{i} * (2 * \mathrm{rand(n)} - 1)$ in MATLAB.

We compute the unique positive definite solution X_+, Y_+ of the system (1.2) by Algorithms 3.3, 3.4 and 3.5, Algorithms 4.3, 4.4 and 4.5. In this example, we stop our algorithm as the description of all kinds of algorithm with $\varepsilon = 10^{-14}$. The sufficiently accurate unique positive definite solution of the system (1.2) is obtained as

$$X_+ = \begin{pmatrix} 3.3787 + 0.0000\mathrm{i} & 0.7033 - 0.1848\mathrm{i} & 1.7926 - 0.8718\mathrm{i} & -1.4033 - 1.0355\mathrm{i} \\ 0.7033 + 0.1848\mathrm{i} & 3.3038 - 0.0000\mathrm{i} & 0.0475 + 0.2125\mathrm{i} & -0.2006 + 0.2291\mathrm{i} \\ 1.7926 + 0.8718\mathrm{i} & 0.0475 - 0.2125\mathrm{i} & 3.6834 + 0.0000\mathrm{i} & 0.0506 - 2.2129\mathrm{i} \\ -1.4033 + 1.0355\mathrm{i} & -0.2006 - 0.2291\mathrm{i} & 0.0506 + 2.2129\mathrm{i} & 3.9219 + 0.0000\mathrm{i} \end{pmatrix},$$

and

$$Y_+ = \begin{pmatrix} 2.1819 + 0.0000\mathrm{i} & -0.0527 + 0.9413\mathrm{i} & 0.2886 + 0.2401\mathrm{i} & 0.2409 + 0.5423\mathrm{i} \\ -0.0527 - 0.9413\mathrm{i} & 2.1513 - 0.0000\mathrm{i} & 0.3501 + 0.0314\mathrm{i} & 0.4666 - 0.0160\mathrm{i} \\ 0.2886 - 0.2401\mathrm{i} & 0.3501 - 0.0314\mathrm{i} & 1.5011 + 0.0000\mathrm{i} & 0.3567 + 0.2430\mathrm{i} \\ 0.2409 - 0.5423\mathrm{i} & 0.4666 + 0.0160\mathrm{i} & 0.3567 - 0.2430\mathrm{i} & 1.5485 + 0.0000\mathrm{i} \end{pmatrix}.$$

We list the number of required iterations k and the corresponding $\mathrm{Res}(X_k, Y_k)$, where $\mathrm{Res}(X_k, Y_k)$ is defined as follows

$$\mathrm{Res}(X_k, Y_k) = \|I_n - X_k + A^* \bar{Y}_k^{-1} A\|_F + \|I_n - Y_k + B^* \bar{X}_k^{-1} B\|_F.$$

The corresponding numerical results are given in Table 1. From Table 1, we find that

Table 1. Numerical comparison of the testing algorithms for Example 5.1

	Algorithm 3.3		Algorithm 3.4		Algorithm 3.5
k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$
1	0.4481	1	0.1631	1	0.1390
2	0.0021	2	8.5034e-04	2	5.6614e-04
3	8.2274e-08	3	2.6672e-08	3	2.6037e-08
4	6.9643e-15	4	3.1411e-15	4	1.8683e-15
5	6.9643e-15	5	3.1411e-15	5	1.8683e-15
	Algorithm 4.3		Algorithm 4.4		Algorithm 4.5
k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$
1	0.4481	1	0.1631	1	0.1390
2	0.0021	2	8.5034e-04	2	5.6614e-04
3	8.2274e-08	3	2.6672e-08	3	2.6037e-08
4	6.9643e-15	4	3.1411e-15	4	1.8683e-15
5	6.9643e-15	5	3.1411e-15	5	1.8683e-15

structure-preserving doubling algorithm and cyclic reduction algorithm are actually the same and they have quadratically convergence rate and good numerical stability, which has been discussed in [32, 33].

