

## Research Article

# Newton's Method for the Matrix Nonsingular Square Root

Chun-Mei Li<sup>1</sup> and Shu-Qian Shen<sup>2</sup>

<sup>1</sup> Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

<sup>2</sup> College of Science, China University of Petroleum, Qingdao, Shandong 266580, China

Correspondence should be addressed to Chun-Mei Li; lengyuel23@126.com

Received 14 June 2014; Accepted 10 August 2014; Published 31 August 2014

Academic Editor: Shi-Liang Wu

Copyright © 2014 C.-M. Li and S.-Q. Shen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Two new algorithms are proposed to compute the nonsingular square root of a matrix  $A$ . Convergence theorems and stability analysis for these new algorithms are given. Numerical results show that these new algorithms are feasible and effective.

## 1. Introduction

Consider the following nonlinear matrix equation:

$$F(X) = X^2 - A = 0, \quad (1)$$

where  $A$  is an  $n \times n$  nonsingular complex matrix. A solution  $X$  of (1) is called a square root of  $A$ . The matrix square roots have many applications in the boundary value problems [1] and the computation of the matrix logarithm [2, 3].

In the last few years there has been a constantly increasing interest in developing the theory and numerical methods for the matrix square roots. The existence and uniqueness of the matrix square root can be found in [4–6]. Here, it is worthwhile to point out that any nonsingular matrix has a square root, and the square root is also nonsingular [4]. A number of methods have been proposed for computing square root of a matrix [5, 7–16]. The computational methods for the matrix square root can be generally separated into two classes. The first class is the so-called direct methods, for example, Schur algorithm developed by Björck and Hammarling [7]. The second class is the iterative methods. Matrix iterations  $X_{k+1} = f(X_k)$ , where  $f$  is a polynomial or a rational function, are attractive alternatives for computing square roots [9, 11–13, 15, 17]. A well-known iterative method for computing matrix square root is Newton's method. It has nice numerical behavior, for example, quadratic convergence. Newton's method for solving (1) was proposed in [18]. Later, some simplified Newton's methods were developed in [11, 19, 20].

Unfortunately, these simplified Newton's methods have poor numerical stability.

In this paper, we propose two new algorithms to compute the nonsingular square root of a matrix, which have good numerical stability. We first apply Samanskill technique, especially, proposed in [21] to compute the matrix square root. Convergence theorems and stability analysis for these new algorithms are given in Sections 3 and 4. In Section 5, we use some numerical examples to show that these new algorithms are more effective than the known ones in some aspects. And the final conclusions are given in Section 6.

## 2. Two New Algorithms

In order to compute the square root of matrix  $A$ , a natural approach is to apply Newton's method to (1), and this can be stated as follows.

*Algorithm 1* (see [11, 19] (Newton's method for (1))). We consider the following.

*Step 0.* Given  $X_0$  and  $\varepsilon$ , set  $k = 0$ .

*Step 1.* Let  $\text{Res}(X_k) = \|X_k^2 - A\|/\|A\|$ . If  $\text{Res}(X_k) < \varepsilon$ , stop.

*Step 2.* Solve for  $H_k$  in Sylvester equation:

$$X_k H_k + H_k X_k = -F(X_k). \quad (2)$$

*Step 3.* Update  $X_{k+1} = X_k + H_k$ ,  $k = k + 1$ , and go to Step 1.

Applying the standard local convergence theorem to Algorithm 1 [19, P. 148], we deduce that the sequence  $\{X_k\}$  generated by Algorithm 1 converges quadratically to a square root  $X_*$  of  $A$  if the starting matrix  $X_0$  is sufficiently close to  $X_*$ .

In this paper, we propose two new algorithms to compute the nonsingular square root of the matrix  $A$ . Our idea can be stated as follows. If (1) has a nonsingular solution  $X$ , then we can transform (1) into an equivalent nonlinear matrix equation:

$$G(X) = X - AX^{-1} = 0. \quad (3)$$

Then we apply Newton's method to (3) for computing the nonsingular square root of  $A$ .

By the definition of F-differentiable and some simple calculations, we obtain that if the matrix  $X$  is nonsingular, then the mapping  $G$  is F-differentiable at  $X$  and

$$G'_X(H) = H + AX^{-1}HX^{-1}. \quad (4)$$

Thus Newton's method for (3) can be written as

$$\text{Given } X_0, \quad X_{k+1} = X_k - \left(G'_{X_k}\right)^{-1} (G(X_k)), \quad (5)$$

$$k = 0, 1, 2, \dots$$

Combining (4), the iteration (5) is equivalent to the following.

