International Journal of
Mathematical Modelling & Computations
Vol. 05, No. 01, Winter 2015, 69- 79

A Stable Coupled Newton's Iteration for the Matrix Inverse p^{th} Root

A. Sadeghi [*]

Young Researchers Club, Shahre-Rey branch, Islamic Azad University, Tehran, Iran.

Abstract. The computation of the inverse roots of matrices arises in evaluating unsymmetric eigenvalue problems, solving nonlinear matrix equations, computing some matrix functions, control theory and several other areas of applications. It is possible to approximate the matrix inverse p^{th} roots by exploiting a specialized version of Newton's method, but previous researchers have mentioned that some iterations have poor convergence and stability properties. In this work, a stable recursive technique to evaluate an inverse p^{th} root of a given matrix is presented. The scheme is analyzed and its properties are investigated. Computational experiments are also performed to illustrate the strengths and weaknesses of the proposed method.

Received: 19 April 2014, Revised: 7 July 2014, Accepted: 12 August 2014.

Keywords: Inverse matrix p^{th} roots, Coupled Newton's iterations, Convergency, Stability.

Index to information contained in this paper

1. Introduction

The inverse root of a matrix plays a significant role in certain linear algebra computations. An inverse p^{th} root of a matrix can be approximated by employing iterative methods.

Laasonen [10] by considering the scalar equation $ax^2 - 1 = 0$, has introduced an algorithm using Newton's method for evaluating $a^{-1/2}$. He extended the scalar iteration to matrix iteration by exploiting the matrices which have real positive eigenvalues. His proposed iteration converges quadratically and is utilized to approximate of matrix inverse square root. However, he stated that if the recursions

[*] Email: Sadeghi.usm@gmail.com

http://www.ijm2c.ir

continue indefinitely, due to round off errors, divergence occur and thus the process should be stopped as soon as difference between two successive steps no longer enhances. For computing inverse square root of a matrix, Sherif [12] proposed two iterative method based on Newton's method. He considered the matrix equation $(XA)^{-1} - X = 0$ and applied Newton's method for the purpose of approximating the solution of the matrix equation $X^2A - I = 0$. He also introduced an iteration scheme that converges to the principal inverse square root of A which has conditional stability. For the evaluation of matrix inverse p^{th} root, Lakić [9] with the assumption that A is diagonalizable, obtained some family of iterations and he proved that the iterations converges to inverse p^{th} root of a matrix. In addition, he proves for the special case, if $A \in \mathbb{C}^{n\times n}$ be a Hermitian positive definite matrix, then the proposed iterations converges to $A^{-1/p}$ which is the principal inverse p^{th} root of A.

For computation of the matrix inverse p^{th} root, one approach is to apply Newton's method for the equation $X^{-p} - A = 0$. In other words, by considering the matrix iteration

$$X_{k+1} = \frac{1}{p}\left((p+1)X_k - X_k^{p+1}A\right), \quad X_0 = A \tag{1}$$

which has conditional stability, the matrix inverse p^{th} root can be estimated. Iannazzo [7] and following him, Guo [4] have solved the instability issue by employing a special auxiliary variable and $X_0 = \frac{1}{c}I$ where c is a constant. They have proved that the proposed method is convergent and also it has good stability. Furthermore, for the evaluation of matrix inverse root, Bini et. al. [1] by assuming the iteration (1) with $X_0 = I$ have shown the convergence of the residual $R_k = I - X_k^pA$. Then they have proved that if $\|R_0\| < 1$ then $\|R_k\|$ is monotonically converges to 0 where $k \to \infty$. Furthermore, Bini et. al. [1] have shown that if all the eigenvalues of A are real and positive and $\rho(A) < p+1$ then X_k converges to any inverse p^{th} root of A. This result completely agree with Smith's idea in [14] that is if the initial matrix X_0 satisfies $\rho(I - X_0^pA) < 1$, then X_k generated by (1) converges to an inverse p^{th} root of A, where ρ denotes the spectral radius of A.