Example 5.2. In this example, we consider the system (1.2) with matrices A and B are given as follows:

$$A = \mathrm{diag}\left\{\frac{-99}{2n}, \frac{-98}{2n}, \cdots, \frac{n-100}{2n}\right\},\ B = \mathrm{diag}\left\{\frac{1}{n+50}, \frac{2}{n+50}, \cdots, \frac{n}{n+50}\right\}.$$

We could obtain the unique positive definite solution X_+, Y_+ by the algorithm mentioned above. In this example, we stop our algorithm as the description of all kinds of algorithm with $\varepsilon = 10^{-14}$. We take $n = 64$ to test our algorithm. The numerical results are given in Table 2. Table 2 further shows that our proposed algorithm is efficiency and the convergence rate of our proposed algorithm is quite fast.

Example 5.3. In this example, we use the function $(2 * \mathrm{rand}(8, 8) - 1) + i * (2*$ rand $(8, 8) - 1)$ in MATLAB to generate the matrices A and B.

In this example, we stop our algorithm as the description of all kinds of algorithm with $\varepsilon = 10^{-14}$. In Figure 1, the convergence histories of mentioned algorithms are depicted, where the residual is defined as follows

$$\mathrm{Res}(X_k, Y_k) = \|I_n - X_k + A^* \bar{Y}_k^{-1} A\|_F + \|I_n - Y_k + B^* \bar{X}_k^{-1} B\|_F.$$

Table 2. Numerical comparison of the testing algorithms for Example 5.2

Algorithm 3.3		Algorithm 3.4		Algorithm 3.5	
k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$
1	0.0042	1	0.0018	1	0.0018
2	1.0274e-06	2	4.4472e-07	2	4.4472e-07
3	6.9694e-14	3	3.0831e-14	3	3.0773e-14
4	2.5924e-15	4	1.5806e-15	4	1.6164e-15
5	2.5924e-15	5	1.5806e-15	5	1.6164e-15
Algorithm 4.3		Algorithm 4.4		Algorithm 4.5	
k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$	k	$\mathrm{Res}(X_k, Y_k)$
1	0.0042	1	0.0018	1	0.0018
2	1.0274e-06	2	4.4472e-07	2	4.4472e-07
3	6.9694e-14	3	3.0831e-14	3	3.0773e-14
4	2.5924e-15	4	1.5806e-15	4	1.6164e-15
5	2.5924e-15	5	1.5806e-15	5	1.6164e-15

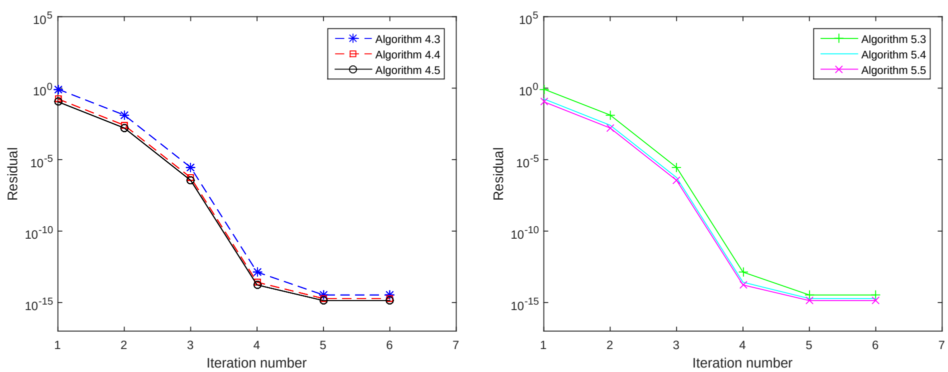

Figure 1. The comparison of convergence rates of different algorithms for Example 5.3

From Figure 1, we find that the residual of Algorithms 3.4 and 3.5 decrease faster than that of Algorithm 3.3. Similarly, the residual of Algorithms 4.4 and 4.5 decrease faster than that of Algorithm 4.3.