*Algorithm 2* (Newton's method for (3)). We consider the following.

*Step 0.* Given  $X_0$  and  $\varepsilon$ , set  $k = 0$ .

*Step 1.* Let  $\text{Res}(X_k) = \|X_k^2 - A\|/\|A\|$ . If  $\text{Res}(X_k) < \varepsilon$ , stop.

*Step 2.* Solve for  $H_k$  in generalized Sylvester equation:

$$AX_k^{-1}H_kX_k^{-1} + H_k = -G(X_k). \quad (6)$$

*Step 3.* Update  $X_{k+1} = X_k + H_k$ ,  $k = k + 1$ , and go to Step 1, where  $\text{Res}(X_k) = \|X_k^2 - A\|/\|A\|$ .

By using Samanskii technique [21] to Newton's method (5), we get the following algorithm.

*Algorithm 3* (Newton's method for (3) with Samanskii technique). We consider the following.

*Step 0.* Given  $X_0$ ,  $m$ , and  $\varepsilon$ , set  $k = 0$ .

*Step 1.* Let  $\text{Res}(X_k) = \|X_k^2 - A\|/\|A\|$ . If  $\text{Res}(X_k) < \varepsilon$ , stop.

*Step 2.* Let  $X_{k,0} = X_k$ ,  $i = 1$ .

*Step 3.* If  $i > m$ , go to Step 6.

*Step 4.* Solve for  $H_{k,i-1}$  in generalized Sylvester equation:

$$AX_k^{-1}H_{k,i-1}X_k^{-1} + H_{k,i-1} = -G(X_{k,i-1}). \quad (7)$$

*Step 5.* Update  $X_{k,i} = X_{k,i-1} + H_{k,i-1}$ ,  $i = i + 1$ , and go to Step 3.

*Step 6.* Update  $X_{k+1} = X_{k,m}$ ,  $k = k + 1$ , and go to Step 1.

*Remark 4.* In this paper, we only consider the case that  $m = 2$ . If  $m = 1$ , then Algorithm 3 is Algorithm 2.

*Remark 5.* Iteration (5) is more suitable for theoretical analysis such as the convergence theorems and stability analysis in Sections 3 and 4, while Algorithms 2 and 3 are more convenient for numerical computation in Section 5. In actual computations, the Sylvester equation  $CXD + EXF = G$  may be solved by the algorithms developed in [22].

Although Algorithms 2 and 3 are also Newton's methods, Algorithms 2 and 3 are more effective than Algorithm 1. Algorithm 3, especially, with  $m = 2$  has cubic convergence rate.

### 3. Convergence Theorems

In this section, we establish local convergence theorems for Algorithms 2 and 3. We begin with some lemmas.

**Lemma 6** (see [23, P. 21]). *Let  $T$  be an (nonlinear) operator from a Banach space  $E$  into itself and let  $x^* \in E$  be a solution of  $x = Tx$ . If  $T$  is Frechet differentiable at  $x^*$  with  $\rho(T'_{x^*}) < 1$ , then the iteration  $x_{n+1} = Tx_n$ ,  $n = 0, 1, 2, \dots$ , converges to  $x^*$ , provided that  $x_0$  is sufficiently close to  $x^*$ .*

**Lemma 7** (see [17, P. 45]). *Let  $A, B \in C^{n \times n}$  and assume that  $A$  is invertible with  $\|A^{-1}\| \leq \alpha$ . If  $\|A - B\| \leq \beta$  and  $\alpha\beta < 1$ , then  $B$  is also invertible, and*

$$\|B^{-1}\| \leq \frac{\alpha}{1 - \alpha\beta}. \quad (8)$$

**Lemma 8.** *If the matrix  $\widehat{X} \in C^{n \times n}$  is nonsingular, then there exist  $\gamma > 0$  and  $L > 0$  such that, for all  $X, Y \in B(\widehat{X}, \gamma)$ , it holds that*

$$\|G'_X - G'_Y\| \leq L \|X - Y\|, \quad (9)$$

where  $B(\widehat{X}, r) = \{X \mid \|X - \widehat{X}\| < r\}$  and  $G'_X, G'_Y$  are the F-derivative of the mapping  $G$  defined by (4) at  $X, Y$ .

*Proof.* Let  $\alpha = \|\widehat{X}^{-1}\|$ , and we select  $0 < \gamma < \alpha^{-1}$ .

