Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 601243, 6 pages
http://dx.doi.org/10.1155/2013/601243

Research Article

A Note on the Square Roots of a Class of Circulant Matrices

Ying Zhang, Huisheng Zhang, and Guoyan Chen

Department of Mathematics, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Ying Zhang; yingzhang@dlmu.edu.cn

Received 28 July 2013; Accepted 3 November 2013

Academic Editor: Frank Werner

We prove that any k-circulant matrix and any even order skew k-circulant matrix are diagonalizable for any $k \in \mathbb{C}$. Then, we propose two algorithms for computing the square roots of the k-circulant matrix and the skew k-circulant matrix, respectively. In particular, we show that the square roots of the k-circulant matrix are still k-circulant matrices. Both the theoretical analysis and the numerical experiments show that our algorithms are faster than the standard Schur method.

1. Introduction

Given a matrix A, a matrix X is called a square root of A if $X^2 = A$. Matrix square roots appear in a variety of branches of mathematics, such as Markov models of finance, the solution of differential equations, the computation of the polar decomposition, and the matrix sign function [1]. A number of methods have been proposed for computing the square roots of a matrix [2–9]. Among them, the Schur method [7] is the most popular and becomes the standard method for computing the matrix square roots. However, Schur method is not so efficient when the matrix order is relatively high. Thus, it is very desirable to design fast computing methods which can make full use of the particular properties of the matrices when the matrices possess special structures.

Circulant matrices and their generalizations have a wide range of applications in signal processing, coding theory, digital image disposal, self-regress design, Toeplitz systems, and so on [10–14]. A relatively comprehensive survey about circulant matrices can be found in [15]. Recently, Lu and Gu [16] presented two efficient algorithms to compute the square roots of circulant matrices and quasi-skew circulant matrices, respectively. As they are based on LL iteration [17] and the modified Schulz iterative method, the two algorithms are faster than the standard Schur method. Subsequently, Mei generalized those methods and presented algorithms to compute the square roots of k-circulant matrices and skew k-circulant matrices [18]. These algorithms are also faster than the standard Schur algorithm, but the work is restricted to the case in which the matrix is of even order and $k \in \mathbb{R}$ and can not be directly extended to compute the pth root.

In this paper, we first show that k-circulant matrices of any order for any complex number k are diagonalizable and develop an algorithm to compute their principal square roots. Then, we show that skew k-circulant matrices of even order for any complex number k are diagonalizable, and develop an algorithm to compute their principle square roots. Both of our algorithms are theoretically and experimentally proved to be faster than the standard Schur method. Compared with the work in [18], our methods are more general in that they are valid for any k-circulant matrix and any even order skew k-circulant matrix, where k can be any complex number. The remainder of this paper is organized as follows. In Section 2, we compute the square roots of k-circulant matrices. In Section 3, we compute the square roots of skew k-circulant matrices. In Section 4, we present two numerical experiments to exhibit the efficiency of the proposed algorithms in terms of the CPU time.

2. Square Roots of k-Circulant Matrices

An $n \times n$ complex matrix

$$\mathrm{Circ}_k(a) := \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ ka_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ ka_{n-2} & ka_{n-1} & a_0 & \cdots & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ka_1 & ka_2 & ka_3 & \cdots & a_0 \end{bmatrix} \qquad (1)$$

is called a *k-circulant matrix*, where $a = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{C}^n$ and $k \in \mathbb{C}$. In particular, 1-circulant matrices are *circulant matrices*, and -1-circulant matrices are *skew circulant matrices* [19].

Another equivalent definition of a k-circulant matrix is as follows [18]: let $\mathbb{C}^{n\times n}$ be the set of all $n \times n$ complex matrices, and then, $A \in \mathbb{C}^{n\times n}$ is a k-circulant matrix if and only if $A = G^{-1}AG$, where $G = \mathrm{Circ}_k([0, 1, 0, \ldots, 0])$. In this section, we show that k-circulant matrices are diagonalizable.

Lemma 1 (see [20]). *If A and B are two k-circulant matrices of the same order, then $A + B$ is also a k-circulant matrix.*

Lemma 2 (see [20]). *If A is a k-circulant matrix, then for any $l \in \mathbb{C}$, lA is also a k-circulant matrix.*

Lemma 3 (see [20]). *Let $G = Circ_k([0, 1, 0, \ldots, 0])$; then,*

$$G^i = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \\ k & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & k & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2}$$

is also a k-circulant matrix. In particular, $G^n = kE$, where E is the identity matrix.