For the evaluation of the p^{th} root of a matrix, first Hoskins and Walton [6] considered the iteration

$$X_{k+1} = \frac{1}{p}\left((p-1)X_k + AX_k^{1-p}\right), \quad X_0 = A. \tag{2}$$

They only focus on symmetric positive definite matrices that X_k are all converges to $A^{1/p}$. Moreover, Smith [13] showed that this iteration is not generally numerical convergent. In addition, even is the matrix A is a symmetric positive definite, the iteration is not numerically stable unless the condition number is extremely restricted. For solving instability of the equation (2), Iannazzo [7] has introduced the auxiliary variable $N_k = AX_k^{-1}$. He shows that for initial matrix I, his proposed iterations converge to the principal roots. It should be emphasized that Iannazzo's algorithm has quadratic convergence and is known to be stable in neighborhood of solution.

More recently, Iannazzo and Meini [8] proposed the cyclic reduction algorithm to palindromic matrix polynomials. They have shown that the proposed algorithm is convergence. The proposed algorithm related to other algorithms as the evaluation and interpolation at the roots of unity of a certain Laurent matrix polynomial, the trapezoidal rule for a certain integral and an algorithm based on the finite sections of a tridiagonal block Toeplitz matrix are given. Furthermore, Soleymani et. al. [15]

introduced some iterative methods for computing the matrix sign function. They have analytically shown that the new iterations are asymptotically stable and also convergent.

In this work, we developed the conditional stable recursion introduce in [11] by present authors for computing the inverse p^{th} roots. The convergence and stability of the two new techniques which involves some special Newton's method are explored. Numerical experiments and comparisons are also performed.

2. Newton's Method

In this part, we summarize some basic concepts which was mentioned in [10, 12]. Assume a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ has no eigenvalues on $\mathbb{R}^-$. A solution X of the nonlinear matrix equation

$$X^pA - I = 0 \tag{3}$$

is called an inverse p^{th} root of A. In particular, the principal inverse p^{th} root is the unique matrix X such that $X^{-p} - A = 0$ and the eigenvalues lie in the segment

$$S = \{z \in \mathbb{C}\backslash\{0\} : -\pi/p < \arg(z) < \pi/p\}, \tag{4}$$

that is denoted by $X = A^{-1/p}$ [4]. It should be mentioned that for a nonsingular $n\times n$ complex matrix A, an inverse p^{th} root X always exists. Moreover, one approach for computing X is to use Newton's iteration to solve the nonlinear matrix equation

$$(XA)^{-1} - X^{p-1} = 0, \tag{5}$$

where $X = A^{-1/p}$ is a solution of the equation (5). It is known that change of the standard form is motivated to apply Newtons method easily. For the standard version, using the normal Newtons method give us problematic iteration [5].

For a function $F : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, the solution of $F(X) = 0$ using Newton's method

$$X_{\ell+1} = X_\ell - F'(X_\ell)^{-1}F(X_\ell), \quad \ell = 0, 1, 2, \ldots \tag{6}$$

where X_0 is given and F' is the so called "Fréchet derivative" of F [3, 12]. For a nonsingular $A \in \mathbb{C}^{n\times n}$, it is required to solve

$$F(X) \equiv (XA)^{-1} - X^{p-1} = 0 \tag{7}$$

By using the Taylor series expansion for F about X and also "Sherman-Morrison formula" [16], present authors obtained the Fréchet derivative of F as follows:

$$F'(X)H = -\left[A^{-1}X^{-1}HX^{-1} + \sum_{j=0}^{p-2} X^{p-j-2}HX^j\right] \tag{8}$$

Thus, we can stated the Newton's method for the inverse p^{th} root as: given X_0, solve the following matrix equation

$$\begin{cases} A^{-1}X_k^{-1}H_kX_k^{-1} + \sum_{j=0}^{p-2} X_k^{p-j-2}H_kX_k^j = (X_kA)^{-1} - X_k^{p-1} \\ X_{k+1} = X_k + H_k, \qquad\qquad k = 0, 1, 2, \dots \end{cases} \tag{9}$$