From Lemma 7 it follows that  $X$  is nonsingular and  $\|X^{-1}\| \leq \alpha/(1 - \alpha\gamma)$  for each  $X \in B(\widehat{X}, \gamma)$ . Then  $G'_X$  is well defined, and so does  $G'_Y$ , where  $Y \in B(\widehat{X}, \gamma)$ . According to (4), we have

$$\begin{aligned} & \|G'_X(H) - G'_Y(H)\| \\ &= \|(H + AX^{-1}HX^{-1}) - (H + AY^{-1}HY^{-1})\| \\ &= \|AX^{-1}HX^{-1} - AY^{-1}HY^{-1}\| \\ &= \|A[(X^{-1}HX^{-1} - X^{-1}HY^{-1}) \\ &\quad + (X^{-1}HY^{-1} - Y^{-1}HY^{-1})]\| \\ &= \|A[X^{-1}H(X^{-1} - Y^{-1}) + (X^{-1} - Y^{-1})HY^{-1}]\| \end{aligned}$$

$$\begin{aligned}
 &= \|A [X^{-1}HY^{-1} (Y - X) X^{-1} \\
 &\quad + Y^{-1} (Y - X) X^{-1}HY^{-1}]\| \\
 &\leq \|A\| \left( \|X^{-1}\|^2 \|Y^{-1}\| \|Y - X\| \|H\| \right. \\
 &\quad \left. + \|Y^{-1}\|^2 \|X^{-1}\| \|Y - X\| \|H\| \right) \\
 &= \|A\| \|X^{-1}\| \|Y^{-1}\| \left( \|X^{-1}\| + \|Y^{-1}\| \right) \|X - Y\| \|H\| \\
 &\leq \frac{\alpha}{1 - \alpha\gamma} \frac{\alpha}{1 - \alpha\gamma} \left( \frac{\alpha}{1 - \alpha\gamma} + \frac{\alpha}{1 - \alpha\gamma} \right) \|A\| \|X - Y\| \|H\| \\
 &= 2 \left( \frac{\alpha}{1 - \alpha\gamma} \right)^3 \|A\| \|X - Y\| \|H\| \\
 &= L \|X - Y\| \|H\|, \tag{10}
 \end{aligned}$$

where  $L = 2(\alpha/(1 - \alpha\gamma))^3 \|A\|$ .

Hence, we have

$$\|G'_X - G'_Y\| \leq L \|X - Y\|. \tag{11}$$

**Theorem 9.** *If (3) has a nonsingular solution  $X_*$  and the mapping  $G'_{X_*} : C^{n \times n} \rightarrow C^{n \times n}$  is invertible, then there exists a close ball  $S = B(X_*, \delta)$ , such that, for all  $X_0 \in S$ , the sequence  $\{X_k\}$  generated by Algorithm 2 converges at least quadratically to the solution  $X_*$ .*

*Proof.* Let  $\varphi(X) = X - (G'_X)^{-1}(G(X))$ . By Taylor formula in Banach space [24, P. 67], we have

$$\begin{aligned}
 &\lim_{\|H\| \rightarrow 0} \frac{\|\varphi(X_* + H) - \varphi(X_*)\|}{\|H\|} \\
 &= \lim_{\|H\| \rightarrow 0} \left( \left\| \left[ X_* + H - (G'_{X_*+H})^{-1} (G(X_* + H)) \right] \right. \right. \\
 &\quad \left. \left. - \left[ X_* - (G'_{X_*})^{-1} (G(X_*)) \right] \right\| \times \|H\|^{-1} \right) \\
 &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1} (G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\
 &\quad \left. \left. \times (G(X_* + H)) \right\| \times \|H\|^{-1} \right) \\
 &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1} (G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\
 &\quad \left. \left. \times \left[ G(X_*) + G'_{X_*} (H) + \frac{1}{2} G''_{X_*} (H^2) + \dots \right] \right\| \right. \\
 &\quad \left. \times \|H\|^{-1} \right)
 \end{aligned}$$

$$\begin{aligned}
 &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1} (G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\
 &\quad \left. \left. \times (G(X_*)) + (G'_{X_*+H})^{-1} \right. \right. \\
 &\quad \left. \left. \times (G'_{X_*} (H)) + \frac{1}{2} (G'_{X_*+H})^{-1} (G''_{X_*} (H^2)) \right. \right. \\
 &\quad \left. \left. + \dots \right\| \times \|H\|^{-1} \right) \\
 &= 0. \tag{12}
 \end{aligned}$$

Hence, the F-derivative of  $\varphi$  at  $X_*$  is 0. By Lemma 6, we derive that the sequence  $\{X_k\}$  generated by the iteration (5) converges to  $X_*$ . It is also obtained that the sequence  $\{X_k\}$  generated by Algorithm 2 converges to  $X_*$ .