Lemma 4 (see [20]). *Let $G = Circ_k([0, 1, 0, \ldots, 0])$ and let $k = re^{i\theta}$ ($r \geq 0$, $0 \leq \theta \leq 2\pi$); then, the eigenvalues of G are*

$$\theta_j = r^{1/n} e^{i(\theta + 2\pi(j-1)/n)}, \quad (j = 1, \ldots, n), \tag{3}$$

where i is the imaginary unit.

Theorem 5 (see [20]). *The matrix $C = Circ_k(a)$ is a k-circulant matrix of the form* (1) *if and only if C can be represented by*

$$C = \sum_{j=0}^{n-1} a_j G^j, \tag{4}$$

where $a = (a_0, \ldots, a_{n-1})$, $G = Circ_k([0, 1, 0, \ldots, 0])$, and $G^0 = E$.

Theorem 6 (see [20]). *Let $C = Circ_k(a)$ be a k-circulant matrix; then, the eigenvalues of C are $\lambda_j = \phi(\theta_j)$ $(j = 1, \ldots, n)$, where $a = (a_0, \ldots, a_{n-1})$, $\phi(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$, and θ_j is defined by* (3).

Theorem 7 (see [20]). *Any k-circulant matrix $Circ_k(a)$ can be diagonalized as follows:*

$$V^{-1} \cdot Circ_k(a) \cdot V = \mathrm{diag}\left[\phi(\theta_1), \phi(\theta_2), \ldots, \phi(\theta_n)\right] \tag{5}$$

with the matrix

$$V = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_n \\ \theta_1^2 & \theta_2^2 & \theta_3^2 & \cdots & \theta_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_1^{n-1} & \theta_2^{n-1} & \theta_3^{n-1} & \cdots & \theta_n^{n-1} \end{bmatrix}, \tag{6}$$

where θ_j ($j = 1, \ldots, n$) is defined as (3).

By (5), we can easily obtain the following result.

Corollary 8. *The square roots of k-circulant matrix $Circ_k(a)$ are as follows:*

$$V \,\mathrm{diag}\left[\phi(\theta_1)^{1/2}, \phi(\theta_2)^{1/2}, \ldots, \phi(\theta_n)^{1/2}\right] V^{-1}. \tag{7}$$

Remark 9. We mention that the diagonalization methods [19–22] were used to compute the square roots of k-circulant matrices, where k is restricted to 1 or -1. However, our diagonalization method is valid for any $k \in \mathbb{C}$.

Next, we show that the square roots of a k-circulant matrix are still k-circulant matrices.

Theorem 10. *Let $k = re^{i\theta}$ ($r \geq 0$, $0 \leq \theta \leq 2\pi$), let $\theta_j = r^{1/n} e^{i(\theta + 2\pi(j-1)/n)}$ ($j = 1, \ldots, n$), and let V be defined as* (6). *For any diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{C}^{n\times n}$, $V\Lambda V^{-1}$ is a k-circulant matrix $Circ_k([b_0, \ldots, b_{n-1}])$, where $(b_0, \ldots, b_{n-1}) = (\lambda_1, \ldots, \lambda_n)V^{-1}$.*

Proof. By Theorem 7, whether $V\Lambda V^{-1}$ is a k-circulant matrix depends on whether the following system of linear equations with unknown vector $(b_0, \ldots, b_{n-1})$

$$\begin{gathered} b_0 + b_1\theta_1 + b_2\theta_1^2 + \cdots + b_{n-1}\theta_1^{n-1} = \lambda_1, \\ b_0 + b_1\theta_2 + b_2\theta_2^2 + \cdots + b_{n-1}\theta_2^{n-1} = \lambda_2 \\ \vdots \\ b_0 + b_1\theta_n + b_2\theta_n^2 + \cdots + b_{n-1}\theta_n^{n-1} = \lambda_n \end{gathered} \tag{8}$$

is consistent. Obviously the coefficient matrix of (8) is invertible. Thus, we have

$$(b_0, \ldots, b_{n-1}) = (\lambda_1, \ldots, \lambda_n) V^{-1}. \tag{9}$$

That is to say, $V\Lambda V^{-1}$ is a k-circulant matrix with the form of $\mathrm{Circ}_k([b_0, \ldots, b_{n-1}])$. □

Remark 11. Theorem 10 provides a method to construct k-circulant matrix with given eigenvalues. Obviously, the square roots with the form of (7) are k-circulant matrices $\mathrm{Circ}_k(b)$, where $b = (\phi(\theta_1)^{1/2}, \ldots, \phi(\theta_n)^{1/2})V^{-1}$.