Newton's method requires the solution for H_k in (9). For values of greater than 2, this can be accomplished using the "Kronecker product" and "vec operator" together by using the identities (9). Finally it can be written as

$$\left[(X^T \otimes (XA)^{-1}) + \sum_{j=0}^{p-2} ((X^j)^T \otimes X^{p-j-2}) \right] \mathrm{vec}(H) = \mathrm{vec}((XA)^{-1} - X^{p-1}) \tag{10}$$

Since (10) is an $n^2 \times n^2$ linear system, storage and computation tests are expensive for it to be solved using Gaussian elimination [10, 12]. For this reason, applying of this reation should be avoided in computations. As was done in [7], to reduce the cost of solving (10), it is reasonable to assume the commutativity relation

$$X_0H_0 = H_0X_0$$

holds. Then (9) can be written as the following matrix equation

$$\begin{cases} (X_k^2A)^{-1}H_k + (p-1)X_k^{p-2}H_k = (X_k^2A)^{-1}H_k + (p-1)H_kX_k^{p-2} = (X_kA)^{-1} - X_k^{p-1} \\ X_{k+1} = X_k + H_k, \qquad\qquad k = 0, 1, 2, \dots \end{cases} \tag{11}$$

By using the relation (11), the simplified Newton's iteration is obtained as follows [11]:

$$X_{k+1} = pX_k \left((p-1)I + AX_k^p \right)^{-1}, \tag{12}$$

Suppose that A is diagonalizable. It is known that there exists a nonsingular matrix Q such that

$$Q^{-1}AQ = \Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n), \tag{13}$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of A. However, the iteration (12) can be diagonalized. If the following definition is made,

$$D_k = Q^{-1}Y_kQ, \tag{14}$$

then, from (14), we have

$$D_{k+1} = pD_k((p-1)I + \Lambda D_k^p)^{-1} \tag{15}$$

Assuming the initial matrix D_0 is diagonal, then it is known that all successive iterates D_k are also diagonal. The convergence of the diagonalized iterates can be analyzed as follows

$$D_k = \mathrm{diag}(d_1^{(k)}, \dots, d_n^{(k)}). \tag{16}$$

The iteration (16) thus becomes

$$d_i^{(k+1)} = \frac{pd_i^{(k)}}{(p-1)+\lambda(d_i^{(k)})^p}, \quad i = 0, 1, 2, \ldots \tag{17}$$

Thus, it is sufficient to only consider the scalar Newton iteration

$$z_{n+1} = \frac{pz_n}{(p-1)+az_n^p}, \quad n = 0, 1, 2, \ldots \tag{18}$$

for the inverse p^{th} root of a.

It is known that even the scalar iterations of (18) has fractal behavior. It guides us to investigate the existence of regions where Newton's iterates converge to fixed points of the function. We would like to find the initial values for which the Newton iteration converges to a special root. In particular case $p = 2$, the the solution is easy while for the cases $p > 2$ finding appropriate initial guess is very complicated. As was done in [7] we explore the behavior of the Newton iteration (18), we utilized MATLAB 2013(Ra) with a square of $90,000$ points to generate plots of the set of points for which the iteration converges to a specific root and also their boundary points of the iterates. The associated pictures for $p = 2, 3, 4,$ and 5 are shown in Fig. 1. For $p > 2$ the Newton's iterations do not have simple boundaries and have fractal behavior. It must be pointed out that we would like to obtain the principal root. Hence, the iteration must be started at a point inside the sector (4). Therefore, finding the suitable value of z belonging to the sector is difficult. Moreover, as can be seen for any point belong to nonnegative real axis, Newton's iteration (12) converges to the principal inverse p^{th} root. According to this discussion it can be concluded that for positive definite matrix, the Newton iterations (12) converges to the unique positive definite inverse p^{th} root of A where the initial guess matrix is positive definite which is presented in [6]. It should be emphasized that this result is consistent for the computation of matrix p^{th} root presented by [13].

