J. Math. Study
doi: 10.4208/jms.v48n1.15.04

Vol. **48**, No. 1, pp. 53-65
March 2015

# Numerical Methods to Solve the Complex Symmetric Stabilizing Solution of the Complex Matrix Equation $X+A^TX^{-1}A=Q$

Yao Yao, Xiao-Xia Guo*

*School of Mathematical Sciences, Ocean University of China, Qingdao 266100, Shandong, P.R. China.*

Received 6 December, 2014; Accepted (in revised version) 15 January, 2015

**Abstract.** When the matrices $A$ and $Q$ have special structure, the structure-preserving algorithm was used to compute the stabilizing solution of the complex matrix equation $X+A^TX^{-1}A=Q$. In this paper, we study the numerical methods to solve the complex symmetric stabilizing solution of the general matrix equation $X+A^TX^{-1}A=Q$. We not only establish the global convergence for the methods under an assumption, but also show the feasibility and effectiveness of them by numerical experiments.

**AMS subject classifications**: 65F10, 65F35, 65F50

**Key words**: Complex matrix, complex symmetric stabilizing solution, fixed-point method, structure-preserving algorithm.

## 1 Introduction

The nonlinear matrix equation $X+A^TX^{-1}A=Q$, where $A$ is real and $Q$ is symmetric positive definite, arises in several applications, such as the analysis of ladder network, dynamic programming, the Green's function in nano research, control theory and stochastic filtering. These equations have been studied in [5,6], for example.

Recently, there arises the need to consider the matrix equation

$$X+A^TX^{-1}A=Q, \tag{1.1}$$

where $A$ is complex and $Q$ is complex symmetric. First, it is explained in [2] that the computation of the surface Green's function in nano research [7] can be reduced to the problem of solving the matrix equation (1.1), where $Q=Q_1+i\eta I$ with $Q_1$ real symmetric and $\eta$ positive scalar, but the matrix $A$ is still a real matrix. And then it is shown in [4] that a quadratic eigenvalue problem arising from the vibration analysis of fast trains can

*Corresponding author. *Email addresses:* guoxiaoxia@ouc.edu.cn (X.-X. Guo), 877133678@qq.com (Y. Yao)

 

be solved efficiently and accurately by solving a matrix equation of the form (1.1), where $A$ is complex and $Q$ is complex symmetric. Moreover, the matrix $A$ has only one nonzero block in the upper-right corner, and $Q$ is block tridiagonal and block Toeplitz. In those two applications, the existence of a unique complex symmetric stabilizing solution has been proved using advanced results on linear operators. The fixed-point method and doubling algorithm were given to solve the stabilizing solution of the matrix equation (1.1).

For the more general complex equation (1.1), the existence of a unique complex symmetric stabilizing solution has been proved in [1]. However, the corresponding numerical experiments were not given. In this paper, according to the idea proposed in [1], we mainly discuss the numerical algorithms to solve the stabilizing solution of this equation. In Section 2, we introduce the preliminaries of the complex matrix equation (1.1). In Section 3, the fixed-point method (FPI), modified fixed-point method (MFPI) and structure-preserving algorithm (SPA) are proposed to find the complex symmetric stabilizing solution of (1.1) and their convergence are analyzed under an assumption. In Section 4, numerical examples are given to show the feasibility and effectiveness of the FPI, MFPI and SPA methods, and concluding remarks are made in Section 5.

## 2 Preliminaries

For equation (1.1) we write:

$$\begin{aligned} &A=A_1+iA_2, \quad Q=Q_1+iQ_2, \\ &A_1,A_2\in\mathbb{R}^{n\times n}, \quad Q_1=Q_1^T, \quad Q_2=Q_2^T\in\mathbb{R}^{n\times n}. \end{aligned} \tag{2.1}$$

**Definition 2.1.** We define that

(a) a solution $X$ of (1.1) is said to be stabilizing if $\rho(X^{-1}A)<1$, where $\rho(\cdot)$ denotes the spectral radius;

(b) $W>0$ denotes the positive definiteness of a Hermitian matrix $W$.

The following theorem is given by [1].