Based on Theorem 7 and Corollary 8, we give the following algorithm for computing the principal square root of a k-circulant matrix.

Algorithm 12. Compute a principal square root of a k-circulant matrix $A \in \mathbb{C}^{n\times n}$.

Step 1. Compute the eigenvalues $\phi(\theta_j) = \sum_{s=0}^{n-1} a_s(\theta_j)^s$ $(j = 1, \ldots, n)$ of A.

Step 2. Compute $\phi(\theta_j)^{1/2}$ $(j = 1, \ldots, n)$ such that $\arg(\phi(\theta_j)^{1/2})$ $(j = 1, \ldots, n) \in (-\pi/2, \pi/2)$.

Step 3. Compute the inverse of V.

Step 4. Compute $\operatorname{diag}[\phi(\theta_1)^{1/2}, \phi(\theta_2)^{1/2}, \ldots, \phi(\theta_n)^{1/2}]V^{-1}$.

Step 5. Compute the square root $V \operatorname{diag}[\phi(\theta_1)^{1/2}, \phi(\theta_2)^{1/2}, \ldots, \phi(\theta_n)^{1/2}]V^{-1}$ of A.

Then, we obtain $B = \sqrt{A} = \mathrm{Circ}_k([b_0, \ldots, b_{n-1}])$.

The cost of Step 1 is about $O(n \log n)$ flops by discrete Fourier transform [18]. The cost of Step 2 is $O(n)$ flops. The cost of Step 3 is about $O(n \log^2 n)$ flops [23]. The cost of Step 4 is about $O(n^2)$ flops. The cost of Step 5 is about $O(n^2 \log^2 n)$ flops [24]. So, it needs about $O(n^2 \log^2 n)$ flops in total. The algorithm has the same complexity as the diagonalization methods in [19–21]. But the methods therein are only concerned with the case that $k = 1$ and $k = -1$. If we use the Schur method, it needs about $O(n^3)$ flops in total [7]. We also mention a related work in [18], which only needs about $O(n \log n)$ flops to compute the primary square root of a k-circulant matrix. However, that work restricts the matrix to be of even order and k to be of real number. Those restrictions are not needed in our algorithm.

3. Square Roots of Skew k-Circulant Matrices

Let n be an even number; then, $A \in \mathbb{C}^{n\times n}$ is a skew k-circulant matrix if $A = -G^{-1}AG$, where $G = \mathrm{Circ}_k([0, 1, 0, \ldots, 0])$ (see [18]). Let $k = re^{i\theta}$ $(r \geq 0, 0 \leq \theta \leq 2\pi)$ and let

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 \end{bmatrix}. \qquad (10)$$

Lemma 13. *Let $G = Circ_k([0, 1, 0, \ldots, 0])$. Then,*

(i) *$JGJ = -G$;*

(ii) *$JG^j = -G^jJ$ for odd number j; $JG^j = G^jJ$ for even number j.*

Lemma 14. *Let*

$$V = [v_1, v_2, \ldots, v_{2m}] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \theta_1 & \theta_2 & \cdots & \theta_{2m} \\ \theta_1^2 & \theta_2^2 & \cdots & \theta_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1^{2m-1} & \theta_2^{2m-1} & \cdots & \theta_{2m}^{2m-1} \end{bmatrix}, \qquad (11)$$

where $\theta_j = r^{1/2m} e^{i(\theta + 2\pi(j-1)/2m)}$ $(j = 1, \ldots, 2m)$; then, $Jv_t = v_{m+t} (t = 1, \ldots, m)$.