Let  $\|(G'_{X_*})^{-1}\| = \beta$ , according to  $X_k \rightarrow X_*$  ( $k \rightarrow \infty$ ) and Lemma 7; for large enough  $k$ , we have

$$\|(G'_{X_k})^{-1}\| \leq \frac{\beta}{1 - \beta(1/2\beta)} = 2\beta. \tag{13}$$

By Lemma 8, we have

$$\|G'_{X_k} (X_k - X_*) - G'_{X_*} (X_k - X_*)\| \leq L \|X_k - X_*\|^2. \tag{14}$$

By making use of Taylor formula once again, for all  $t \in [0, 1]$ , we have

$$\begin{aligned}
 &\|G(X_k) - G(X_*) - G'_{X_*} (X_k - X_*)\| \\
 &\leq \left\| \int_0^1 \left( G'_{X_*+t(X_k-X_*)} (X_k - X_*) - G'_{X_*} (X_k - X_*) \right) dt \right\| \\
 &\leq \int_0^1 \|G'_{X_*+t(X_k-X_*)} - G'_{X_*}\| dt \|X_k - X_*\| \\
 &\leq L \|X_k + t(X_* - X_k) - X_*\| \|X_k - X_*\| \\
 &= L(1 - t) \|X_k - X_*\|^2 \\
 &\leq L \|X_k - X_*\|^2. \tag{15}
 \end{aligned}$$

Hence,

$$\begin{aligned}
 &\|X_{k+1} - X_*\| \\
 &= \|X_k - (G'_{X_k})^{-1} (G(X_k)) - X_*\| \\
 &= \|(G'_{X_k})^{-1} [G'_{X_k} (X_k - X_*) - GX_k]\| \\
 &= \|(G'_{X_k})^{-1} [(G'_{X_k} (X_k - X_*) - G'_{X_*} (X_k - X_*)) \\
 &\quad - (G(X_k) - G(X_*) - G'_{X_*} (X_k - X_*))]\| \\
 &\leq \|(G'_{X_k})^{-1}\| \left( \|G'_{X_k} (X_k - X_*) - G'_{X_*} (X_k - X_*)\| \right. \\
 &\quad \left. + \|G(X_k) - G(X_*) - G'_{X_*} (X_k - X_*)\| \right). \tag{16}
 \end{aligned}$$

Combining (13)–(16), we have

$$\begin{aligned} \|X_{k+1} - X_*\| &\leq 2\beta L \|X_k - X_*\|^2 + 2\beta L \|X_k - X_*\|^2 \\ &= 4\beta L \|X_k - X_*\|^2, \end{aligned} \quad (17)$$

which implies that the sequence  $\{X_k\}$  generated by Algorithm 2 converges at least quadratically to the solution  $X_*$ .  $\square$

**Theorem 10.** *If (1) has a nonsingular solution  $X_*$  and the mapping  $G'_{X_*} : C^{n \times n} \rightarrow C^{n \times n}$  is invertible, then there exists a close ball  $S = B(X_*, \delta)$ , such that, for all  $X_0 \in S$ , the sequence  $\{X_k\}$  generated by Algorithm 3 converges at least cubically to the solution  $X_*$ .*

*Proof.* Let  $\varphi(X) = X - (G'_X)^{-1}(G(X))$ . By Taylor formula in Banach space [24, P. 67], we have

$$\begin{aligned} &\lim_{\|H\| \rightarrow 0} \frac{\|\varphi(X_* + H) - \varphi(X_*)\|}{\|H\|} \\ &= \lim_{\|H\| \rightarrow 0} \left( \left\| \left[ X_* + H - (G'_{X_*+H})^{-1}(G(X_* + H)) \right] \right. \right. \\ &\quad \left. \left. - \left[ X_* - (G'_{X_*})^{-1}(G(X_*)) \right] \right\| \times \|H\|^{-1} \right) \\ &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1}(G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\ &\quad \left. \left. \times (G(X_* + H)) \right\| \times \|H\|^{-1} \right) \\ &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1}(G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\ &\quad \left. \left. \times \left[ G(X_*) + G'_{X_*}(H) + \frac{1}{2}G''_{X_*}(H^2) + \dots \right] \right\| \right. \\ &\quad \left. \times \|H\|^{-1} \right) \\ &= \lim_{\|H\| \rightarrow 0} \left( \left\| H + (G'_{X_*})^{-1}(G(X_*)) - (G'_{X_*+H})^{-1} \right. \right. \\ &\quad \left. \left. \times (G(X_*)) + (G'_{X_*+H})^{-1}(G'_{X_*}(H)) \right. \right. \\ &\quad \left. \left. + \frac{1}{2}(G'_{X_*+H})^{-1}(G''_{X_*}(H^2)) + \dots \right\| \times \|H\|^{-1} \right) \\ &= 0. \end{aligned} \quad (18)$$

Hence, the F-derivative of  $\varphi$  at  $X_*$  is 0. By Lemma 6, we derive that the sequence  $\{X_k\}$  generated by iteration (5) converges to  $X_*$ . It is also obtained that the sequence  $\{X_k\}$  generated by Algorithm 3 converges to  $X_*$ .

