Ilorin Journal of Science
Volume 4, Number 1, 2017, pp. 1 – 16 (Printed in Nigeria)
ISSN: 2408 – 4840 © 2017 Faculty of Physical Sciences, University of Ilorin
https://doi.org/10.54908/iljs.2017.04.01.001

ILORIN
JOURNAL OF SCEINCE

ILJS-17-026

Extracting P-th Root of a Matrix with Positive Eigenvalues via Newton and Halley's Methods

Uwamusi*, S. E.

Department of Mathematics, University of Benin, Benin City, Edo State, Nigeria.

Abstract

This paper presents a class of methods for finding zeros of a polynomial of single variable as a prelude to deriving Newton and Halley's iteration methods for computing the p-th root of a matrix where $p \geq 2$. We use the LU Factorization and Givens QR–Cholesky Decomposition to invert the matrix which appears in the Newton and Halley's methods. The givens QR algorithm when runs to completion could provide the eigenvalues of the symmetric matrices free of charge without additional task. The nature of distribution for the eigenvalues of matix A during iteration phases is discussed. Numerical illustration is demonstrated with the described procedures which simultaneously provide the p-th root of a matrix and its inverse whose eigenvalues are not in the left-hand side in the extended real line. We compare note with results obtained from Guo and Higham who used polar decomposition in their approach for the cases p=12 and 52 respectively. It was observed that for for $p \geq 10$, the p-th root tends to a diagonal matrix. As a special case for p = 2, we computed Square root of Newton iteration with Euler-Chevbyshev method due to Lakic and Petkovic for the lower bound of this class of methods.

Keywords: Polynomial zeros, p-th root of a matrix, Newton's method, Halley's method, distribution of eigenvalues, LU decomposition, QR algorithm, polar decomposition.

1. Introduction

The paper presents numerical methods for Newton and Halley's iterations and their variants for approximating p-th root of a matrix for the case $p \geq 2$ provided such eigenvalues of the matrix are not on the R^- axis. The need to compute the p-th root of a diagonalizable matrix with positive eigenvalues is one major task often faced by researchers in numerical computation and engineering practices. Popular in these methods is the Newton Schultz Hyper Power method which can be amenable to Hensenl's and p- adic lifting processes Bini and Pan (1994) a useful tool in aerospace computation for the Covariance matrix. As an example to this discussion is the matrix square root information filters and smoothers, which play crucial role in the solution to discrete time and Kalman filter problem Sharp and Allen

*Corresponding Author: Uwamusi, S. E.
Email: stephen_uwamusi@yahoo.com

(2006). Computing the principal p-th root of a matrix is often met with stiff resistance but at the same time very crucial in the formation of product of a matrix square root and a vector in the dynamical state space differential equations (Bjorck, 2009). Besides, other application area of principal p-th root of a matrix is the need also to computing square root of a matrix occurring in the Wierner operator with covariance matrix, is a very important tool for discussing the solution process to stochastic parabolic partial differential equation (Yan, 2005). It follows that the quest for studying computation of principal p-th root of a positive definite matrix is an interesting topic in numerical analysis hence, we aim to provide a simplified approach to this problem.

We thus introduce into the discussion, the integral representation of a matrix square root defined in the form:

$A^{\frac{1}{2}} = \frac{2}{\pi}\int_0^{\infty}\left(t^2 I\_A\right)^{-1} dt$, where for the p-th root case, it is given in the form:

$$A^{\frac{1}{2}} = \frac{p\sin\left(\frac{\pi}{p}\right)}{\pi} A\int_0^{\infty}\left(t^2 I + A\right)^{-1} dt .$$

Given the scalar equation of degree $p = 2$ in the form $x^2 - a = 0$, the Newton iteration formula for matrix square root can be developed as follows:

$$X_{k+1} = \frac{1}{2}\left(X_k + Y_k^{-1}\right), X_0 = A, Y_0 = I, (k = 0,1,2,...,.) \quad . \tag{1}$$

Following this technique which is expressed in Equation (1), Lakic and Petkovic (1998) obtained a cubically convergent matrix square root formula derived from the Schultz Hyper method for Newton iteration in the form (Altman, 1960):

$$\begin{aligned} X_{k+1} &= \frac{1}{3} X_k\left(I + 8(I + 3Y_k X_k)^{-1}\right), X_0 = A; \\ Y_{k+1} &= \frac{1}{3}\left(I + 8\left(I + 3Y_k X_k\right)^{-1}\right)Y_k , Y_0 = I \end{aligned} \quad , \tag{2}$$

The fact that $X_{k+1} \to A^{\frac{1}{2}}$ while $Y_{k+1} \to A^{-\frac{1}{2}}$ simultaneously for $k \to \infty$ is a numerical problem that is worth giving attention to.

A close resemblance to matrix square root iteration of Equation (2) is the Euler-Chevbyshev method due to Altman (1960) in the form:

Algorithm 1.1 (Euler-Chevbyshev method (Altman, 1960).

Given a matrix $A \in R^{n\times n}$, whose real eigenvalues are not on R^{-}, it is required to compute $X_k \to A^{\frac{1}{2}}$ as $k \to \infty$.

(1) Define $C = \dfrac{A}{\|A\|}$, $R_0 = I, S_0 = C$,

(2) $R_{k+1} = R_k\left(\dfrac{3}{8}I + \dfrac{3}{4}S_k\left(I - \dfrac{1}{6}S_k\right)\right)$,

(3) $S_{k+1} = S_k\left(\dfrac{3}{8}I + \dfrac{3}{4}S_k\left(I - \dfrac{1}{6}S_k\right)\right)^{-2}$,

(4) $X_k = \sqrt{\|A\|}\, R_k$.