**Theorem 2.1.** ([1]) *If the matrices $A_2$ and $Q_2$ satisfy that*

$$Q_2+e^{i\theta}A_2^T+e^{-i\theta}A_2>0,\theta\in[0,2\pi], \tag{2.2}$$

*then the equation* (1.1) *has a stabilizing solution.*

We suppose the inequality (2.2) holds throughout this paper. Obviously, if a positive semi-definite matrix is added to $Q_2$, it still holds. Let

$$M_0=\begin{bmatrix} A & 0 \\ Q & -I \end{bmatrix}, \quad L_0=\begin{bmatrix} 0 & I \\ A^T & 0 \end{bmatrix}. \tag{2.3}$$

It's easily seen that the matrix pair $(M_0,L_0)$ satisfies the relation:

$$M_0 J M_0^T = L_0 J L_0^T,$$

where

$$J = \begin{bmatrix} O & I \\ -I & 0 \end{bmatrix}.$$

Then the matrix pair $(M_0,L_0)$ or the matrix pencil $M_0-\lambda L_0$ is called $T$-symplectic.

Since the $M_0-\lambda L_0$ has no eigenvalues on the unit circle (see [1, lemma 1]), then there is a matrix $\begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{C}^{2n\times n}$ of full rank spanning the stable invariant subspace of $M_0-\lambda L_0$ corresponding to the stable eigenvalue matrix $S\in\mathbb{C}^{n\times n}$, i.e.,

$$M_0 \begin{bmatrix} U \\ V \end{bmatrix} = L_0 \begin{bmatrix} U \\ V \end{bmatrix} S, \tag{2.4}$$

where $\rho(S)<1$, and the matrix $U$ is invertible. Further, by Theorem 3 in [1], we have the following theorem.

**Theorem 2.2.** *Let $X_s = VU^{-1}$. Then*

*(a) $X_s$ is complex symmetric;*

*(b) $X_s$ is invertible;*

*(c) $X_s$ is a stabilizing solution of* (1.1)*;*

*(d) $X_{s,2} = Im(X_s)$ is positive definite.*

**Theorem 2.3.** *(Bendixson's theorem) if $X$ and $Y$ are Hermitian $n\times n$ matrices with eigenvalues*

$$\xi_1 \le \xi_2 \le \cdots \le \xi_n, \quad \eta_1 \le \eta_2 \le \cdots \le \eta_n,$$

*then every eigenvalue $\lambda$ of $X+iY$ is contained in the rectangle*

$$\xi_1 \le Re(\lambda) \le \xi_n, \quad \eta_1 \le Im(\lambda) \le \eta_n.$$

## 3 The numerical methods for the equation (1.1)

In this section, we introduce the fixed-point method (FPI), modified fixed-point method (MFPI) and structure-preserving algorithm (SPA) to solve the complex symmetric stabilizing solution of the matrix equation (1.1). Then we give the feasibility analysis of the FPI and the convergence analysis of the SPA, respectively.

**Algorithm 1.** (The fixed-point iteration method (FPI))

$$X_0 = Q,$$
$$X_{k+1} = Q - A^T X_k^{-1} A, \quad k = 0,1,2,\cdots.$$

**Theorem 3.1.** *Let $A$ and $Q$ be as in* (2.1)*. The sequence $\{X_k\}$ generated by Algorithm 1 is well-defined, and $\{X_k\}$ is complex symmetric.*

*Proof.* We write $T_k$ be the block $k\times k$ ($k\geq 1$) matrix given by

$$\begin{aligned} T_k &= \begin{bmatrix} Q & -A^T & & \\ -A & Q & \ddots & \\ & \ddots & \ddots & -A^T \\ & & -A & Q \end{bmatrix} \\ &= \begin{bmatrix} Q_1 & -A_1^T & & \\ -A_1 & Q_1 & \ddots & \\ & \ddots & \ddots & -A_1^T \\ & & -A_1 & Q_1 \end{bmatrix} + i \begin{bmatrix} Q_2 & -A_2^T & & \\ -A_2 & Q_2 & \ddots & \\ & \ddots & \ddots & -A_2^T \\ & & -A_2 & Q_2 \end{bmatrix}. \end{aligned}$$

Let

$$C_k = \begin{bmatrix} Q_1 & -A_1^T & & \\ -A_1 & Q_1 & \ddots & \\ & \ddots & \ddots & -A_1^T \\ & & -A_1 & Q_1 \end{bmatrix}, \; D_k = \begin{bmatrix} Q_2 & -A_2^T & & \\ -A_2 & Q_2 & \ddots & \\ & \ddots & \ddots & -A_2^T \\ & & -A_2 & Q_2 \end{bmatrix}. \tag{3.1}$$