For numerical stability of the iterations, Sadeghi et. al [11] have shown that it is required that the error amplification factor does not exceed unity in modulus. That is,

$$\left| 1 - \frac{1}{p} \sum_{r=1}^{p} \left(\frac{\lambda_i}{\lambda_j} \right)^{\frac{r-1}{p}} \right| < 1 \tag{19}$$

For the inverse square root ($p = 2$) of an Hermitian positive definite A, (19) is equivalent to $\kappa_2(A) \leqslant 9$, where $\kappa(\cdot)$ is condition number for matrix A. For the cube root of a Hermitian positive definite A, (19) requires that $\kappa_2(A) \leqslant 1.79$. It should be noticed that this condition for higher order roots are sought, and therefore the condition for numerical stability becomes more restrictive.

3. Stable Version of the Newton's Method

In this section, we present a modification of the proposed stable variant Newton method for computing matrix inverse p^{th} roots in [11]. For this purpose, an auxiliary variable $M_k = AX_k^p$ was introduced. Then they have obtained the following variant

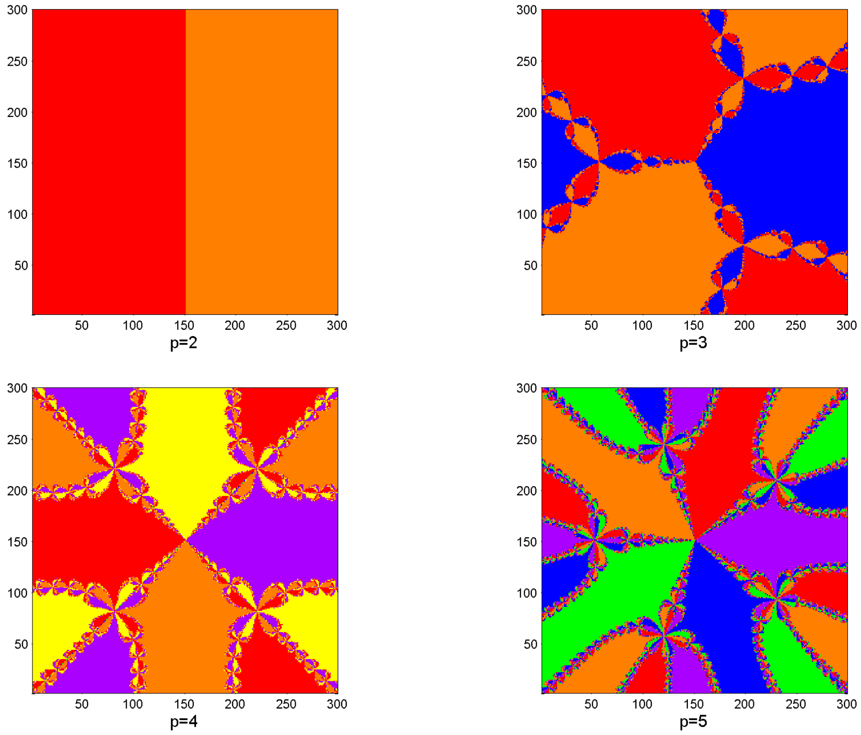

Figure 1. Fractal behavior of the Newton's iteration (18) for the solution of $ax^p - 1 = 0$.

of coupled Newton's iteration

$$\begin{cases} X_0 = I, \quad M_0 = A \\ X_{k+1} = X_k \left(\frac{(p-1)I + M_k}{p} \right)^{-1} \\ M_{k+1} = M_k \left(\frac{(p-1)I + M_k}{p} \right)^{-p} \end{cases} \tag{20}$$