Example 5.4. In this example, we compare the iteration steps and the CPU time required to compute a sufficient accurate solution by Algorithms 3.1, 3.3, 4.1 and 4.3. We first construct the following block matrices

$$C = \begin{pmatrix} \mathrm{zeros}(n,n) & 2*\mathrm{rand}(n)-1+i*(2*\mathrm{rand}(n)-1) \\ 2*\mathrm{rand}(n)-1+i*(2*\mathrm{rand}(n)-1) & \mathrm{zeros}(n,n) \end{pmatrix},$$

$$L = \begin{pmatrix} \mathrm{zeros}(n,n) & 2*\mathrm{rand}(n)-1+i*(2*\mathrm{rand}(n)-1) \\ 2*\mathrm{rand}(n)-1+i*(2*\mathrm{rand}(n)-1) & \mathrm{zeros}(n,n) \end{pmatrix}.$$

Set $V = L.' * L + 1/(2*n)*\mathrm{eye}(2*n)$ be the exact solution of the nonlinear matrix equation (1.4). Then, by the relations (1.3) and (1.4), we can construct the matrices

A and B.

Now we compute the unique positive definite solution of the system (1.2) for different values of n. We compare the performance of all algorithms from the aspects of iteration steps (denoted by 'Iter'), elapsed CPU time in seconds (denoted by 'Time') and the Frobenius norm of the residuals (denoted by 'Res'). Here, 'Res' is defined as

$$\mathrm{Res}(X_k, Y_k) = \|I_n - X_k + A^* \bar{Y}_k^{-1} A\|_F + \|I_n - Y_k + B^* \bar{X}_k^{-1} B\|_F.$$

And, we stop our algorithm as the description of all kinds of algorithm with $\varepsilon = 10^{-13}$. The corresponding numerical results are listed in Table 3. From Table 3, we find that the iteration steps of Algorithm 3.1 is the same as that of Algorithm 3.3. However, the CPU time of Algorithm 3.3 is less than that of Algorithm 3.1. There are similar results about the relationship between Algorithm 4.1 and Algorithm 4.3. That shows that Algorithms 3.3 and 4.3 outperform Algorithms 3.1 and 4.1, respectively.

Table 3. Numerical results for Example 5.4.

	n	60	90	120	200
Algorithm 3.3	Iter	5	5	5	4
	Time	0.0398	0.0723	0.1186	0.3200
	Res	2.3701e-15	3.0396e-15	7.7951e-15	4.9472e-15
Algorithm 3.1	Iter	5	5	5	4
	Time	0.0445	0.1087	0.2129	0.7722
	Res	2.3649e-15	3.0579e-15	7.7646e-15	4.8259e-15
Algorithm 4.3	Iter	5	5	5	4
	Time	0.0261	0.0603	0.1068	0.3351
	Res	2.3701e-15	3.0396e-15	7.7951e-15	4.9472e-15
Algorithm 4.1	Iter	5	5	5	4
	Time	0.0374	0.0856	0.1640	0.5939
	Res	2.3649e-15	3.0579e-15	7.7645e-15	4.8259e-15

6. Conclusions

In this paper, we present a structure-preserving doubling Algorithm 3.3 and two modified structure-preserving doubling Algorithms 3.4 and 3.5 to compute the unique positive definite solution of the system (1.2). Two modified structure-preserving doubling algorithm both can quickly implement decoupling calculation of X and Y. Then, we consider a cyclic reduction Algorithm 4.3 and two modified cyclic reduction Algorithms 4.4 and 4.5 for finding the unique positive definite solution of the system (1.2). Similarly, two modified cyclic reduction algorithm both can quickly implement decoupling calculation of X and Y. Furthermore, we establish the convergence theory of Algorithms 3.3 and 4.3.

Acknowledgements

The authors deeply thank the anonymous referee for helping to improve the original manuscript by valuable suggestions.

References

[1] A. M. Al-Dubiban, *Iterative algorithm for solving a system of nonlinear matrix equations*, J. Appl. Math., 2012. DOI:10.1155/2012/461407.

[2] A. M. Al-Dubiban, *On the iterative method for the system of nonlinear matrix equations*, Abst. Appl. Anal., 2013. DOI:10.1155/2013/685753.

[3] W. N. Anderson, Jr., T.D. Morley, G.E. Trapp, *Positive solutions to* $X = A - BX^{-1}B^*$, Linear Algebra Appl., 1990, 134, 53–62.