Then for each $k\geq 1$ we can write

$$T_k = C_k + iD_k. \tag{3.2}$$

Note that $Q_2 + e^{i\theta} A_2^T + e^{-i\theta} A_2 > 0$ for all $\theta \in [0,2\pi]$ is equivalent to that $D_k$ is positive definite. It follows from Theorem 2.3 (Bendixson's theorem) that $T_k$ is invertible. By the block Gaussian elimination performed on the matrix

$$T = \begin{bmatrix} Q & -A^T & & \\ -A & Q & -A^T & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix} \tag{3.3}$$

We can obtain the sequence $\{X_k\}$. In fact, $X_0 = Q$ is the (1,1) block in (3.3); when the (1,1) block is used to eliminate the (2,1) block, the new (2,2) block is $X_1$; when the new (2,2) block is used to eliminate the (3,2) block, the new (3,3) block is $X_2$; and so on. Because $T_k$ is invertible for each $k \geq 1$, $\{X_k\}$ is well-defined and invertible for each $k \geq 0$. $Q$ is complex symmetric, i.e. $X_0$ is complex symmetric. We can suppose that $X_k$ is complex symmetric. It can obtains that

$$X_{k+1}^T = Q^T - A^T X_k^{-T} A = Q - A^T X_k^{-1} A = X_{k+1}.$$

$X_{k+1}$ is complex symmetric. $\{X_k\}$ is thus complex symmetric. □

When $\rho(X_s^{-1}A) \approx 1$, the convergence of Algorithm 1 will be very slow in general. The strategy proposed in [14] for improving the convergence of Algorithm 1 generates the following modified fixed-point method (MFPI).

**Algorithm 2.** (The modified fixed-point iteration method (MFPI))

$$\begin{aligned}
&X_0 = Q,\\
&X_{k+1} = Q - A^T X_k^{-1} A,\\
&X_{k+1} = (X_k + X_{k+1})/2, k = 0,1,2,\cdots.
\end{aligned}$$

Numerical experiments will show that the convergence of Algorithm 2 is often much faster than that of Algorithm 1. However, a rigorous convergence analysis remains an open problem.

Let $M_0$ and $L_0$ be as given in (2.3), then we have the following algorithm.

**Algorithm 3.** (The structure-preserving algorithm (SPA)) Let $A_0 = A, Q_0 = Q, P_0 = 0$. For $k = 0,1,2,\cdots$, compute

$$\begin{aligned}
&A_{k+1} = A_k (Q_k - P_k)^{-1} A_k,\\
&Q_{k+1} = Q_k - A_k^T (Q_k - P_k)^{-1} A_k,\\
&P_{k+1} = P_k + A_k (Q_k - P_k)^{-1} A_k^T.
\end{aligned}$$

We will show that the SPA will not break down, and $Q_k$ converges to $X_s$ quickly.

**Lemma 3.1.** *Let $A$ and $Q$ be as in (2.1), and the sequences $\{A_k\}$, $\{Q_k\}$ and $\{P_k\}$ be generated by the SPA. Let $W_k = Q_k - P_k$, where $W_0 = Q$, $k \geq 0$. If $T_k[-A^T, Q, -A]$ is an $k \times k$ block tridiagonal and invertible matrix having the structure given in (3.3), then $W_k$ is nonsingular.*

*Proof.* Proceed by induction. Since $W_k = Q_k - P_k$, where $W_0 = Q$, $k \geq 1$. We suppose that the sequence $\{W_k\}$ satisfies:

$$\begin{aligned} W_{k+1} &= Q_{k+1} - P_{k+1} \\ &= Q_k - A_k^T (Q_k - P_k)^{-1} A_k - P_k - A_k (Q_k - P_k)^{-1} A_k^T \\ &= W_k - A_k^T W_k^{-1} A_k - A_k W_k^{-1} A_k^T. \end{aligned}$$

For $k=0$, we apply the even-odd permutation of block rows and columns of $T_3[-A^T, Q, -A]$ and obtain the matrix

$$\begin{bmatrix} W_0 & 0 & -A^T \\ 0 & W_0 & -A \\ -A & -A^T & W_0 \end{bmatrix} = \begin{bmatrix} G_2[W_0] & F_1[-A^T, -A] \\ E_1[-A, -A^T] & G_1[W_0] \end{bmatrix},$$

where $G_j[W]$ is the $j \times j$ block diagonal matrix with diagonal blocks equal to $W$, $F_j[C,R]$ is the $(j+1) \times j$ block lower bidiagonal matrix having $C$ on the main diagonal and $R$ on the lower diagonal, and $E_j[C,R]$ is the $j \times (j+1)$ block upper bidiagonal matrix having $R$ on the main diagonal and $C$ on the upper diagonal. For convenience, we denote the matrix $G_2[W_0]$, $F_1[-A^T, -A]$, $E_1[-A, -A^T]$ and $G_1[W_0]$ as $G_2$, $F_1$, $E_1$ and $G_1$, respectively.