Proof. The proof is obvious by paying attention to the fact that $\theta_{m+t} = \theta_t e^{i\pi}$. □

Lemma 15. *A skew k-circulant matrix A of order $2m$ can be written in the form of*

$$JCirc_k([a_0, a_1, \ldots, a_{2m-1}]). \qquad (12)$$

Proof. Write $n = 2m$ and $A = (a_{ij})_{n\times n}$. Since $A = -G^{-1}AG$ we have $GA = -AG$; that is,

$$\begin{bmatrix} a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \\ ka_{11} & ka_{12} & ka_{13} & \cdots & ka_{1n} \end{bmatrix} = \begin{bmatrix} -ka_{1n} & -a_{11} & -a_{12} & \cdots & -a_{1(n-1)} \\ -ka_{2n} & -a_{21} & -a_{22} & \cdots & -a_{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -ka_{(n-1)n} & -a_{(n-1)1} & -a_{(n-1)2} & \cdots & -a_{(n-1)(n-1)} \\ -ka_{nn} & -a_{n1} & -a_{n2} & \cdots & -a_{n(n-1)} \end{bmatrix}. \qquad (13)$$

Then,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ -ka_{1n} & -a_{11} & -a_{12} & \cdots & -a_{1(n-1)} \\ -ka_{2n} & -a_{21} & -a_{22} & \cdots & -a_{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -ka_{(n-1)n} & -a_{(n-1)1} & -a_{(n-1)2} & \cdots & -a_{(n-1)(n-1)} \end{bmatrix}$$

$$\begin{aligned} &(ka_{11}, ka_{12}, ka_{13}, \ldots, ka_{1n}) \\ &\quad = (-ka_{nn}, -a_{n1}, -a_{n2}, \ldots, -a_{n(n-1)}). \end{aligned} \qquad (14)$$

Thus, we have that

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ -ka_{1n} & -a_{11} & -a_{12} & \cdots & -a_{1(n-1)} \\ ka_{1(n-1)} & ka_{1n} & a_{11} & \cdots & a_{1(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -ka_{12} & -ka_{13} & -ka_{14} & \cdots & -a_{11} \end{bmatrix} \\ &= J\mathrm{Circ}_k([a_{11}, a_{12}, \ldots, a_{1n}]). \end{aligned} \qquad (15)$$

□

Theorem 16. *A skew k-circulant matrix $JCirc_k([a_0, \ldots, a_{2m-1}])$ of order $n = 2m$ is diagonalizable.*

Proof. Write $C = \mathrm{Circ}_k([a_0, \ldots, a_{2m-1}])$; then $C = a_0G^0 + a_1G + a_2G^2 + \cdots + a_{2m-1}G^{2m-1}$. By Lemma 13,

$$JC = \left(a_0G^0 - a_1G + a_2G^2 - \cdots - a_{2m-1}G^{2m-1}\right)J. \qquad (16)$$

Thus, we have

$$C = J\left(a_0G^0 - a_1G + a_2G^2 - \cdots - a_{2m-1}G^{2m-1}\right)J. \qquad (17)$$

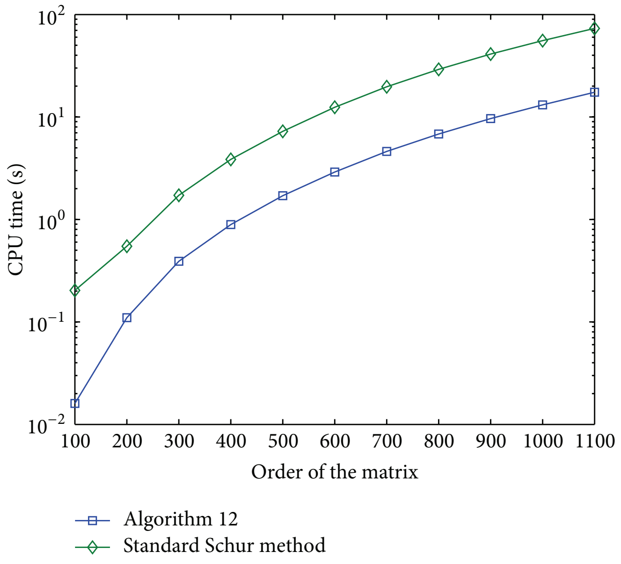

Figure 1: Comparison of the CPU time for Algorithm 12 and the standard Schur method in logarithmic scale.