It must be noted that the number of square roots may be finite, zero, or infinite. The most important for applications of the square roots is the principal square root whose eigenvalues are all in the right half plane. The following facts are introduced here as a follow up in our presentation.

Fact 1: If A has Principal square root and A^* be the adjoint of A with respect to an arbitrary scalar product (bilinear or sesquilinear). Then:

(i) $(A^*)^{\frac{1}{2}}$ and $(A^{-1})^{\frac{1}{2}}$ both exist, (ii) $(A^*)^{\frac{1}{2}} = \left((A^{\frac{1}{2}}\right)^*$ and $\left(A^{-1}\right)^{\frac{1}{2}} = \left(A^{\frac{1}{2}}\right)^{-1}$.

In what follows we draw the synergy of the theory of square root of a matrix to be extended to cover the automorphism group, Lie group and Jordan algebra. Therefore, the structured principal root has the following characteristics.

Fact 2: Let G, L and J denote the automorphism group, Lie group and Jordan algebra, respectively of an arbitrary scalar product. Then

(i) $A \in G \Rightarrow A^{\frac{1}{2}} \in G$, (ii) $A \in L \Rightarrow A^{\frac{1}{2}} \notin L$, (iii) $A \in J \Rightarrow A^{\frac{1}{2}} \in J$.

(ii) In the case of a Hermittian matrix, with A^* denoting the adjoint of A with respect to unitary scalar product, then

(iii) (i) $(A^*)^{\frac{1}{2}}$ exists and $(A^*)^{\frac{1}{2}} = (A^{\frac{1}{2}})^*$, (ii) if A is non-singular, then $(A^{-1})^{\frac{1}{2}}$ exists and $(A^{-1})^{\frac{1}{2}} = (A^{\frac{1}{2}})^{-1}$.

Therefore in forming the product of two square root of matrices $X_{k+1}^* X_{k+1}$, the following technique applies in the sense of Higham *et al*. (2004):

$$\begin{aligned} X_{k+1}^* X_{k+1} - I &= \frac{1}{4}\left[X_k^* X_k + X_k^* X_k^{-*} + \left(X_k^{-*}\right)^* X_k + \left(X_k^{-*}\right)^* X_k^{-*} - 4I\right] \\ &= \frac{1}{4}\left[X_k^* X_k + I + I + X_k^{-*} X_k^* - 4I\right] \\ &= \frac{1}{4}\left(X_k^* X_k\right)^{-1}\left(\left(X_k^* X_k\right)^2 - 2X_k^* X_k + I\right) \end{aligned}$$

.

In that case, the equality $X_{k+1}^* X_{k+1} - I = \frac{1}{4}\left(X_k^* X_k\right)^{-1}\left(X_k^* X_k - I\right)^2$ can be detailed.

Experimental evidence suggested that method of Equation (1) may converge at a very slow rate if there were present eigenvalues of large magnitude or very small values for the matrix A. To remedy this problem, it is suggested that determinant scaling or something of its equivalent form can be used in method of Equation (1). Thus scaled Newton method is then written as:

$$X_{k+1} = \frac{1}{2}\left(c_k X_k + \frac{1}{c_k} Y_k^{-1}\right), \tag{3}$$

The c_k appearing above in equation (1.2) is defined to be any of the following

(i) $c_k = \left|\frac{\det(X_k)}{\det\left(A^{\frac{1}{2}}\right)}\right|^{\frac{-1}{n}}$,where $\det(A^{\frac{1}{2}})$ should be computed as $\det(A)^{\frac{1}{2}}$,

(ii) $c_k = \left|\det(X_k)\det(Y_k)\right|^{\frac{-1}{2n}}$, (iii) $c_k = \sqrt{\frac{\rho\left(X_k^{-1}\right)}{\rho(X_k)}}$, (iv) $c_k = \sqrt{\frac{\left\|X_k^{-1}\right\|}{\|X_k\|}}$

More fundamentally, the Scaled Denver-Beaver method is expressed in Equation (2) and is defined as:

$$\begin{aligned} X_{k+1} &= \frac{1}{2}\left(c_k X_k + c_k^{-1} Y_k^{-1}\right), X_0 = A; \\ Y_{k+1} &= \frac{1}{2}\left(c_k Y_k + c_k^{-1} X_k^{-1}\right), Y_0 = I \end{aligned} \tag{4}$$

Further insight into this presentation is that, it should not be confused with the three- term recurrence relation earlier obtained in Higham *et al*. (2004) for method of Equation (4), and is written in the form:

$$\begin{aligned} M_{k+1} &= \frac{1}{2}\left(I + c_k^2 M_k + c_k^{-2} M_k^{-1}\right), M_0 = A, \\ X_{k+1} &= \frac{1}{2} c_k X_k \left(I + c_k^{-2} M_k^{-1}\right), X_0 = A, \\ Y_{k+1} &= \frac{1}{2} c_k Y_k \left(I + c_k^{-21} M_k^{-1}\right), Y_0 = I, \end{aligned} \tag{5}$$

$(k = 0,1,...,.)$.

In the absence of scaling in Equation (5), we give the product form for the Denver Beaver iteration developed by the equations Higham *et al*. (2004), Uwamusi (2005):

$$\begin{aligned} M_{k+1} &= \frac{1}{2}\left(I + \frac{M_k + M_k^{-1}}{2}\right), \quad M_0 = A, \\ X_{k+1} &= \frac{1}{2} X_k \left(I + M_k^{-1}\right), \qquad X_0 = A, \\ Y_{k+1} &= \frac{1}{2} Y_k \left(I + M_k^{-1}\right), \qquad Y_0 = I, \end{aligned} \tag{6}$$

$(k = 0,1,2,...,.)$.