Let  $\|(G'_{X_*})^{-1}\| = \beta$ , according to  $X_k \rightarrow X_*$  ( $k \rightarrow \infty$ ) and Lemma 7; for large enough  $k$ , we have

$$\|(G'_{X_k})^{-1}\| \leq \frac{\beta}{1 - \beta(1/2\beta)} = 2\beta. \quad (19)$$

By Lemma 8, we have

$$\begin{aligned} &\|G'_{X_k}(X_{k,1} - X_*) - G'_{X_*}(X_{k,1} - X_*)\| \\ &\leq L \|X_k - X_*\| \|X_{k,1} - X_*\|. \end{aligned} \quad (20)$$

By making use of Taylor formula once again, for all  $t \in [0, 1]$ , we have

$$\begin{aligned} &\|G(X_{k,1}) - G(X_*) - G'_{X_*}(X_{k,1} - X_*)\| \\ &\leq \left\| \int (G'_{X_{k,1}+t(X_*-X_{k,1})}(X_{k,1} - X_*) - G'_{X_*}(X_{k,1} - X_*)) dt \right\| \\ &\leq \int \|G'_{X_{k,1}+t(X_*-X_{k,1})} - G'_{X_*}\| dt \|X_{k,1} - X_*\| \\ &\leq L \|X_{k,1} + t(X_* - X_{k,1}) - X_*\| \|X_{k,1} - X_*\| \\ &= L(1-t) \|X_{k,1} - X_*\|^2 \\ &\leq L \|X_{k,1} - X_*\|^2. \end{aligned} \quad (21)$$

Hence,

$$\begin{aligned} &\|X_{k+1} - X_*\| \\ &= \|X_{k,1} - (G'_{X_k})^{-1}(G(X_{k,1})) - X_*\| \\ &= \|(G'_{X_k})^{-1} [G'_{X_k}(X_{k,1} - X_*) - G(X_{k,1})]\| \\ &= \|(G'_{X_k})^{-1} \\ &\quad \times [(G'_{X_k}(X_{k,1} - X_*) - G'_{X_*}(X_{k,1} - X_*)) \\ &\quad - (G(X_{k,1}) - G(X_*)) \\ &\quad - G'_{X_*}(X_{k,1} - X_*)]\| \\ &\leq \|(G'_{X_k})^{-1}\| \\ &\quad \times [\|G'_{X_k}(X_{k,1} - X_*) - G'_{X_*}(X_{k,1} - X_*)\| \\ &\quad + \|G(X_{k,1}) - G(X_*) - G'_{X_*}(X_{k,1} - X_*)\|]. \end{aligned} \quad (22)$$

Combining (19)–(22) and Theorem 9, we have

$$\begin{aligned} &\|X_{k+1} - X_*\| \\ &\leq 2\beta [L \|X_k - X_*\| \|X_{k,1} - X_*\| + L \|X_{k,1} - X_*\|^2] \\ &\leq 2\beta L [4\beta L^2 \|X_k - X_*\|^3 + 16\beta^2 L^3 \|X_k - X_*\|^4] \\ &= (8\beta^2 L^3 + 32\beta^3 L^4 \|X_k - X_*\|) \|X_k - X_*\|^3 \\ &\leq (8\beta^2 L^3 + 32\beta^3 L^4 \delta) \|X_k - X_*\|^3 \\ &= M \|X_k - X_*\|^3, \end{aligned} \quad (23)$$

where  $M = 8\beta^2 L^3 + 32\beta^3 L^4 \delta$ . Therefore, the sequence  $\{X_k\}$  generated by Algorithm 3 converges at least cubically to the solution  $X_*$ .  $\square$

#### 4. Stability Analysis

In accordance with [2] we define an iteration  $X_{k+1} = f(X_k)$  to be stable in a neighborhood of a solution  $X = f(X)$ , if the error matrix  $E_k = X_k - X_*$  satisfies

$$E_{k+1} = L(E_k) + O(\|E_k\|^2), \quad (24)$$

where  $L$  is a linear operator that has bounded power; that is, there exists a constant  $c > 0$  such that, for all  $n > 0$  and arbitrary  $E$  of unit norm,  $\|L^n(E)\| < c$ . This means that a small perturbation introduced in a certain step will not be amplified in the subsequent iterations.