Since $W_0 = Q$ is nonsingular, so the matrix $G_2 = \begin{bmatrix} W_0 & 0 \\ 0 & W_0 \end{bmatrix}$ is nonsingular. Using one step of block Gaussian elimination to the above permuted matrix we can get

$$\begin{bmatrix} G_2 & F_1 \\ E_1 & G_1 \end{bmatrix} \begin{bmatrix} I & -G_2^{-1} F_1 \\ 0 & I \end{bmatrix} = \begin{bmatrix} G_2 & 0 \\ E_1 & G_1 - E_1 G_2^{-1} F_1 \end{bmatrix}.$$

It is easily seen that

$$\begin{bmatrix} G_2 & F_1 \\ E_1 & G_1 \end{bmatrix} = \begin{bmatrix} G_2 & 0 \\ E_1 & G_1 - E_1 G_2^{-1} F_1 \end{bmatrix} \begin{bmatrix} I & G_2^{-1} F_1 \\ 0 & I \end{bmatrix}.$$

Thus,

$$|G_1 - E_1 G_2^{-1} F_1| = \frac{|T_3[-A^T, Q, -A]|}{|G_2|}.$$

Since $T_3[-A^T, Q, -A]$ and $G_2$ are invertible, the matrix $G_1 - E_1 G_2^{-1} F_1$ is nonsingular. Obviously, the $W_1$ is nonsingular, which can be expressed by

$$\begin{aligned} W_1 &= G_1 - E_1 G_2^{-1} F_1 \\ &= W_0 - [-A, -A^T] \begin{bmatrix} W_0 & 0 \\ 0 & W_0 \end{bmatrix}^{-1} \begin{bmatrix} -A^T \\ -A \end{bmatrix} \\ &= W_0 - A^T W_0^{-1} A - A W_0^{-1} A^T, \end{aligned}$$

where $A=A_0$.

Next, considering the $k$ case, we assume that $W_i\,(i=1,\ldots,k-1)$ is nonsingular.

Applying the even-odd permutation of block rows and columns to the matrix $T_{2^{k+1}-1}[-A_0^T,Q_0,-A_0]$ yields

$$\begin{bmatrix} G_{2^k}[Q_0] & F_{2^k-1}[-A_0^T,-A_0] \\ E_{2^k-1}[-A_0,-A_0^T] & G_{2^k-1}[Q_0] \end{bmatrix}.$$

After performing one step of Gaussian elimination we obtain the matrix

$$\begin{aligned}&T_{2^k-1}[-A_1^T,Q_1,-A_1]\\ =&G_{2^k-1}[Q_0]-E_{2^k-1}[-A_0,-A_0^T]G_{2^k}[Q_0^{-1}]F_{2^k-1}[-A_0^T,-A_0].\end{aligned}$$

By the properties of the Schur complement it follows that if $Q_0$ and $T_{2^{k+1}-1}[-A_0^T,Q_0,-A_0]$ are nonsingular, then $T_{2^k-1}[-A_1^T,Q_1,-A_1]$ is nonsingular. From the inductive hypothesis, assuming $W_i\,(i=1,\ldots,k-1)$ nonsingular, then the $k$th step of cyclic reduction can be performed, starting with blocks $-A_1^T,Q_1,-A_1$, i.e., $W_k$ is nonsingular for each $k\geq 0$. □

**Theorem 3.2.** *Let $A$ and $Q$ be as in* (2.1). *Let $X_s$ be the stabilizing solution of* (1.1) *and $Y_s$ be the stabilizing solution of the dual equation $Y+AY^{-1}A^T=Q$ (The existence of $Y_s$ is also guaranteed by the argument leading to Theorem* 2.1*). Then*

*(a) The sequences $\{A_k\}$, $\{Q_k\}$ and $\{P_k\}$ generated by Algorithm 3 are well-defined. Moreover, $Q_k$ and $P_k$ are complex symmetric;*

*(b) $Q_k$ converges to $X_s$ quadratically, $A_k$ converges to 0 quadratically, $Q-P_k$ converges to $Y_s$ quadratically, with*

$$\begin{aligned}&\limsup_{k\to\infty}\sqrt[2^k]{\|Q_k-X_s\|}\leq(\rho(X_s^{-1}A))^2,\quad \limsup_{k\to\infty}\sqrt[2^k]{\|A_k\|}\leq\rho(X_s^{-1}A),\\ &\limsup_{k\to\infty}\sqrt[2^k]{\|Q-P_k-Y_s\|}\leq(\rho(X_s^{-1}A))^2,\end{aligned}$$

*where $\|\cdot\|$ is any matrix norm.*

*Proof.* (a) From (3.2), we can know that $T_k=C_k+iD_k$ for each $k\geq 1$, where $C_k$ is Hermitian matrix and $D_k$ is positive definite Hermitian matrix. It follows from Theorem 2.3 (Bendixson's theorem) that $T_k$ is invertible.