The following observations are worth noting Higham *et al.* (2004) for the Denver-Beaver iteration, a modified version of Newton iteration formula for finding square root of a positive definite matrix in the presentation of Equation (6). The $M_k \to I$ as $k \to \infty$, with resultant consequences that:

$$\left(X_{k+1}-A^{\frac{1}{2}}\right)\left(X_{k+1}+A^{\frac{1}{2}}\right)^{-1}=\left(\left(X_k-A^{\frac{1}{2}}\right)\left(X_k+A^{\frac{1}{2}}\right)^{-1}\right)^2, \tag{7}$$

$$\left(Y_{k+1}-A^{-\frac{1}{2}}\right)\left(Y_{k+1}+A^{-\frac{1}{2}}\right)^{-1}=\left(\left(Y_k-A^{-\frac{1}{2}}\right)\left(Y_k+A^{\frac{-1}{2}}\right)^{-1}\right)^2, \tag{8}$$

$$\left(M_{k+1}^{\frac{1}{2}}-I\right)\left(M_{k+1}^{\frac{1}{2}}+I\right)^{-1}=\left((M_k-I)(M_k+I)^{-1}\right)^2 . \tag{9}$$

One main advantage in the described inverse scaling and squaring in the Denver-Beaver method is that, we are able to compute the logarithm of a matrix almost inexpensively. Furthermore, using the already computed results that $X_k \to A^{\frac{1}{2}}$ as $Y_k \to A^{-\frac{1}{2}}$, then it could be written that $\log A = 2\log X_k - \log X_k Y_k$, and because the product $X_k Y_k \to I$ with additional quality in that $\log X_k Y_k \to O$, respectively simultaneously. The bound norms then satisfy the inequality:

$$\left\|X_{k+1}Y_{k+1}-I\right\| \le \left\|X_{k+1}-X_k\right\|\left\|Y_{k+1}-Y_k\right\|.$$

In addition to the aforementioned facts, it also holds that $\det(X_k)\det(Y_k)=o$. After all these, for the matrix A, we give the bound for the condition number for the matrix square root in the form:

$$K_{sqrt}(A^{\frac{1}{2}}) = \frac{\left\|A^{-1}\right\|_2^{\frac{1}{2}}\|A\|_F}{2}\frac{\|A\|_F}{\left\|A^{\frac{1}{2}}\right\|_F} \le K_{sqrt}(X) \quad , \tag{10}$$

$$\frac{1}{2n^{\frac{3}{2}}}K_F\left(A^{\frac{1}{2}}\right) \le K_{sqrt}\left(A^{\frac{1}{2}}\right) \le \frac{1}{2}K_F(A^{\frac{1}{2}}) . \tag{11}$$

We give bounds for the singular values of the matrix A in line with ideas due to Bini and Pan (1994) analogous to spectral radius of the matrix A.

Theorem 1.1: Bini and Pan (1994). Let $A \in R^{n\times n}$ be given and suppose that $\left\|I - A^H A\right\| \le \alpha < 1$ Then, for a full rank matrix A we have the following:

(i) $\sqrt{1-\alpha} \le \sigma_i(A) \le \sqrt{1-\alpha}$,

(ii) $\frac{1}{\sqrt{1-\alpha}} \le \sigma_i(A) \le \frac{1}{\sqrt{1-\alpha}}$,

(iii) $\sqrt{1-\alpha} \le \|A\| \le \sqrt{1+\alpha}$,

(iv) $\frac{1}{\sqrt{1-\alpha}} \le \|A^+\| \le \frac{1}{\sqrt{1+\alpha}}$.

2. Newton's and Halley's methods for the pth root of a Symmetric matrix.

Following the methods of Uwamusi (2005a), we derive a class of Newton and Halley's methods for zeros of the single variable polynomial equation

$F(x) = 0$, (12)

where $F \in C[a,b]$ and possessed the necessary derivatives in the neighborhood zeros of F. For convenience, F has been specified to be a function of x with simple zero, ξ. Thus, f $h(\xi) \neq 0$ where it was derived that h was a reduced polynomial of F, it will hold that Equation (13) assumes the form:

$F = (x-\xi)h$. (13)

Further taking logarithm of both sides of Equation 2.2, and differentiating with respect to x it will be that:

$$\frac{F'}{F} = \frac{1}{x-\xi} + \frac{h'}{h}. \tag{14}$$

After some serious but rigorous analysis, a family of iteration formula was obtained by Hansen and Patrick (1977) in the form:

$$x_i^{(k+1)} = \theta\left(x_i^{(k)}\right), (k = 0,1,...,.) , \tag{15}$$

where it is understood that θ is a rational map described by the equation:

$$\theta\left(x_i^{(k+1)}\right) = x_i^{(k)} - \frac{(\alpha+1)F}{\alpha F' \pm \left[(F')^2(\alpha+1)FF''\right]^{\frac{1}{2}}}. \tag{16}$$

The α is a variable parameter which rules the governing Equation (15) and it is based on the approximation of second order derivative of h to the square of its first derivative. In what

follows we aim to construct functional iterative methods out of Equation (15) for useful purpose.