Note that this definition of stability is an asymptotic property and is different from the usual concept of numerical stability, which concerns the global error propagation, aiming to bound the minimum relative error over the computed iterates.

Now we consider the iteration (5) and define the error matrix  $E_k = X_k - X_*$ ; that is,

$$X_k = E_k + X_*. \quad (25)$$

For the sake of simplicity, we perform a first order error analysis; that is, we omit all the terms that are quadratic in the errors. Equality up to second order terms is denoted with the symbol  $\doteq$ .

Substituting (25) into (5) we get

$$E_{k+1} + X_* = E_k + X_* - (G'_{E_k+X_*})^{-1} (G(E_k + X_*)); \quad (26)$$

combining (4) we have

$$(E_{k+1} + X_*)(E_k + X_*) + A(E_k + X_*)^{-1} (E_{k+1} + X_*) = 2A, \quad (27)$$

which implies that

$$\begin{aligned} & E_{k+1}E_k + E_{k+1}X_* + X_*E_k + X_*^2 \\ & + A(X_*^{-1} - X_*^{-1}E_kX_*^{-1} + O(E_k^2))(E_{k+1} + X_*) = 2A. \end{aligned} \quad (28)$$

Omitting all terms that are quadratic in the errors, we have

$$E_{k+1}X_* + X_*E_k + X_*^2 + AX_*^{-1}E_{k+1} + A - AX_*^{-1}E_k \doteq 2A. \quad (29)$$

By using  $AX_*^{-1} = X_*$ , we have

$$E_{k+1}X_* + X_*E_k + A + X_*E_{k+1} + A - X_*E_k \doteq 2A; \quad (30)$$

that is,

$$E_{k+1}X_* + X_*E_{k+1} = 0, \quad (31)$$

which means that iteration (5) is self-adaptive; that is to say, the error  $E_k$  in the  $k$ th iteration does not propagate to the  $(k + 1)$ st iteration. When  $X_*$  and  $-X_*$  have no eigenvalue in common, especially, the matrix equation  $EX_* + X_*E = 0$  has a unique solution  $E = 0$  [17, P. 194]. Therefore, under the condition that  $X_*$  and  $-X_*$  have no eigenvalue in common, the iteration (5) has optimal stability; that is, the operator  $L$  defined in (24) coincides with the null operator.

#### 5. Numerical Examples

In this section, we compare our algorithms with the following.

*Algorithm 11* (the Denman-Beavers iteration [9]). Consider

$$\begin{aligned} Y_0 &= A, & Z_0 &= I \\ Y_{k+1} &= \frac{Y_k + Z_k^{-1}}{2} \\ Z_{k+1} &= \frac{Z_k + Y_k^{-1}}{2}, \quad k = 0, 1, 2, \dots \end{aligned} \quad (32)$$

*Algorithm 12* (the scaled Denman-Beavers iteration [13]). Consider

$$\begin{aligned} Y_0 &= A, & Z_0 &= I \\ r_k &= |\det(Y_k) \det(Z_k)|^{-1/2n} \\ Y_{k+1} &= \frac{r_k Y_k + r_k^{-1} Z_k^{-1}}{2} \\ Z_{k+1} &= \frac{r_k Z_k + r_k^{-1} Y_k^{-1}}{2}, \quad k = 0, 1, 2, \dots \end{aligned} \quad (33)$$

*Algorithm 13* (the Pade iteration [13]). Consider

$$\begin{aligned} Y_0 &= A, & Z_0 &= I \\ Y_{k+1} &= \frac{1}{p} Y_k \sum_{i=1}^p \frac{1}{\xi_i} (Z_k Y_k + \alpha_i^2 I)^{-1} \\ Z_{k+1} &= \frac{1}{p} Z_k \sum_{i=1}^p \frac{1}{\xi_i} (Y_k Z_k + \alpha_i^2 I)^{-1}, \quad k = 0, 1, 2, \dots, \end{aligned} \quad (34)$$

where  $p \geq 1$  is a chosen integer:

$$\begin{aligned} \xi_i &= \frac{1}{2} \left( 1 + \cos \frac{(2i-1)\pi}{2p} \right), & \alpha_i^2 &= \frac{1}{\xi_i} - 1, \\ & & k &= 0, 1, 2, \dots, p. \end{aligned} \quad (35)$$