Let $W_k=Q_k-P_k$, where $W_0=Q$, $k\geq 0$. Then $W_k$ is nonsingular for each $k\geq 0$ from Lemma 3.1. The sequences $\{A_k\}$, $\{Q_k\}$ and $\{P_k\}$ in Algorithm 3 are well-defined. $Q_k$ and $P_k$ are complex symmetric because $Q$ is complex symmetric.

(b) $X_s$ be the stabilizing solution of (1.1) if and only if

$$M_0\begin{bmatrix} I \\ X_s \end{bmatrix}=L_0\begin{bmatrix} I \\ X_s \end{bmatrix}X_s^{-1}A. \tag{3.4}$$

We now define the sequences $\{M_k\}$ and $\{L_k\}$, where

$$M_k = \begin{bmatrix} A_k & 0 \\ Q_k & -I \end{bmatrix}, \quad L_k = \begin{bmatrix} -P_k & I \\ A_k^T & 0 \end{bmatrix}. \tag{3.5}$$

$W_k = Q_k - P_k$ is nonsingular for each $k \geq 0$, we can define the following matrix

$$\widetilde{M}_k = \begin{bmatrix} A_k(Q_k - P_k)^{-1} & 0 \\ -A_k^T(Q_k - P_k)^{-1} & I \end{bmatrix}, \quad \widetilde{L}_k = \begin{bmatrix} I & -A_k(Q_k - P_k)^{-1} \\ 0 & A_k^T(Q_k - P_k)^{-1} \end{bmatrix},$$

and we also know that $\widetilde{M}_k L_k = \widetilde{L}_k M_k (k \geq 0)$. By computing $\widetilde{L}_k L_k$ and $\widetilde{M}_k M_k$ $(k \geq 0)$, gives

$$\begin{aligned} \widetilde{L}_k L_k &= \begin{bmatrix} -(P_k + A_k(Q_k - P_k)^{-1} A_k^T) & I \\ A_k^T(Q_k - P_k)^{-1} A_k^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} -P_{k+1} & I \\ A_{k+1}^T & 0 \end{bmatrix} = L_{k+1}, \end{aligned}$$

$$\begin{aligned} \widetilde{M}_k M_k &= \begin{bmatrix} A_k(Q_k - P_k)^{-1} A_k & 0 \\ Q_k - A_k^T(Q_k - P_k)^{-1} A_k & -I \end{bmatrix} \\ &= \begin{bmatrix} A_{k+1} & 0 \\ Q_{k+1} & -I \end{bmatrix} = M_{k+1}. \end{aligned}$$

Premultiplying (3.4) with $\widetilde{M}_0$, and using $\widetilde{M}_0 L_0 = \widetilde{L}_0 M_0$, $M_1 = \widetilde{M}_0 M_0$, $L_1 = \widetilde{L}_0 L_0$, we get that

$$\widetilde{M}_0 M_0 \begin{bmatrix} I \\ X_s \end{bmatrix} = \widetilde{M}_0 L_0 \begin{bmatrix} I \\ X_s \end{bmatrix} X_s^{-1} A,$$

$$\begin{aligned} M_1 \begin{bmatrix} I \\ X_s \end{bmatrix} &= \widetilde{L}_0 M_0 \begin{bmatrix} I \\ X_s \end{bmatrix} X_s^{-1} A = \widetilde{L}_0 \left(L_0 \begin{bmatrix} I \\ X_s \end{bmatrix} X_s^{-1} A\right) X_s^{-1} A \\ &= L_1 \begin{bmatrix} I \\ X_s \end{bmatrix} (X_s^{-1} A)^2. \end{aligned}$$

So for each $k \geq 0$, we can know that

$$M_k \begin{bmatrix} I \\ X_s \end{bmatrix} = L_k \begin{bmatrix} I \\ X_s \end{bmatrix} (X_s^{-1} A)^{2^k}. \tag{3.6}$$