It is clear that the sequence X_k converges to $A^{-1/p}$ and the sequence M_k converges to the identity matrix. Numerical experiment reveal that by modification of this scheme, considerable accuracy can be obtained. The modification employs matrix square root which can be evaluated by MATLAB command `sqrtm(A)`. First of all, for given matrix A with no non-positive real eigenvalues, the normalized matrix square root can be considered as follows

$$C = \frac{B}{\|B\|}. \tag{21}$$

where $B = A^{1/2}$. Clearly $\rho(C) \leqslant \|C\| = 1$ and consequently, the matrix C has the set of eigenvalues belongs to right half-plane and thus the spectrum of B belong to the set $\mathcal{D}$ introduced by Iannazzo [7] as follows

$$\mathcal{D} = \{z \in \mathbb{C} : \mathrm{Re} z > 0, |z| \leqslant 1\} \cup \mathbb{R}^+, \tag{22}$$

whenever $\mathbb{R}^+$ denotes the open positive real axis. For the computation of matrix p^{th} root, he has shown that if eigenvalues of the matrix A belong to $\mathcal{D}$ then the Newton's method is converges quadratically to matrix p^{th} root [7]. Here, we demonstrate that Newton's method can be applied for the equation $X^pC - I = 0$. By using this assumption the inverse p^{th} root of a given matrix A can be evaluated accurately. Assume we have $S = C^{-2/p}$ for even p and $S = C^{-1/p}$ for odd p. Hence, for even p we have

$$\begin{aligned} X &= S.\|B\|^{-2/p} \\ &= C^{-2/p}.\|B\|^{-2/p} \\ &= \left(\tfrac{B}{\|B\|}\right)^{-2/p}.\|B\|^{-2/p} \\ &= B^{-2/p} \\ &= A^{-1/p} \end{aligned}$$

and also for odd p it can be seen

$$\begin{aligned} X &= \left(S.\|B\|^{-1/p}\right)^2 \\ &= \left(C^{-1/p}.\|B\|^{-1/p}\right)^2 \\ &= \left(\left(\tfrac{B}{\|B\|}\right)^{-1/p}.\|B\|^{-1/p}\right)^2 \\ &= B^{-2/p} \\ &= A^{-1/p} \end{aligned}$$

The developed algorithm is given as follows.

Algorithm 1: `(Computing matrix inverse` p^{th} `roots using CNM)`

```
(1) Compute $B = A^{1/2}$;
(2) Put $C = \frac{B}{\|B\|}$;
(3) Using iteration (20);
        If $p$ is an even integer, compute $S = C^{-2/p}$ and set $X = S.\|B\|^{-2/p}$;
        If $p$ is an odd integer, compute $S = C^{-1/p}$ and set $X = \left(S.\|B\|^{-1/p}\right)^2$;
(4) End.
```

It should be emphasized that in this algorithm for the computation of the matrix power $-p$, we first evaluate the binary power technique of the matrix and then apply the inversion of the matrix. We have carried out the recursion till the stop criteria is hold. Notice that the scheme which computes the matrix square root is highly effective in obtaining substantial accuracy. In our implementation, the `sqrtm(A)` presented in MATLAB which uses the Schur method is employed.

In this part, an analysis to show the stability of the proposed method is presented. According to [2] an iteration $X_{k+1} = g(X_k)$ is defined be stable in a neighborhood of a solution $X = g(X)$ if the associated error matrices $E_k = X_k - X$ satisfy

$$E_{k+1} = L(E_k) + O(\|E_k\|^2), \tag{23}$$

where L is a linear operator with bounded powers. Now we provide an analysis for the stability of the proposed method. To perform an error analysis, error matrices $E_k = X_k - A^{-1/p}$ and $F_k = M_k - I$ are introduced and all the terms that are quadratic in their errors are removed.

From $M_k = I + F_k$, we have

$$\begin{aligned} \left(\tfrac{(p-1)I+M_k}{p}\right)^{-p} &= \left(\tfrac{(p-1)I+I+F_k}{p}\right)^{-p} \\ &\approxeq \left(I + \tfrac{F_k}{p}\right)^{-p} \\ &\approxeq I - F_k \end{aligned} \tag{24}$$

The symbol $\approxeq$ denoted equality up to second order terms the errors. The relation for the errors is thus,

$$\begin{pmatrix} E_{k+1} \\ F_{k+1} \end{pmatrix} \approxeq \begin{pmatrix} I & -\frac{1}{p}A^{-1/p} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} E_k \\ F_k \end{pmatrix} = L \begin{pmatrix} E_k \\ F_k \end{pmatrix}. \tag{25}$$

Since the coefficient matrix L is such that $L^2 = L$ and thus has powers which are bounded, it follows that the iteration is stable.