Firstly, by setting $\alpha = 0$ in equation 2.5 and using this in method 2.4 we obtain that Ostrowski formula (Ostrowski, 1966):

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{\pm\left[\left(F^{/}\right)^2 - F^{//}\right]^{\frac{1}{2}}}, \quad (k = 0,1,...,.) . \tag{17}$$

Again, letting $\alpha = \infty$, a limiting case of Equation 2.4 is arrived at, in which case the Newton's quadratic order of convergence is obtained:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{F^{/}}, \quad (k = 0,1,...,.) . \tag{18}$$

Besides, we further take $\alpha = 1$ to obtain:

(Euler's method)

$$x_i^{(k+1)} = x_i^{(k)} - \frac{2F}{F + \left[\left(F^{/}\right)^2 - 2FF^{//}\right]^{\frac{1}{2}}}. \tag{19}$$

On the other hand, we take $\alpha = -1$, an iterative formula of Halley's type is obtained:

(Halley's method)

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{F^{/} - \frac{FF^{//}}{2F^{/}}}, \quad (k = 0,1,...,.) . \tag{20}$$

Equivalently, method of Equation (20) can be rewritten in the form:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F/F^{/}}{1 - (\frac{1}{2} FF^{//} / \left(F^{/}\right)^2)}, \quad (k = 0,1,...,.) . \tag{21}$$

To obtain the Laguerre method in the sense of Ostrowski (1966), Petkovic and Trickovic (1995), we have to set $\alpha = \frac{1}{n-1}$ and is given by the equation (Laguerre's method):

$$x_i^{(k+1)} = x_i^{(k)} - \frac{nF}{F^{/} \pm \left[(n-1)^2 F^{/2} - n(n-1)FF^{//}\right]^{\frac{1}{2}}}, \ (k = 0,1,...,) \ . \tag{22}$$

We shall move two more steps further in this class of methods of Equation (15) by respectively taking $\alpha = -\frac{1}{2}$ and $\alpha = \frac{1}{2}$ in the form:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{-\frac{1}{2}F}{\left(-\frac{1}{2}F^{/} \pm \left[\left(F^{/}\right)^2 - \frac{1}{2}FF^{//}\right]^{\frac{1}{2}}\right)} \ ; \ \left(\alpha = -\frac{1}{2}\right), \tag{23}$$

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{F^{/} - \frac{FF^{//}}{2F^{/}} + \frac{F^{/2}F^{//2}}{5F^{/2}}} ; \ \left(\alpha = \frac{1}{2}\right). \tag{24}$$

In what follows we create an implicit formula out of family of methods of Equation (15) in the sense of Uwamusi (2005a) by substituting the Halley's Correction formula $H(x_i^{(k)}) = \frac{F}{F^{/} - \frac{FF^{//}}{2F^{/}}}$ of Equation (2.8) into Taylor Series expansion:

$$0 = F\left(x_i^{(k)}\right) + \left(x_i^{(k+1)} - x_i^{(k)}\right)F^{/}\left(x_i^{(k)}\right) + \frac{1}{2}\left(x_i^{(k+1)} - x_i^{(k)}\right)^2 F^{//}\left(x_i^{(k)}\right), \tag{25}$$

We then have an iterative formula:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{F^{/}}\left[1 + \frac{2F\,F^{//}}{\left(2F^{/2} - FF^{//}\right)^2}\right]. \tag{26}$$

Now we make the following assumptions concerning Equation (26) that $|F|$ is reasonably small enough and that the extra term $FF^{//}$ in the denominator be ignored, we then write an iterative formula of Chevbyshev in the form:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F}{F^{/}}\left(1 + \frac{FF^{//}}{2F^{/2}}\right), \quad (k = 0,1,...,) \ . \tag{27}$$

By using ideas due to Uwamusi (2005a), we create a forward phasing implicit formula out of Equation (27) by using that:

$$F^{//}(x_i^{(k)}) = \frac{F^{/}\left(x_i^{(k)}\right) - F^{/}(x_i^{(k-1)})}{x_i^{(k)} - x_i^{(k-1)}} \quad , \tag{28}$$

If we take into cognizance that the Laguerre's disk

$$\left|x_i^{(k)} - x_i^{(k-1)}\right| \leq n\left(\left|\frac{F(x_i^{(k)}}{F^{/}(x_i^{(k)}}\right|\right), \tag{29}$$

in the sense of Breass and Hadeler (1973), see also Wasilkowski (1980) ,contains at least one zero of F ,and using this connection of optimal x , we relate the inequality in Equation (29) with equality wherein the Laguerre's disk is made to be in contact with a Circle such that:

$$\left|x_i^{(k)} - x_i^{(k-1)}\right| = n\left(\left|\frac{F(x_i^{(k)}}{F^{/}(x_i^{(k)}}\right|\right). \tag{30}$$

As a result of Equation (30) we then write that

$$F^{//}\left(x_i^{(k+1)}\right) = \frac{F^{/}(x_i^{(k)}) - F^{/}(x_i^{(k-1)})}{n\dfrac{F\left(x_i^{(k)}\right)}{F^{/}\left(x_i^{(k)}\right)}}. \tag{31}$$

By substituting Equation (31) into the Chevbyshev formula of Equation (27), we obtain another iterative formula earlier reported in Uwamusi (2005b):

$$x_i^{(k+1)} = x_i^{(k)} - \frac{F\left(x_i^{(k)}\right)}{F^{/}(x_i^{(k)}}\left[1 + \frac{(F^{/}(x_i^{(k)}) - F^{/}\left(x_i^{(k-1)}\right))}{2n\dfrac{F\left(x_i^{(k)}\right)}{F^{/}\left(x_i^{(k)}\right)}}\right]. \tag{32}$$

By realizing further that we can instead rewrite $F^{//}(x_i^{(k)})$ as

$$F^{//}(x_i^{(k)}) = \frac{F^{/}(x_i^{(k)}) - F^{/}(x_i^{(k+1)})}{x_i^{(k)} - x_i^{(k+1)}}. \tag{33}$$

We have implicit phasing iterative formula obtained from Chevbyshev method

$$x_i^{(k+1)} = x_i^{(k)}\left[1+\frac{F^{/}\left(x_i^{(k)}\right)-F^{/}(x_i^{(k+1)})}{2nF^{/}\left(x_i^{(k)}\right)}\right], \quad (k=0,1,...,.). \tag{34}$$

The methods under consideration were obtained from solving the scalar polynomial equation of order p>1 in the form: $x^p - a = 0$.