Substituting $M_k$ and $L_k$ into (3.6) yields

$$A_k = (X_s - P_k)(X_s^{-1} A)^{2^k}, \quad Q_k - X_s = A_k^T (X_s^{-1} A)^{2^k}. \tag{3.7}$$

Similarly we have

$$\hat{M}_0\begin{bmatrix} I \\ Y_s \end{bmatrix} = \hat{L}_0\begin{bmatrix} I \\ Y_s \end{bmatrix} Y_s^{-1}A^T,$$

where

$$\hat{M}_0 = \begin{bmatrix} A^T & 0 \\ Q & -I \end{bmatrix}, \quad \hat{L}_0 = \begin{bmatrix} 0 & I \\ A & 0 \end{bmatrix}.$$

We also know that $M_0 - \lambda L_0 = \begin{bmatrix} A & -\lambda I \\ Q-\lambda A^T & -I \end{bmatrix}$, so

$$\begin{bmatrix} A & -\lambda I \\ Q-\lambda A^T & -I \end{bmatrix}\begin{bmatrix} I & 0 \\ -(-I)^{-1}(Q-\lambda A^T) & I \end{bmatrix} = \begin{bmatrix} \lambda^2 A^T - \lambda Q + A & -\lambda I \\ 0 & -I \end{bmatrix}.$$

Taking the determinant on the two sides we obtain

$$|M_0 - \lambda L_0| = |\lambda^2 A^T - \lambda Q + A||-I| = (-1)^n|\lambda^2 A^T - \lambda Q + A|.$$

It follows that $M_0 - \lambda L_0$ has the same eigenvalues as $\lambda^2 A^T - \lambda Q + A$. Similarly, $\hat{M}_0 - \lambda\hat{L}_0$ has the same eigenvalues as $\lambda^2 A - \lambda Q + A^T$ $((\lambda^2 A^T - \lambda Q + A)^T = \lambda^2 A - \lambda Q + A^T)$. Then $X_s^{-1}A$ and $Y_s^{-1}A^T$ have the same eigenvalues, and $\rho(X_s^{-1}A) = \rho(Y_s^{-1}A^T)$. For each $k \geq 0$, we have

$$\hat{M}_k\begin{bmatrix} I \\ Y_s \end{bmatrix} = \hat{L}_k\begin{bmatrix} I \\ Y_s \end{bmatrix}(Y_s^{-1}A^T)^{2^k}, \tag{3.8}$$

where

$$\hat{M}_k = \begin{bmatrix} A_k^T & 0 \\ \hat{Q}_k & -I \end{bmatrix}, \quad \hat{L}_k = \begin{bmatrix} -\hat{P}_k & I \\ A_k & 0 \end{bmatrix}, \quad \hat{P}_k = Q - Q_k, \quad \hat{Q}_k = Q - P_k.$$

Substituting $\hat{M}_k$ and $\hat{L}_k$ into (3.8) yields

$$A_k^T = (Y_s - \hat{P}_k)(Y_s^{-1}A^T)^{2^k}, \quad \hat{Q}_k - Y_s = A_k(Y_s^{-1}A^T)^{2^k}. \tag{3.9}$$

It follows from (3.7)-(3.9) that

$$\begin{aligned} Q_k - X_s &= A_k^T(X_s^{-1}A)^{2^k} \\ &= (Y_s - \hat{P}_k)(Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k} \\ &= (Q_k - X_s + (X_s + Y_s - Q))(Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k}. \end{aligned}$$

Consequently

$$\begin{aligned} &(Q_k - X_s) - (Q_k - X_s)(Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k} = (X_s + Y_s - Q)(Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k}, \\ &(Q_k - X_s)(I - (Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k}) = (X_s + Y_s - Q)(Y_s^{-1}A^T)^{2^k}(X_s^{-1}A)^{2^k}. \end{aligned}$$

It follows that

$$\limsup_{k\to\propto} \sqrt[2^k]{\| Q_k - X_s \|} \le \rho(X_s^{-1}A)\rho(Y_s^{-1}A^T) = (\rho(X_s^{-1}A))^2 < 1.$$

So $Q_k$ converges to $X_s$ quadratically. Since $\hat{P}_k = Q - Q_k$ and $\{Q_k\}$ is bounded, then $\{\hat{P}_k\}$ is bounded. By (3.9), we know

$$\limsup_{k\to\propto} \sqrt[2^k]{\| A_k \|} \le \rho(Y_s^{-1}A^T) = \rho(X_s^{-1}A) < 1.$$