4. Numerical Experiments

In this section, we supplement the theory which has been developed so far with several numerical implementations. All the computations have been done using MATLAB 2013(Ra). We also used Higham's Matrix Function Toolbox [5]. In addition, the accuracy is measured by means of the size of:

$$e(\widehat{X}) = \|A\widehat{X}^p - I\|, \tag{26}$$

$$Res(\widehat{X}) = \frac{\|A\widehat{X}^p - I\|}{\|A\|}, \tag{27}$$

and

$$\rho_{A^{-1}}(\widehat{X}) = \frac{\| A\widehat{X}^p - I \|}{\| \widehat{X} \| \| \sum_{i=0}^{p}(\widehat{X}^i A \widehat{X}^{p-i} - I) \|}, \tag{28}$$

where $\widehat{X}$ is the computed inverse p^{th} roots of A and $\| \cdot \|$ is any norm (In our tests we use the Frobniuos norm). Note that $\rho_{A^{-1}}(\widehat{X})$ was presented by Guo in [4].

Test 1. First example made considering an 4×4 symmetric positive definite well-conditioned matrix with condition number $\kappa(A) = 10$. This matrix is defined by

$$A = \begin{pmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{pmatrix}.$$

Numerical experiments are presented to compare the behavior of our proposed methods and other schemes. Our method is denoted by CNM which means the coupled Newton's method. Other methods include the $A^{-1/p} = \exp(-\frac{1}{p}\log(A))$ based method [5] which can be calculated using the functions expm and logm in MATLAB, and also Schur Newton method [4]. Results are arranged in Tables 1 to 3. Furthermore, we have implemented the values of $p = 5^{\ell}$ for $\ell = 1, 2, 3, 4$ and 5. According to tables, in comparison to well-known scheme, it can be seen that the the proposed method has comparable accuracy.

Table 1. **Comparing error $e(\widehat{X})$ among different methods for Test 1.**

p	CNM	$\exp(-\frac{1}{p}\log(A))$	Schur Newton method
5	8.2623e-13	1.8544e-15	4.9340e-15
25	3.7221e-11	8.4099e-15	4.2026e-14
125	7.1852e-11	6.2919e-14	1.4384e-13
625	8.1553e-11	2.2286e-13	6.9880e-13
3125	8.2415e-11	5.3474e-13	3.5952e-12

Table 2. **Comparing error $Res(\widehat{X})$ among different methods for Test 1.**

p	CNM	$\exp(-\frac{1}{p}\log(A))$	Schur Newton method
5	7.2465e-14	1.6264e-16	4.3274e-16
25	3.2645e-12	7.3759e-16	3.6859e-15
125	6.3018e-12	5.5184e-15	1.2616e-14
625	7.1527e-12	1.9546e-14	6.1289e-14
3125	7.2283e-12	4.6899e-14	3.1532e-13

Table 3. **Comparing error $\rho_{A^{-1}}(\widehat{X})$ among different methods for Test 1.**

p	CNM	$\exp(-\frac{1}{p}\log(A))$	Schur Newton method
5	3.7886e-15	8.5037e-18	2.2625e-17
25	6.1883e-15	1.3982e-18	6.9873e-18
125	4.6837e-16	4.1015e-19	9.3768e-19
625	2.1179e-17	5.7879e-20	1.8148e-19
3125	8.5545e-19	5.5505e-21	3.7318e-20

Test 2. In this example, six matrices A, B, C, D, E, and F are assumed. These matrices that are either well conditioned or ill conditioned can be utilized to compare the accuracy of the proposed methods. The result are reported on Table 4 for CNM. From result, it can be concluded that the accuracy is not feasible for the ill conditioned matrices. For instance, for the matrices A, D and F which are ill conditioned, the errors are larger than other matrices that are well conditioned.