[4] J. H. Bevis, F. J. Hall and R. E. Hartwing, *Consimilarity and the matrix equation* $A\bar{X} - XB = C$, Current Trends in Matrix Theory, North-Holland, New York, 1987, 51–64.

[5] J. H. Bevis, F. J. Hall and R. E. Hartwing, *The matrix equation* $A\bar{X} - XB = C$ *and its special cases*, SIAM J. Matrix Anal. Appl., 1998, 9, 348–359.

[6] J. Cai and G. Chen, *On the Hermitian positive definite solutions of nonlinear matrix equation* $X^s + A^*X^{-t}A = Q$, Appl. Math. Comput., 2010, 217, 117–123.

[7] C. Y. Chiang, Eric K. W. Chu and W. W. Lin, *On the H-Sylvester equation* $AX \pm X^*B^* = C$, Appl. Math. Comput., 2012, 218, 8393–8407.

[8] F. Ding and T. Chen, *On iterative solutions of general coupled matrix equations*, SIAM J. Control Optim., 2006, 44, 2269–2284.

[9] 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., 2008, 197, 41–50.

[10] F. Ding, *System Identification–new Theory and Methods*, Science Press, Beijing, 2013.

[11] X. F. Duan, C. Li and A. P. Liao, *Solutions and perturbation analysis for the nonlinear matrix equation* $X + \sum_{i=1}^{m} A_i^* X^{-1} A_i = I$, Appl. Math. Comput., 2011, 218, 4458–4466.

[12] S. M. El-sayed, *Two iteration processes for computing positive definite solutions of the equation* $X - A^*X^{-n}A = Q$, Comput. Math. Appl., 2001, 41, 579–588.

[13] S. M. El-sayed and A. M. Al-Dbiban, *A new inversion free iteration for solving the equation* $X + A^TX^{-1}A = Q$, J. Comput. Appl. Math., 2005, 181, 148–156.

[14] J. C. Engwerda, A. C. M. Ran and A. L. Rijkeboer, *Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation* $X + A^*X^{-1}A = Q$, Linear Algebra Appl., 1993, 186, 255–275.

[15] A. Ferrante and B. C. Levy, *Hermitian solution of the equation* $X = Q - NX^{-1}N^*$, Linear Algebra Appl., 1996, 247, 359–373.

[16] C. H. Guo and W. W. Lin, *The matrix equation* $X + A^TX^{-1}A = Q$ *and its application in nano research*, SIAM J. Sci. Comput., 2010, 32, 3020–3038.

[17] C. H. Guo, Y. C. Kuo and W. W. Lin, *Complex symmetric stabilizing solution of the matrix equation* $X + A^T X^{-1} A = Q$, Linear Algebra Appl., 2011, 435, 1187–1192.

[18] L. Guo, L. Liu, Y. Wu, *Existence of positive solutions for singular fractional differential equations with infinite-point boundary conditions*, Nonlinear Anal. Model. Control, 2015, 21, 635–650.

[19] M. Han, L. Sheng, X. Zhang, *Bifurcation theory for finitely smooth planar autonomous differential systems*, J. Differ. Equations, 2018, 264, 3596–3618.

[20] M. Han, X. Hou, L. Sheng, C. Wang, *Theory of rotated equations and applications to a population model*, Discrete Cont. Dyn. Syst. -A, 2018, 38, 2171–2185.

[21] V. Hasanov and I. G. Ivanov, *Solutions and perturbation estimates for the matrix equations* $X \pm A^* X^{-n} A = Q$, Appl Math Comput., 2004, 156, 513–525.

[22] N. Huang and C. F. Ma, *The structure-preserving doubling algorithms for positive definite solution to a system of nonlinear matrix equations*, Linear Multilinear Algebra, 2018, 66, 827–839.

[23] N. Huang and C. F. Ma, *Two structure-preserving-doubling like algorithms for obtaining the positive definite solution to a class of nonlinear matrix equation*, Comput. Math. Appl., 2015, 69, 494–502.

[24] I. G. Ivanov and S. M. El-sayed, *Properties of positive definite solutions of the equation* $X + A^* X^{-2} A = I$, Linear Algebra Appl., 1998, 279, 303–316.