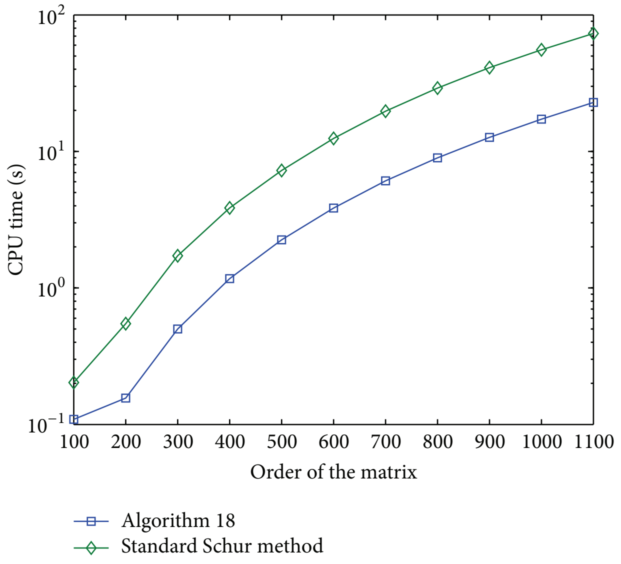

Figure 2: Comparison of the CPU time for Algorithm 18 and the standard Schur method in logarithmic scale.

By Theorem 7, there exists an invertible matrix V defined in Lemma 14 such that

$$\begin{aligned} &V^{-1}J\left(a_0G^0 - a_1G + a_2G^2 - \cdots - a_{2m-1}G^{2m-1}\right)JV \\ &\quad = \operatorname{diag}\left[\phi\left(\theta_1\right), \phi\left(\theta_2\right), \ldots, \phi\left(\theta_{2m}\right)\right], \end{aligned} \tag{18}$$

where $\phi(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{2m-1}x^{2m-1}$. Then,

$$\begin{aligned} &\left(a_0G^0 - a_1G + a_2G^2 - \cdots - a_{2m-1}G^{2m-1}\right)JV \\ &\quad = JV \operatorname{diag}\left[\phi\left(\theta_1\right), \phi\left(\theta_2\right), \ldots, \phi\left(\theta_{2m}\right)\right]. \end{aligned} \tag{19}$$

By (16), we have

$$\begin{aligned} &JC\left[v_1, v_2, \ldots, v_{2m}\right] \\ &\quad = \left[Jv_1, Jv_2, \ldots, Jv_{2m}\right] \operatorname{diag}\left[\phi\left(\theta_1\right), \phi\left(\theta_2\right), \ldots, \phi\left(\theta_{2m}\right)\right]. \end{aligned} \tag{20}$$

In order to obtain the eigenvalues and eigenvectors of JC, for $t = 1, \ldots, m$, we assume that there exist k_t, k_{m+t}, and $l \in \mathbb{C}$, such that

$$JC\left(k_tv_t + k_{m+t}v_{m+t}\right) = l\left(k_tv_t + k_{m+t}v_{m+t}\right). \tag{21}$$

Combining this with (20) gives

$$k_t\phi\left(\theta_t\right)Jv_t + k_{m+t}\phi\left(\theta_{m+t}\right)Jv_{m+t} = lk_tv_t + lk_{m+t}v_{m+t}. \tag{22}$$

Now, we use Lemma 14 to get

$$k_t\phi\left(\theta_t\right)v_{m+t} + k_{m+t}\phi\left(\theta_{m+t}\right)v_t = lk_{m+t}v_{m+t} + lk_tv_t. \tag{23}$$

Since v_t and v_{m+t} are linearly independent, we have

$$\begin{aligned} k_t\phi\left(\theta_t\right) &= lk_{m+t}, \\ k_{m+t}\phi\left(\theta_{m+t}\right) &= lk_t. \end{aligned} \tag{24}$$

So, $l = \pm\sqrt{\phi(\theta_t)\phi(\theta_{m+t})}$, $k_{m+t}/k_t = \pm\sqrt{\phi(\theta_t)/\phi(\theta_{m+t})}$. Namely,

$$\begin{aligned} &JC\left(\sqrt{\phi\left(\theta_{m+t}\right)}v_t + \sqrt{\phi\left(\theta_t\right)}v_{m+t}\right) \\ &\quad = \sqrt{\phi\left(\theta_t\right)\phi\left(\theta_{m+t}\right)}\left(\sqrt{\phi\left(\theta_{m+t}\right)}v_t + \sqrt{\phi\left(\theta_t\right)}v_{m+t}\right), \\ &JC\left(\sqrt{\phi\left(\theta_{m+t}\right)}v_t - \sqrt{\phi\left(\theta_t\right)}v_{m+t}\right) \\ &\quad = -\sqrt{\phi\left(\theta_t\right)\phi\left(\theta_{m+t}\right)}\left(\sqrt{\phi\left(\theta_{m+t}\right)}v_t - \sqrt{\phi\left(\theta_t\right)}v_{m+t}\right). \end{aligned} \tag{25}$$