Thus the formulation for Newton and respectively, Halley's method (Uwamusi (2005a, Hansen and Patrick 1975, Ostrowski 1966, Petkovic and Trickovic 1995, Uwamusi 2005b, Breass and Hadeller 1973, Wasilkowski 1980, Innazzo 2006) can be written in the form:

Newton method:

$$x_{k+1} = \frac{1}{p}\left((p-1)x_k + ax_k^{1-p}\right), x_0 = 1. \tag{36}$$

Halley's method:

$$x_{k+1} = x_k\left(\frac{(p-1)x_k^p + (p+1)a}{(p+1)x_k^p + (p-1)a}\right), x_0 = 1. \tag{37}$$

In matrix formulation, the p-th root of a matrix equation $X^p - A = 0$ in the form:

Newton matrix iteration:

$$X_{k+1} = \frac{1}{p}\left((p-1)X_k + AX_k^{1-p}\right), X_0 = I. \tag{38}$$

Halley's matrix pth root iteration (Ostrowski 1966, Petkovic and Trickovic 1995, Innazzo 2006, Guo and Higham 2006, Higham et al 2005, Cheng et al 2001, Guo 2009) is written in the form:

$$X_{k+1} = X_k\left((p+1)X_k^p + (p-1)A\right)^{-1}\left((p-1)X_k^p + (p+1)A\right), X_0 = 1. \tag{39}$$

The eigenvalues of X lie in the segment $z: -\frac{\pi}{p} < \arg(z) < \frac{\pi}{p}$.

It should be noted that the inverse counterpart of equation (38) for finding p-th root of equation $X^{-p} - A = O$, is expressed in the form:

$$X_{k+1} = \frac{1}{p}[(p+1)X_k - X_k^{p+1}A], X_0 A = AX_0 .$$

A variant of Newton's *pth* root iteration for the matrix equation given above is presented called the stable forms for Newton and Halley's methods below:

Stable Newton method Higham *et al*. (2005), Cheng *et al*. (2001):

$$X_0 = I,\ N_0 = A;\quad \text{for } k = 0,1,2,..,$$

Compute:

$$X_{k+1} = X_k\left(\frac{(p-1)I + N_k}{p}\right), \tag{40}$$

where

$$N_{k+1} = \left(\frac{(p-1)I + N_k}{p}\right)^{-p} N_k . \tag{41}$$

Halley's method in Stable form Higham *et al*. (2005), Guo (2009) is defined below in the form:

$$X_0 = I, N_0 = A,$$

$$X_{k+1} = X_k\left((p+1)I + (p-1)N_k\right)^{-1}\left((p-1)I + (p+1)N_k\right), \tag{42}$$

$$N_{k+1} = N_k\left((p+1)I + (p-1)N_k\right)^{-1}\left((p-1)I + (p+1)N_k\right)^{-p}, \tag{43}$$

In all cases in equations (16)-(19), $N_k \to I$ as $k \to \infty$ when $A \to A^{\frac{1}{P}}$.

Those methods of Equations (36)-(41) were derived from Altman Hyper power method Guo and Higham (2006):

$$X_{k+1} = \frac{1}{p}\left[(p-1)X_k - \left(\left(I - X_k^p A\right)^q - I\right)X_k^{1-p}A^{-1}\right] \quad , \quad (X_0 \in R^{n\times n}) \tag{44}$$

This means that

$$X_{k+1} = \frac{1}{p} X_k \left[(p-1)X_k - \left(R^q - I\right)\left(I - R_k\right)^{-1} X_k \right]$$
$$= \frac{1}{p} X_k \left[(p-1) + \left(R_k^{q-1} + R_k^{q-2} + \ldots + R_k + I\right) \right] \quad (45)$$
$$= \frac{1}{p} X_k \left(pI + \left(\sum_{i=1}^{q-1} R_k^i \right) \right)$$

The matrix R_k^q appearing is a residual matrix of type $I - X_k^q A$. In the case of Newton p-th root matrix iteration, define that:

$$X \to \frac{1}{p} \left\{ (p-1)I - \sum_{i=1}^{q} \binom{q}{i} (-1)^i \left(X^p A\right)^{i-1} \right\} X \text{ , (the } X \to A^{\frac{1}{p}}\text{) .} \quad (46)$$

The Binomial term expansion:

$$\sum_{i=1}^{q} \binom{q}{i} (-1)^i \left(X^p A\right)^{i-1} = \sum_{i=1}^{q} \binom{q}{i} (-1)^i \left(X^p A^i\right)\left(X^p A\right)^{-1} = \left(\sum_{i=0}^{q} \binom{q}{i} (-1)^i \left(X^p A\right)^i - I\right)\left(X^p A\right)^{-1}$$

$$= \left(\left(I - X^p A\right)^q - I\right)\left(X^p A\right)^{-1}.$$

It follows that the map

$$F(X) = \frac{1}{p} \left\{ (p-1)X - \left(\left(I - X^p A\right)^q - I\right) X^{1-p} A^{-1} \right\} \quad (47)$$

so defined holds good.