Thus $A_k$ converges to 0 quadratically. By $\hat{Q}_k - Y_s = A_k(Y_s^{-1}A^T)^{2^k}$ in (3.9) and (3.8) we have

$$\limsup_{k\to\propto} \sqrt[2^k]{\| \hat{Q}_k - Y_s \|} = \limsup_{k\to\propto} \sqrt[2^k]{\| Q - P_k - Y_s \|} \le (\rho(X_s^{-1}A))^2 < 1.$$

So $Q - P_k$ converges to $Y_s$ quadratically. □

The SPA is said to be structure-preserving since for each $k \ge 0$, $M_k$ and $L_k$ have the structures given in (3.5), and the pencil $M_k - \lambda L_k$ is $T$-symplectic.

# 4 Numerical experiments

In this section we present some numerical results to illustrate the convergence behavior of the algorithms for computing the stabilizing solution $X_s$ of the equation (1.1). We use the relative residual (denoted as "RES")

$$\text{RES} = \frac{\| X + A^T X^{-1} A - Q \|}{\| X \| + \| A \|^2 \| X^{-1} \| + \| Q \|},$$

where $\| \cdot \|$ is the spectral norm.

In our implementations, all iterations are terminated when the current iterate satisfies $\|X_{k+1} - X_k\| < 10^{-10}$. The numerical experiments were done in Matlab R2010a with respect to the initial value ($X_0 = Q$), the numbers of iterations (denoted as "IT"), the CPU time in seconds.

**Example 4.1.** Consider the matrix equation $X + A^T X^{-1} A = Q_1 + i\eta I$, where $A \in \mathbb{R}^{n\times n}$, $Q_1 = Q_1^T \in \mathbb{R}^{n\times n}$, $A$ and $Q$ are generated randomly and $\eta \ge 0$. We will take $n = 16, 32, 64, 128$, and $\eta = \frac{1}{4}, \frac{1}{2}, 1$, respectively. The numerical results are shown in Table 1, 2, 3.

**Example 4.2.** Consider the matrix equation $X + A^T X^{-1} A = Q$, where $A$ and $Q$ are given by (2.1), and $Q_2 + A_2 > 0$. Moreover, $A_1$, $Q_1$, $A_2$ and $Q_2$ are generated randomly. In this example, we take $n = 16, 32, 64$, and 128, respectively. The numerical results are shown in Table 4.

Table 1: The numerical results for example 1 ($\eta=\frac{1}{4}$).

| Method | | n | | | |
|---|---|---|---|---|---|
| | | 16 | 32 | 64 | 128 |
| FPI | IT | 349 | 431 | 494 | 809 |
| | CPU | 1.152 | 3.436 | 11.445 | 76.475 |
| | RES | 2.67e-14 | 4.20e-015 | 5.79e-16 | 6.74e-17 |
| MFPI | IT | 199 | 238 | 245 | 331 |
| | CPU | 0.569 | 0.573 | 4.109 | 14.475 |
| | RES | 2.27e-14 | 4.16e-15 | 4.62e-16 | 6.31e-17 |
| SPA | IT | 11 | 20 | 23 | 32 |
| | CPU | 0.054 | 0.092 | 0.753 | 2.376 |
| | RES | 2.08e-16 | 3.50e-16 | 1.86e-16 | 1.44e-17 |

Table 2: The numerical results for example 1 ($\eta=\frac{1}{2}$).

| Method | | n | | | |
|---|---|---|---|---|---|
| | | 16 | 32 | 64 | 128 |
| FPI | IT | 141 | 228 | 280 | 348 |
| | CPU | 0.427 | 2.147 | 6.776 | 33.656 |
| | RES | 3.32e-14 | 3.45e-015 | 5.48e-16 | 6.98e-17 |
| MFPI | IT | 96 | 144 | 173 | 229 |
| | CPU | 0.352 | 0.639 | 1.503 | 4.028 |
| | RES | 3.11e-14 | 3.30e-15 | 5.15e-16 | 5.31e-17 |
| SPA | IT | 9 | 18 | 21 | 30 |
| | CPU | 0.041 | 0.111 | 0.648 | 2.104 |
| | RES | 5.61e-17 | 5.54e-17 | 1.67e-17 | 1.25e-17 |

Numerical results in Tables $1-3$ show that the effects of the FPI, MFPI and SPA methods become more effective with the increase value $\eta$, since the value of $\rho(X_s^{-1}A)$ is close to 1 with small $\eta$. From Tables $1-4$, we observe that the FPI, MFPI and SPA methods are feasible to compute the stabilizing solution of (1.1). More specifically, it can also see that CPU(SPA)$<$ CPU(MFPI)$<$CPU(FPI), IT(SPA)$<$IT(MFPI)$<$IT(FPI), RES(SPA)$<$RES(MFPI) $<$RES (MFPI). This indicates that the SPA is more efficient than the MFPI and FPI.