TABLE 1

	IT	CPU	ERR
Algorithm 1	7	0.0086	$2.17 \times 10^{-16}$
Algorithm 2	7	0.0080	$2.04 \times 10^{-16}$
Algorithm 3	5	0.0103	$1.96 \times 10^{-16}$
Algorithm 11	9	0.0172	$1.99 \times 10^{-16}$
Algorithm 12	6	0.0136	$2.03 \times 10^{-16}$
Algorithm 13 with $p = 1$	11	0.0094	$2.71 \times 10^{-16}$
Algorithm 13 with $p = 2$	9	0.0101	$2.68 \times 10^{-16}$
Algorithm 14 with $p = 1$	9	0.0127	$1.97 \times 10^{-16}$
Algorithm 14 with $p = 2$	6	0.0108	$3.61 \times 10^{-16}$

TABLE 2

	IT	CPU	ERR
Algorithm 1	6	7.6310	$5.72 \times 10^{-16}$
Algorithm 2	6	8.7200	$3.61 \times 10^{-16}$
Algorithm 3	4	9.0258	$2.60 \times 10^{-16}$
Algorithm 11	8	13.2301	$3.87 \times 10^{-16}$
Algorithm 12	7	11.6758	$2.98 \times 10^{-16}$
Algorithm 13 with $p = 1$	10	8.8936	$9.36 \times 10^{-16}$
Algorithm 13 with $p = 2$	6	9.4387	$5.78 \times 10^{-16}$
Algorithm 14 with $p = 1$	9	10.3571	$2.89 \times 10^{-16}$
Algorithm 14 with $p = 2$	5	8.1043	$3.87 \times 10^{-16}$

Algorithm 14 (the scaled Pade iteration [13]). Consider

$$\begin{aligned}
 Y_0 &= A, & Z_0 &= I \\
 r_k &= |\det(Y_k) \det(Z_k)|^{-1/2n} \\
 Y_{k+1} &= \frac{1}{p} r_k Y_k \sum_{i=1}^p \frac{1}{\xi_i} (r_k^2 Z_k Y_k + \alpha_i^2 I)^{-1} \\
 Z_{k+1} &= \frac{1}{p} Z_k \sum_{i=1}^p \frac{1}{\xi_i} (r_k^2 Y_k Z_k + \alpha_i^2 I)^{-1}, \quad k = 0, 1, 2, \dots
 \end{aligned} \tag{36}$$

All tests are performed by using MATLAB 7.1 on a personal computer (Pentium IV/2.4 G), with machine precision  $2.2 \times 10^{-16}$ . The stopping criterion for these algorithms is the relative residual error:

$$\text{Res} = \frac{\|X_k^2 - A\|}{\|A\|} < 10^{-15}, \tag{37}$$

where  $X_k$  is the current, say the  $k$ th, iteration value.

Example 1. Consider the matrix

$$A = (a_{ij})_{10 \times 10} = \begin{cases} \frac{j}{20}, & i = j; \\ \frac{i+j}{1000}, & i \neq j. \end{cases} \tag{38}$$

We use Algorithms 1, 2, and 3 with  $X_0 = 0.3I$  and Algorithms 11–14 to compute the nonsingular square root of  $A$ . We list the iteration steps (denoted by “IT”), CPU time (denoted by “CPU”), and the relative residual error (denoted by “ERR”) in Table 1.

Example 2. Consider the matrix

$$A = (a_{ij})_{200 \times 200} = \begin{cases} 1, & i = j; \\ \frac{1}{i+j-1}, & i \neq j. \end{cases} \tag{39}$$

We use Algorithms 1, 2, and 3 with the starting matrix  $X_0 = 0.9I$  and Algorithms 11–14 to compute the nonsingular square root of  $A$ . We list the numerical results in Table 2.

From Tables 1 and 2, we can see that Algorithms 2 and 3 outperform Algorithms 1, 11, 12, and 13 in both iteration steps and approximation accuracy, and Algorithm 3 outperforms Algorithms 1, 2, and 11–14 in both iteration steps and approximation accuracy. Therefore, our algorithms are more effective than the known ones in some aspects.