To obtain the inverse Newton iteration, we proceed in a manner analogous to Equation (46) as follows:

$$F\left(A^{\frac{-1}{p}}\right) = \frac{1}{p} \left\{ (p+1)A^{\frac{-1}{p}} - \left(\left(I - \left(A^{\frac{-1}{p}} \right)^p A \right)^q - I \right) \left(A^{\frac{-1}{p}} \right)^{1-p} A^{-1} \right\} . \quad (48)$$

After minor rearrangements, Equation (48) would yield, Higham *et al*. (2005) that:

$$F\left(A^{\frac{-1}{p}}\right) = \frac{1}{p} \left\{ (p+1)X_k - X_k^{p+1} A \right\}, \quad (X_0 A = A X_0) \quad (49)$$

3. Distribution of eigenvalues for the matrix A

In what follows, we discuss the nature of resulting eigenvalues for the matrix A. By the real Schur decomposition, assuming $A \in R^{n\times n}$ is symmetric, there exists an orthogonal Q such that:

$$Q^T AQ = diag(\lambda_1, \lambda_2, ..., \lambda_n). \tag{50}$$

Since each eigenvalue of a symmetric matrix A is a stationary value of the map

$$x \to \frac{x^T Ax}{x^T x},\ x \neq 0\,. \tag{51}$$

Courant-Fisher minimax characterization shows that for each $k = 1,2,...,n$, there holds that

$$\lambda_k(A) = \max_{\dim(S)=k} \min_{0\neq y\in S} \frac{y^T Ay}{y^T y}. \tag{52}$$

The interlacing property reveals that the leading $r-by-r$ principal submatrix of an $n\times n$ symmetric matrix A implies that

$$\lambda_{r+1}(A_{r+1}) \le \lambda_r(A_r) \le \lambda_r(A_{r+1}) < ... < \lambda_r(A_{r+1}) \le \lambda_2(A_{r+1}) \le \lambda_1(A_r) \le \lambda_1(A_{r+1}),\ (r = 1,2,...,n-1) \tag{53}$$

Computation of Rayleigh Quotient iteration $\lambda = r(x) = \frac{x^T Ax}{x^T x}$ for the minimizing $\|(A-\lambda I)x\|_2$ in the least square sense is as follows:

Algorithm 3.1

Given a nonzero vector $x_0 \in R^n$, $\|x_0\|_2 = 1$,

for k= $0(1)\infty$

$\mu_k = r(x_k)$

Solve $(A-\mu_k I)w_{k+1} = x_k$ for w_{k+1}

$x_{k+1} = \frac{w_{k+1}}{\|w_{k+1}\|_2}$, end

In the case that matrix A is not symmetric, we can compute for the dominant eigenvalue by the Power method. We form the product $A^T A$ to obtain estimate $\eta \le \|A\|_2$. Then we have:

Algorithm 3.2

Given $q_0 \in R^n$, and $\|q_0\|_2 = 1$, ε - order of accuracy

for $k = 0:1:\infty$

$s_{k+1} = Aq_0$

$u_{k+1} = A^T s_{k+1}$

$$\eta_{k+1} = \frac{\|u_{k+1}\|_2}{\|s_{k+1}\|_2}$$

is $\|\eta_k - \eta_{k+1}\| \le \varepsilon$, quit , end.

The algorithm 3.2 indicated that the best estimate for the norm $\|A\|_2$ is given by

$$\|A\|_2 \ge \max\left[\frac{\|s_{k+1}\|_2}{\|u_k\|_2}, \frac{\|u_{k+1}\|_2}{\|s_{k+1}\|_2}, \left(\frac{\|u_{k+1}\|_2}{\|u_k\|_2}\right)^{\frac{1}{2}}\right]. \tag{54}$$

The distribution of eigenvalues for the operator in equation (1) defined by $f(X) = \frac{1}{2}\left(X_{k+1} + \frac{1}{X_{k+1}}\right)$ in second phase in the iteration process is distributed in the form Higham (2008):

$$1 \le \sigma_n^{(2)} \le \ldots \le \sigma_1^{(2)} = \frac{1}{2}\left(\sqrt{\frac{\sigma_1^{(1)}}{\sigma_n^{(1)}}} + \sqrt{\frac{\sigma_n^{(1)}}{\sigma_1^{(1)}}}\right) \le \frac{1}{2}\left(\sqrt{\sigma_1^{(1)}} + \frac{1}{\sqrt{\sigma_1^{(1)}}}\right) = f\left(\sqrt{\sigma_1^{(1)}}\right). \tag{55}$$

As earlier said, that the Givens Similarity Transformation preserves length by the Courant-Fischer minimax characterization, trace $A = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i$. Due to this fact, we state the Sylvester Law of Inertia.

Theorem 3.1; Golub and Van Loan (1983): If $A \in R^{n\times n}$ is symmetric and $X \in R^{n\times n}$ is nonsingular, then A and $X^T AX$ have the same inertia.

From the spectral decomposition of $A = Q^T AQ$, we compute the Lipschitz constant for the function of the matrix A thus, because $A = X^2$, it suffices to show that X is the square root of A. This implies writing that $Q^T XQ = (Q^T XQ)^2 = B^2$, therefore proved that B is Hermitian. Computed results would reveal that $B_k^2 = \lambda_k I$. By taking square root of both sides of the later, gives $B = \sqrt{\lambda_k}\, I_k$

Coupling these ideas together, we have $X = QA^{\frac{1}{2}}Q^T = Qdiag(\lambda_k I_k)Q^T = QA^{\frac{1}{2}}Q^T$. We use the opportunity to present the Low rank minimization of the matrix A .After computing the SVD of A ,then for $k = 0,1,...,.$, find $\min\left\|A_k - U_k AV_k^T\right\|$, also compute the root mean square error by the equation

$$\text{RMSRE}=\sqrt{\frac{1}{n}\sum\left\|A_k - U_k AV_k^T\right\|} \quad . \tag{56}$$

Thus the Low rank approximation tends to remove noise from the data and it has been found useful in machine learning, computer vision and information retrieval problems. Now we aim at computing

$$\frac{\left\|A_k - U_k \sum_k V^T\right\|}{n^2} = O\left(k^2\left(\frac{s_{k+1}^2(A)}{n^2} + \frac{\log(n)}{\sqrt{\delta n}}\right)\right). \tag{57}$$

The term $s_{k+1}(A)$ corresponds to the (k+1)-th singular values of A. In addition $\frac{\log^4(n)}{n} \le \delta \le n^{-\frac{8}{9}}$, where $k \le \log^6(n)$.