# 5 Conclusion

In this paper, we present the fixed-point iteration (FPI), the modified fixed-point iteration (MFPI) and the structure-preserving algorithm (SPA) for computing the stabilizing solu-

Table 3: The numerical results for example 1 ($\eta$=1).

| Method | | n | | | |
|---|---|---|---|---|---|
| | | 16 | 32 | 64 | 128 |
| FPI | IT | 134 | 202 | 230 | 275 |
| | CPU | 0.590 | 0.714 | 3.875 | 26.773 |
| | RES | 4.76e-14 | 3.94e-014 | 4.28e-16 | 6.84e-17 |
| MFPI | IT | 79 | 80 | 89 | 121 |
| | CPU | 0.246 | 0.354 | 1.499 | 3.385 |
| | RES | 2.86e-14 | 2.83e-14 | 3.85e-16 | 4.54e-17 |
| SPA | IT | 7 | 13 | 18 | 24 |
| | CPU | 0.024 | 0.070 | 0.519 | 2.072 |
| | RES | 5.13e-17 | 2.37e-17 | 1.71e-17 | 1.32e-17 |

Table 4: The numerical results for example 2.

| Method | | n | | | |
|---|---|---|---|---|---|
| | | 16 | 32 | 64 | 128 |
| FPI | IT | 617 | 1000 | 1000 | 1000 |
| | CPU | 1.547 | 6.159 | 23.312 | 111.304 |
| | RES | 8.26e-15 | 2.11e-15 | 5.42e-04 | 1.89e-05 |
| MFPI | IT | 276 | 821 | 1000 | 1000 |
| | CPU | 0.793 | 5.687 | 22.948 | 110.977 |
| | RES | 3.36e-15 | 2.08e-15 | 1.12e-04 | 1.51e-05 |
| SPA | IT | 12 | 17 | 19 | 38 |
| | CPU | 0.052 | 0.261 | 15.930 | 28.086 |
| | RES | 2.14e-15 | 1.27e-15 | 1.62e-14 | 1.07e-12 |

tion of (1.1). Different from the reference [1], numerical experiments are given to show the feasibility and effectiveness of the FPI, MFPI and SPA methods.

# Acknowledgments

This research was supported by NSFC (No. 11071228) and the Shandong Province Young Scientists Research Awards Fund (BS2010SF032).

**References**

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

[2] 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.
[3] C.-H. Guo, Y.-C. Kuo, and W.-W. Lin. On a nonlinear matrix equation arising in nano research. SIAM J. Matrix Anal. Appl., 2012, 33: 235–262.
[4] C.-H. Guo and W.-W. Lin. Solving a structured quadratic eigenvalue problem by a structure-preserving doubling algorithm. SIAM J. Matrix Anal. Appl., 2010, 31: 2784–2801.
[5] C.-Y. Chiang, E. K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, and S.-F. Xu. Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. SIAM J. Matrix Anal. Appl., 2009, 31: 227–247.
[6] 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.
[7] S. Datta. Nanoscale device modeling: the Green's function method. Superlattices and Microstructures, 2000, 28: 253–278.
[8] D. A. Bini, L. Gemignani, and B. Meini. Computations with infinite Toeplitz matrices and polynomials. Linear Algebra Appl., 2002, 343-344: 21–61.
[9] B. Meini. Efficient computation of the extreme solutions of $X+A^*X^{-1}A=Q$ and $X-A^*X^{-1}A=Q$. Math. Comp., 2002, 71: 1189–1204.
[10] 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, 28: 26–39.
[11] E. K.-W. Chu, H.-Y. Fan, and W.-W. Lin. A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations. Linear Algebra Appl., 2005, 396: 55–58.
[12] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Structured polynomial eigenvalue problems: good vibrations from good linearizations. SIAM J. Matrix Anal. Appl., 2006, 28: 1029–1051.
[13] X. Zhan. Computing the extremal positive definite solutions of a matrix equation. SIAM J. Sci. Comput., 1996, 17: 1167–1174.
[14] J. Tomfohr and O. F. Sankey. Theorelical analysis of electron transport through organic molecules. Journal of Chemical Physics, 2004, 120: 1542–1554.