### 6. Conclusion

In this paper, we propose two new algorithms for computing the nonsingular square root of a matrix  $A$  by applying Newton’s method to nonlinear matrix equation  $G(X) = X - AX^{-1} = 0$ . Convergence theorems and stability analysis for these new algorithms are given. Numerical examples show that our methods are more effective than the known one in some aspects.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

The authors wish to thank the editor and anonymous referees for providing very useful suggestions as well as Professor Xuefeng Duan for his insightful and beneficial discussion and suggestions. This work was supported by National Natural Science Fund of China (nos. 11101100, 11301107, and 11261014) and Guangxi Provincial Natural Science Foundation (nos. 2012GXNSFBA053006, 2013GXNSFBA019009).

### References

- [1] B. A. Schmitt, “On algebraic approximation for the matrix exponential in singularly perturbed bounded value problems,” *SIAM Journal on Numerical Analysis*, vol. 57, pp. 51–66, 2010.
- [2] S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub, “Approximating the logarithm of a matrix to specified accuracy,” *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 4, pp. 1112–1125, 2001.
- [3] L. Dieci, B. Morini, and A. Papini, “Computational techniques for real logarithms of matrices,” *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 3, pp. 570–593, 1996.

- [4] G. W. Cross and P. Lancaster, "Square roots of complex matrices," *Linear and Multilinear Algebra*, vol. 1, pp. 289–293, 1974.
- [5] W. D. Hoskins and D. J. Walton, "A faster method of computing square roots of a matrix," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 494–495, 1978.
- [6] C. R. Johnson and K. Okubo, "Uniqueness of matrix square roots under a numerical range condition," *Linear Algebra and Its Applications*, vol. 341, no. 1–3, pp. 195–199, 2002.
- [7] Å. Björck and S. Hammarling, "A Schur method for the square root of a matrix," *Linear Algebra and Its Applications*, vol. 52–53, pp. 127–140, 1983.
- [8] S. G. Chen and P. Y. Hsieh, "Fast computation of the  $n$ th root," *Computers & Mathematics with Applications*, vol. 17, no. 10, pp. 1423–1427, 1989.
- [9] E. D. Denman and A. N. Beavers Jr., "The matrix sign function and computations in systems," *Applied Mathematics and Computation*, vol. 2, no. 1, pp. 63–94, 1976.
- [10] L. P. Franca, "An algorithm to compute the square root of a  $3 \times 3$  positive definite matrix," *Computers & Mathematics with Applications*, vol. 18, no. 5, pp. 459–466, 1989.
- [11] N. J. Higham, "Newton's method for the matrix square root," *Mathematics of Computation*, vol. 46, no. 174, pp. 537–549, 1986.
- [12] M. A. Hasan, "A power method for computing square roots of complex matrices," *Journal of Mathematical Analysis and Applications*, vol. 213, no. 2, pp. 393–405, 1997.
- [13] N. J. Higham, "Stable iterations for the matrix square root," *Numerical Algorithms*, vol. 15, no. 2, pp. 227–242, 1997.
- [14] Z. Liu, Y. Zhang, and R. Ralha, "Computing the square roots of matrices with central symmetry," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 715–726, 2007.
- [15] Y. N. Zhang, Y. W. Yang, B. H. Cai, and D. S. Guo, "Zhang neural network and its application to Newton iteration for matrix square root estimation," *Neural Computing and Applications*, vol. 21, no. 3, pp. 453–460, 2012.
- [16] Z. Liu, H. Chen, and H. Cao, "The computation of the principal square roots of centrosymmetric H-matrices," *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 319–329, 2006.
- [17] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, Philadelphia, Pa, USA, 2008.
- [18] P. Laasonen, "On the iterative solution of the matrix equation  $AX^2 - I = 0$ ," *Mathematical Tables and Other Aids to Computation*, vol. 12, pp. 109–116, 1958.
- [19] J. M. Ortega, *Numerical Analysis*, Academic Press, New York, NY, USA, 2nd edition, 1972.
- [20] B. Meini, "The matrix square root from a new functional perspective: theoretical results and computational issues," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 362–376, 2004.
- [21] V. Samanskii, "On a modification of the Newton's method," *Ukrainian Mathematical Journal*, vol. 19, pp. 133–138, 1967.
- [22] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler, "Solution of the Sylvester matrix equation  $AXB^T + CXD^T = E$ ," *Association for Computing Machinery: Transactions on Mathematical Software*, vol. 18, no. 2, pp. 223–231, 1992.
- [23] M. A. Krasnoselskii, G. M. Vainikko, P. P. Zabreiko, Y. B. Rutitskii, and V. Y. Stetsenko, *Approximate Solution of Operator Equations*, Wolters-Noordhoff Publishing, Groningen, The Netherlands, 1972.
- [24] D. J. Guo, *Nonlinear Functional Analysis*, Shandong Science Press, Shandong, China, 2009.