4. Numerical Examples.

Problem 1: Consider the matrix $A = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$.

We compute the 10^{th} root for the matrix A using Newton method, Halleys method and Inverse Newton method all implemented with MATLAB Windows 07. Using the above information, various results computed from these methods are displayed in Tables 1 and 2

Table 1: Computed results for the 10^{th} root of a matrix.

Iteration, k	NEWTON METHOD (15)	$K_N(A)$ (15)	INVERSE NEWTON METHOD (49)	$K_{IN}(A)$ (49)	HALLEY'S METHOD (39)	$K_H(A)$ (39)
1	$\begin{pmatrix} 0.9600 & 0.03000 & 0.0100 \\ 0.0200 & 0.9700 & 0.01007 \\ 0.0100 & 0.0100 & 0.9800 \end{pmatrix}$	2.532	$\begin{pmatrix} 1.0400 & -0.0300 & -0.0100 \\ -.0200 & 1.0300 & -0.0100 \\ -0.0100 & 1.0300 & 1.0200 \end{pmatrix}$	2.532	$\begin{pmatrix} 0.9459 & 0.0425 & 0.0116 \\ 0.0281 & 0.9603 & 0.0116 \\ 0.0123 & 0.0108 & 0.9769 \end{pmatrix}$	2.532
2	$\begin{pmatrix} 0.9447 & 0.0437 & 0.0116 \\ 0.0289 & 0.9595 & 0.0116 \\ 0.0124 & 0.0100 & 0.9767 \end{pmatrix}$	1.064	$\begin{pmatrix} 1.0591 & -0.0470 & -0.0120 \\ -0.0310 & 1.0430 & -0.0120 \\ -0.0130 & -0.0110 & 1.0241 \end{pmatrix}$	1.091	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.064
3	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0301 & 0.9582 & 0.0117 \\ 0.0126 & 0.0107 & 0.9766 \end{pmatrix}$	1.093	$\begin{pmatrix} 1.0626 & -0.0505 & -0.0121 \\ -0.0332 & 1.0453 & -0.0121 \\ -0.0134 & -0.0108 & 1.0242 \end{pmatrix}$	1.097	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.093
4	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.097	$\begin{pmatrix} 1.0627 & -0.0506 & -0.0121 \\ -0.0333 & 1.0454 & -0.0121 \\ -0.0134 & -0.0108 & 1.0242 \end{pmatrix}$	1.097	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.097

						1.097

Table 2: Results computed with Iannazzo (2006) in equation (40) for principal 10^{th} root of a positive definite matrix.

Iteration, k	**Iannazzo Method (40)** (X_k) **(MODIFIED NEWTON ITERATION METHOD)**	**Cond** (X_k) **FOR METHOD (40)**	**RESULTS FOR METHOD (18)** (M_k)	**Cond** (M_k) **FOR METHOD (18)**
1	$\begin{pmatrix} 0.9600 & 0.0300 & 0.0100 \\ 0.0200 & 0.9700 & 0.0100 \\ 0.0100 & 0.0100 & 0.9800 \end{pmatrix}$	1.0647	$\begin{pmatrix} 0.8371 & 0.1460 & 0.0169 \\ 0.0944 & 0.8860 & 0.0169 \\ 0.0256 & 0.0083 & 0.9662 \end{pmatrix}$	1.3533
2	$\begin{pmatrix} 0.9447 & 0.0437 & 0.0116 \\ 0.0289 & 0.9595 & 0.0116 \\ 0.0124 & 0.0108 & 0.9767 \end{pmatrix}$	1.0933	$\begin{pmatrix} 0.9778 & 0.0218 & 0.0004 \\ 0.0139 & 0.9857 & 0.0004 \\ 0.0023 & -0.0015 & 0.9992 \end{pmatrix}$	1.0382

3	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0301 & 0.9582 & 0.0117 \\ 0.0126 & 0.0107 & 0.9766 \end{pmatrix}$	1.0974	$\begin{pmatrix} 0.9996 & 0.0004 & 0.0000 \\ 0.0002 & 0.9998 & 0.0000 \\ 0.0000 & -0.0000 & 1.0000 \end{pmatrix}$	1.0006
4	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.0974	$\begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$	1.0000
5	$\begin{pmatrix} 0.9426 & 0.0457 & 0.0117 \\ 0.0302 & 0.9582 & 0.0117 \\ 0.0127 & 0.0107 & 0.9766 \end{pmatrix}$	1.0974	$\begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$	1.0000

Table 3: Showing Computed results for Euler-Chevbyshev Method 1.1a (Square root case) and its Inverse.