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}, \qquad B = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}, \qquad C = \begin{pmatrix} 4 & 1 & 1 \\ 2 & 4 & 1 \\ 0 & 1 & 4 \end{pmatrix}$$

$$D = \begin{pmatrix} 1.00 & 0.50 & 0.33 \\ 0.50 & 0.33 & 0.25 \\ 0.33 & 0.25 & 0.20 \end{pmatrix}, \qquad E = \begin{pmatrix} 5+i & 2+i & 3i \\ 2+i & 5+i & 4+1i \\ 1-2i & 3-2i & 6-2i \end{pmatrix}, \qquad F = \begin{pmatrix} -1 & -2 & 2 \\ -4 & -6 & 6 \\ -4 & -16 & 13 \end{pmatrix}$$

Table 4. **Comparison the errors among different matrices using CNM in Test 2.**

Matrix	$\kappa(A)$	p	$e(\widehat{X})$	$Res(\widehat{X})$	$\rho_{A^{-1}}(\widehat{X})$
		5	3.4043e-14	4.2890e-15	2.0457e-16
A	61.9839	49	6.3178e -01	7.9597 e-02	4.9429e-05
		1982	6.4008 e-01	8.0642 e -02	3.0789e-08
		5	6.8056e-15	5.2822e-15	7.9390e-17
B	2.5324	49	5.7838e-14	4.4891e-14	9.0072e-18
		1982	1.7984e-12	1.3958e-12	1.7608e-19
		5	6.1515e-15	8.2203e-16	5.4832e-17
C	2.4642	49	3.7884e-14	5.0625e-15	3.0728e-18
		1982	1.9488e-12	2.6042e-13	9.5342e-20
		5	9.0266 e-01	6.3854 e-01	6.8533 e-03
D	524.0568	49	9.4211 e-01	6.6645 e-01	1.3512e-004
		1982	9.4576e-01	6.6903 e-01	8.7375e-008
		5	2.7361e-14	2.2644e-15	1.9394e-16
E	9.6468	49	1.7626e-13	1.4588e-14	1.1315e-17
		1982	2.0755e-12	1.7177e-13	8.0121e-20
		5	2.6757e 00	1.1541e-01	2.8635e-02
F	204.7808	49	2.7654 e 00	1.1924e-01	3.0032e-04
		1982	2.7700 e 00	1.1943e-01	1.7981e-07

Test 3. In this example we would like to investigate the impact of the dimension of matrices in obtained accuracy. For this purpose two different matrices A and B which are defined as

$$A = (a_{ij}) = \begin{cases} 0 & \text{if} \quad i < j \\ 1 & \text{if} \quad i = j \\ -1 & \text{if} \quad i > j \end{cases}, \qquad B = (b_{ij}) = \begin{cases} 0.1 & \text{if} \quad i < j \\ 1 & \text{if} \quad i = j \\ -0.1 & \text{if} \quad i > j \end{cases}.$$

are supposed. We have approximated the inverse 67^{th} root of the presented matrices and we reported the accuracy for different dimension. Result are illustrated in Table 5. Moreover, by increasing the dimension of matrix using CNM, error also soared for the matrix A while for the matrix B, increasing dimension does not have any affect to increase the error.