Let

$$\begin{aligned} W = \Big[&\sqrt{\phi\left(\theta_{m+1}\right)}v_1 + \sqrt{\phi\left(\theta_1\right)}v_{m+1}, \\ &\sqrt{\phi\left(\theta_{m+1}\right)}v_1 - \sqrt{\phi\left(\theta_1\right)}v_{m+1}, \ldots, \\ &\sqrt{\phi\left(\theta_{2m}\right)}v_m + \sqrt{\phi\left(\theta_m\right)}v_{2m}, \\ &\sqrt{\phi\left(\theta_{2m}\right)}v_m - \sqrt{\phi\left(\theta_m\right)}v_{2m}\Big]. \end{aligned} \tag{26}$$

Then

$$JCW = W \begin{bmatrix} \sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & & \\ & -\sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & \\ & & \ddots & & \\ & & & \sqrt{\phi(\theta_m)\phi(\theta_{2m})} & \\ & & & & -\sqrt{\phi(\theta_m)\phi(\theta_{2m})} \end{bmatrix}. \quad (27)$$

So,

$$W^{-1}JCW = \begin{bmatrix} \sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & & \\ & -\sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & \\ & & \ddots & & \\ & & & \sqrt{\phi(\theta_m)\phi(\theta_{2m})} & \\ & & & & -\sqrt{\phi(\theta_m)\phi(\theta_{2m})} \end{bmatrix}. \quad (28)$$

□

Corollary 17. *The square roots of an even order skew k-circulant matrix $JCirc_k(a)$ are as follows:*

$$W \begin{bmatrix} \sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & & \\ & -\sqrt{\phi(\theta_1)\phi(\theta_{m+1})} & & & \\ & & \ddots & & \\ & & & \sqrt{\phi(\theta_m)\phi(\theta_{2m})} & \\ & & & & -\sqrt{\phi(\theta_m)\phi(\theta_{2m})} \end{bmatrix}^{1/2} W^{-1}. \quad (29)$$

Proof. This is a direct result from (28). □

Based on Theorem 16 and Corollary 17, we give the following algorithm for computing the principal square root of a skew k-circulant matrix.

Algorithm 18. Compute a principal square root of a skew k-circulant matrix $A \in \mathbb{C}^{2m\times 2m}$.

Step 1. Compute $\phi(\theta_j) = \sum_{s=0}^{2m-1} a_s(\theta_j)^s$ $(j = 1, \ldots, 2m)$.

Step 2. Compute the eigenvalues of A: $\sqrt{\phi(\theta_1)\phi(\theta_{m+1})}$, $-\sqrt{\phi(\theta_1)\phi(\theta_{m+1})}, \ldots, \sqrt{\phi(\theta_m)\phi(\theta_{2m})}, -\sqrt{\phi(\theta_m)\phi(\theta_{2m})}$.

Step 3. Compute the square roots of $\sqrt{\phi(\theta_1)\phi(\theta_{m+1})}$, $-\sqrt{\phi(\theta_1)\phi(\theta_{m+1})}, \ldots, \sqrt{\phi(\theta_m)\phi(\theta_{2m})}, -\sqrt{\phi(\theta_m)\phi(\theta_{2m})}$ whose arguments should be in $(-\pi/2, \pi/2)$.

Step 4. Compute (29).

Let $n = 2m$. The cost of Step 1 is about $O(n \log n)$ flops by discrete Fourier transform [18]. The cost of Step 2 is $O(n)$. The cost of Step 3 is $O(n)$. Since the matrix multiplication and inversion are equivalent in computational complexity [25], the cost of Step 4 is about $O(n^{2.4})$ [26]. So, it needs about $O(n^{2.4})$ flops in total. If we use the Schur method [7] or the method in [18], it needs about $O(n^3)$ flops in total. Moreover, our method allows k to be a complex number, whereas the method in [18] only permits k to be a real number.

4. Numerical Experiments

We present numerical experiments to compare the algorithms presented in this paper and the standard Schur method with respect to the execution time. The code was implemented in MATLAB 7.0 and run on a Windows-based machine with 2 GB of RAM and Intel Pentium Core Duo CPU running at 2.8 GHz.