[25] T. S. Jiang, C. H. Cheng and L. Chen, *An algebraic relation between consimilarity and similarity of complex matrices and its applications*, Physica A, 2006, 38, 9215–9222.

[26] T. S. Jiang and M. S. Wei, *On solutions of the matrix equations* $X - AXB = C$ *and* $X - A\bar{X}B = C$, Linear Algebra Appl., 2003, 367, 225–233.

[27] F. Li, G. Du, *General energy decay for a degenerate viscoelastic Petrovsky-type plate equation with boundary feedback*, J. Appl. Anal. Comput., 2018, 8, 390–401.

[28] M. Li, J. Wang, *Exploring delayed mittag-Leffler type matrix functions to study finite time stability of fractional delay differential equations*, Appl. Math. Comput., 2018, 324, 254–265.

[29] A. J. Liu, G. L. Chen, *On the Hermitian positive definite solutions of nonlinear matrix equation* $X^s + \sum_{i=1}^{m} A_i^* X^{-t_i} A_i = Q$, Appl. Math. Comput., 2014, 243, 950–959.

[30] A. J. Liu, G. L. Chen, X. Y. Zhang, *A new method for the bisymmetric minimum norm solution of the consistent matrix equations* $A_1XB_1 = C_1, A_2XB_2 = C_2$, J. Appl. Math., Vol. 2013, Article ID 125687, 6 pages.

[31] Z. Y. Li, B. Zhou and J. Lam, *Towards positive definite solutions of a class of nonlinear matrix equations*, Appl. Math. Comput., 2014, 237, 546–559.

[32] W. W. Lin and S. F. Xu, *Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl., 2006, 29, 26–39.

[33] B. Meini, *Efficient computation of the extreme solutions of* $X + A^* X^{-1} A = Q$ *and* $X - A^* X^{-1} A = Q$, Math. Comput., 2001, 71, 1189–1204.

[34] M. Monsalve and M. Raydan, *A new inversion-free method for a rational matrix equation*, Linear Algebra Appl., 2010, 433, 64–71.

[35] L. Ren, J. Xin, *Almost global existence for the Neumann problem of quasi-linear wave equations outside star-shaped domains in 3D*, Electron J. Differ. Equations, 2018, 31, 1–22.

[36] H. Tian, M. Han, *Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems*, J. Differ. Equations, 2017, 263, 7448–7474.

[37] B. Wang, F. Meng, Y. Fang, *Efficient implementation of RKN-type Fourier collocation methods for second-order differential equations*, Appl. Numer. Math., 2017, 119, 164–178.

[38] B. Wang, X. Wu, F. Meng, *Trigonometric collocation methods based on Lagrange basis polynomials for multi-frequency oscillatory second order differential equations*, J. Comput. Appl. Math., 2017, 313, 185–201.

[39] B. Wang, *Exponential Fourier collocation methods for solving first-order differential equations*, J. Comput. Appl. Math., 2017, 35, 711–736.

[40] S. F. Xu, *Matrix Computation in Control Theory (in Chinese)*, Higher Education Press, Beijing, 2011.

[41] X. Y. Yin and S. Y. Liu, *Positive definite solutions of the matrix equations* $X \pm A^* X^{-q} A = Q$ $(q \geq 1)$, Comput. Math. Appl., 2010, 59, 3727–3739.

[42] B. Zhou, G. B. Cai and J. Lam, *Positive definite solutions of the nonlinear matrix equation* $X + A^H \bar{X}^{-1} A = I$, Appl. Math. Comput., 2013, 219, 7377–7391.

[43] B. Zhou, G. R. Duan and Z. Li, *Gradient based iterative algorithm for solving coupled matrix equations*, Syst. Control Lett., 2009, 58, 327–333.

[44] B. Zhou, J. Lam and G. Duan, *On Smith-type iterative algorithms for the Stein matrix equation*, Appl. Math. Lett., 2009, 22, 1038–1044.

[45] B. Zhou, J. Lam and G. Duan, *Toward solution of matrix equation* $X = Af(X) + B$, Linear Algebra Appl., 2011, 435, 1370–1398.