Table 5. **Comparison errors for evaluating $A^{-1/67}$ using CNM in Test 3.**

Matrix	N	$e(\widehat{X})$	$Res(\widehat{X})$	$\rho_{A^{-1}}(\widehat{X})$
	3	2.5767e-14	1.05196e-14	3.0723e-18
	4	5.9520e-14	1.8821e-14	5.1645e-18
	5	6.6576e-14	1.71899e-14	3.5564e-18
	6	2.8578e00	6.2363e-01	4.3538e-05
A	7	4.2699e00	8.0693e-01	3.6197e-05
	8	4.4771e00	7.4619e-01	2.8531e-05
	9	3.7691e00	5.6187e-01	2.3040e-05
	10	3.2159e00	4.3363e-01	2.2451e-05
	11	3.1960e00	3.9340e-01	2.8256e-05
	3	2.4397e-14	1.3946e-14	2.0537e-18
	4	5.3564e-14	2.6389e-14	2.9198e-18
	5	8.2451e-14	3.6157e-14	3.2037e-18
	6	7.5790e-14	3.01958e-14	2.2301e-18
B	7	2.4551e-13	9.0131e-14	5.7032e-18
	8	1.6313e-13	5.5759e-14	3.0839e-18
	9	1.3038e-13	4.1822e-14	2.0527e-18
	10	1.6987e-13	5.1454e-14	2.2683e-18
	11	1.6222e-13	4.6636e-14	1.8643e-18

5. Conclusion

In this work, we have developed an algorithm using normalization of an arbitrary matrix A which does not have negative eigenvalues for evaluating the inverse roots of matrices for integer p. This method that has considerable accuracy in comparison other popular method, can compute the matrix inverse root. Numerical experiments indicate that the proposed method is sensitive about the condition number of a matrix. In other words, if the matrix A is ill conditioned then the accuracy of the method will be reduced. Furthermore, numerical tests reveal that for some matrices, by increasing the dimension of the matrix errors will be enhanced suddenly for CNM.

References

[1] D. A. Bini, N. J. Higham and B. Meini, *Algorithms for the matrix pth root*, Numer. Algorithms, **39** (2005) 349-378.

[2] S. H. Cheng, N. J. Higham, C. S. Kenney and A. J. Laub, *Approximating the logarithm of a matrix to specified accuracy*, SIAM J. Matrix Anal. Appl., **2** (2001) 1112-1125.

[3] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Uncon- strained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Clifs, NJ, USA, (1983).

[4] C.-H. Guo and N. J. Higham. *A Schur-Newton method for the matrix pth root and its inverse*, SIAM J. Matrix Anal. Appl., **28** (2006) 788-804.

[5] N. J. Higham, *The Matrix Function Toolbox.* http://www.ma.man.ac.uk/ higham /mctoolbox (2009).

[6] W. D. Hoskins and D. J. Walton, *A faster, more stable method for computing the pth roots of positive definite matrices*, Linear Algebra and Appl., **26** (1979) 139-163.

[7] B. Iannazzo, *On the Newton method for the matrix pth root*, SIAM J. Matrix Anal. Appl., **28** (2006) 503-523.

[8] B. Iannazzo, B. Meini, *The palindromic cyclic reduction and related algorithms*, Calcolo, (2014), DOI: 10.1007/ s10092-014-0106-z]

[9] S. Lakić, *On the computation of the matrix kth root,* ZAMM Z. Angew. Math. Mech., **78** (1989) 167-172.

[10] P. Laasonen, *On the iterative solution of the matrix equation* $AX^2 - I = 0$, Math. Tables Aids Comput., **12** (1958) 109-116.

[11] A. Sadeghi, A. I. Md. Ismail and A. Ahmad, *A special newton iteration for computing inverse matrix roots*, 2011, Apll. Comput. Math., **10** (2011) 472-483.

[12] N. Sherif, *On the computation of a matrix inverse square root*, Computing, **46** (1991) 295-305.

[13] M. I. Smith, *A Schur algorithm for computing matrix pth roots*, SIAM J. Matrix Anal. Appl., **24** (2003) 971-989.

[14] R. A. Smith, *Infinite product expansions for matrix nth roots*, J. Austral. Math. Soc., **8** (1968) 242-249.

[15] F. Soleymani, E. Tohidi, S. Shateyi and F. Khaksar Haghani, *Some matrix iterations for computing matrix sign function*, J. Appl. Math., Article ID 425654, 9 pages, (2014).

[16] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, London, (1973).