We gradually increase the order of the matrix by 100 from 100 to 1100 and record the corresponding CPU time. Figure 1 shows the execution time for Algorithm 12 and the standard

Schur method. Figure 2 compares the execution time for Algorithm 18 and the standard Schur method. From those results, we can confirm that our algorithms are clearly faster than the standard Schur method for computing the square roots of k-circulant matrices and skew k-circulant matrices.

Conflict of Interests

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

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (61101228) and by the Fundamental Research Funds for the Central Universities (3132013337).

References

[1] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, Pa, USA, 2008.

[2] N. J. Higham, "Newton's method for the matrix square root," *Mathematics of Computation*, vol. 46, no. 174, pp. 537–549, 1986.

[3] N. J. Higham, "Stable iterations for the matrix square root," *Numerical Algorithms*, vol. 15, no. 2, pp. 227–242, 1997.

[4] E. D. Denman, "Roots of real matrices," *Linear Algebra and Its Applications*, vol. 36, pp. 133–139, 1981.

[5] L. S. Shieh and N. Chahin, "A computer-aided method for the factorization of matrix polynomials," *International Journal of Systems Science*, vol. 12, no. 3, pp. 305–323, 1981.

[6] 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.

[7] A. 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] W. D. Hoskins and D. J. Walton, "A fast method of computing the square root of a matrix," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 494–495, 1978.

[9] W. D. Hoskins and D. J. Walton, "A faster, more stable method for computing the pth roots of positive definite matrices," *Linear Algebra and Its Applications*, vol. 26, pp. 139–163, 1979.

[10] W. Zhao, "The inverse problem of anti-circulant matrices in signal processing," in *Proceedings of the Pacific-Asia Conference on Knowledge Engineering and Software Engineering*, pp. 47–50, Shenzhen, China, 2009.

[11] G. Zhao, "The improved nonsingularity on the r-circulant matrices in signal processing," in *Proceedings of the International Conference on Computer Techology and Development*, pp. 564–567, Kota Kinabalu, Malaysia, 2009.

[12] V. C. Liu and P. P. Vaidyanathan, "Circulant and skew-circulant matrices as new normal-form realization of IIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 6, pp. 625–635, 1988.

[13] A. Mayer, A. Castiaux, and J.-P. Vigneron, "Electronic Green scattering with n-fold symmetry axis from block circulant matrices," *Computer Physics Communications*, vol. 109, no. 1, pp. 81–89, 1998.

[14] M. K. Ng, "Circulant and skew-circulant splitting methods for Toeplitz systems," *Journal of Computational and Applied Mathematics*, vol. 159, no. 1, pp. 101–108, 2003.

[15] C. Lu, "On the logarithms of circulant matrices," *Journal of Computational Analysis and Applications*, vol. 15, no. 3, pp. 402–412, 2013.

[16] C. Lu and C. Gu, "The computation of the square roots of circulant matrices," *Applied Mathematics and Computation*, vol. 217, no. 16, pp. 6819–6829, 2011.

[17] L. Lin and Z.-Y. Liu, "On the square root of an H-matrix with positive diagonal elements," *Annals of Operations Research*, vol. 103, pp. 339–350, 2001.

[18] Y. Mei, "Computing the square roots of a class of circulant matrices," *Journal of Applied Mathematics*, Article ID 647623, 15 pages, 2012.

[19] R. H. Chan and M. K. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Review*, vol. 38, no. 3, pp. 427–482, 1996.

[20] P. J. Davis, *Circulant Matrices*, John Wiley & Sons, New York, NY, USA, 1979.

[21] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.

[22] C. Lu and C. Gu, "The computation of the inverse of block-wise centrosymmetric matrices," *Publicationes Mathematicae Debrecen*, vol. 82, no. 2, pp. 379–397, 2013.

[23] T. Finck, G. Heinig, and K. Rost, "An inversion formula and fast algorithms for Cauchy-Vandermonde matrices," *Linear Algebra and Its Applications*, vol. 183, pp. 179–191, 1993.

[24] I. Gohberg and V. Olshevsky, "Complexity of multiplication with vectors for structured matrices," *Linear Algebra and Its Applications*, vol. 202, pp. 163–192, 1994.

[25] J. R. Bunch and J. E. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Mathematics of Computation*, vol. 28, pp. 231–236, 1974.

[26] C. H. Papadimitriou, *Computational Complexity*, John Wiley & Sons, Chichester, UK, 2003.