Iteration K	**Euler-Chevbyshev Method 1.1a** (X_k)	**cond** (X_k)	**Inverse-Euler-Chevbyshev Method** (Y_k)	**cond** (Y_k)
1	$\begin{pmatrix} 0.7483 & 0.1968 & 0.0548 \\ 0.1302 & 0.8149 & 0.0548 \\ 0.0578 & 0.0519 & 0.8902 \end{pmatrix}$	1.6291	$\begin{pmatrix} 1.3547 & -0.2903 & -0..0645 \\ -0.1906 & 1.2550 & -0.0645 \\ -0.0733 & -0.0557 & 1.1289 \end{pmatrix}$	1.5552
2	$\begin{pmatrix} 0.7572 & 0.1883 & 0.0544 \\ 0.1248 & 0.8207 & 0.0544 \\ 0.0567 & 0.0522 & 0.8910 \end{pmatrix}$	1.5918	$\begin{pmatrix} 1.3767 & -0.3117 & -0.0651 \\ -0.2043 & 1.2693 & -0.0651 \\ -0.0757 & -0.0545 & 1.1300 \end{pmatrix}$	1.5917
3	$\begin{pmatrix} 0.7572 & 0.1883 & 0.0544 \\ 0.1248 & 0.8207 & 0.0544 \\ 0.0567 & 0.0522 & 0.8910 \end{pmatrix}$	1.5918	$\begin{pmatrix} 1.3767 & -0.3117 & -0.0651 \\ -0.2043 & 1.2693 & -0.0651 \\ -0.0757 & -0.0545 & 1.1300 \end{pmatrix}$	1.5917

In the implementation of Halley's method it is strongly recommended to use numerical methods for inverting the resulting matrix instead of direct methods due ill-conditioning of the elements of the matrix. In this work we use the Lu decomposition and economy size QR decomposition to invert the matrices occurring in the iteration phases.

Define that $A = LDM = LU$, (58)

where L and M are unit triangular matrices, D is a diagonal matrix. That is, the structure of L for the (k+1)-th principal determinant is given by $A^{(0)} = \left(L_1^{-1}.L_2^{-1}...L_k^{-1}\right)A^{(k)}$

$$L = \begin{pmatrix} 1 & & & 0 \\ l_{21} & . & & \\ . & & . & \\ . & & & \\ . & & & . \\ l_{k1,1} & ... & l_{k+1,k} & 1 \end{pmatrix}, \tag{59}$$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} ... & & a_{1k}^{(0)} \\ & a_{22}^{(1)} ... & & a_{2n}^{(1)} \\ & & .. . & \\ & & ... & \\ & & .... & \\ 0 & & & a_{n,n}^{(k)} \end{pmatrix}. \tag{60}$$

Thus each $m_i = 1.a_{11}^{(0)}a_{22}^{(1)}...a_{k+1,k+1}^{(k)} \neq 0$ and $a_{k+1,k+1}^{(k)} \neq 0$. The implication is that since A is symmetric and nonsingular, A has a unique LDM^T factorization where D is a diagonal matrix and $L = M^T$. In the long run in the iteration we have that

$$A = L_1 L_2 ... L_k D M_k M_{k-1} ... M_1, \tag{61}$$

where $k = \log n$. Hence, it follows that $A^{-1} = M^{-1}D^{-1}L^{-1}$. Especially for the Hermitian positive case, we set $\hat{L} = L\sqrt{D}$ and thus, $A = \hat{L}\hat{L}^H$. We give algorithmic structure for the Cholesky the Cholesky Factorization.

Algorithm

Given a symmetric matrix A

For $k = 1,2,...,n$,

$$a_{kk} = \left(a_{kk} - \sum_{j=1}^{k-1} a_{kj}^2 \right)^{\frac{1}{2}},$$

for $i = k+1,...,n$,

$$a_{ik} = \left(a_{ik} - \sum_{j=1}^{k-1} a_{ij} a_{kj} \right) / a_{kk}$$

Endfor

Endfor

End

This has polynomial time of $\frac{n^3}{6}$.

Another method for inverting matrices in the Newton and Halley's methods is the use of Singular Value Decomposition (SVD). This is made possible since the SVD is proven to be backward stable, Uwamusi (2016).

Firstly we obtain a rotation matrix defined in the form

The matrix Q is orthogonal and is defined in the form:

$$
Q(p,q,\theta) = \begin{pmatrix}
1 & \cdots & & & & & 0 \\
 & \ddots & & & & & \\
 & & 1 & & & & \\
 & & & \cos\theta & & \sin\theta & \\
 & & & \vdots & \ddots & & \\
 & & & & & 1 & \\
 & & & -\sin\theta & & \cos\theta & \\
 & & & \vdots & & & \ddots \\
0 & & & & & & 1
\end{pmatrix}
\begin{matrix}
 \\
 \\
 \\
 \\
 \\
\leftarrow p \\
 \\
 \\
\leftarrow q
\end{matrix}
\qquad (62)
$$

With the matrix $Q(p,q,\theta)$ we reduce the matrix A to an equivalent QR-Cholesky Decomposition using the Givens orthogonal matrix plane rotation matrices Uwamusi (2002). The $\cos\theta$ and $\sin\theta$ are computed in the form:

$\cos\theta = \dfrac{p_i}{\sqrt{p_i^2 + q_i^2}}$, $\sin\theta = \dfrac{q_i}{\sqrt{p_i^2 + q_i^2}}$. Where $x \in R^n$ and $y = J(p,q,\theta)x$ and after multiplication by $J(p,q,\theta)$ in the (p,q) coordinate plane gives that

$$y_p = cx_i + sx_q \,; y_q = -sx_i + cx_k \,; y_j = x_j \qquad (63)$$

Continuing in this manner, we eventually upper triangulate the matrix A in the form

$R = \begin{pmatrix} \times \times ... & & \times \\ \times & \times .... & \times \\ & \times \times & ... \times \\ & . . . & \\ O & & ..... \\ & & .... \\ & & \times \end{pmatrix}$, where $\times,\times,...\times$ are element of the Upper triangulated matrix. The elementary matrices $Q_1, Q_2, ...Q_n$ are the constituents Givens rotation matrices. Assembling these results, we write that $\hat{A} = Q_m Q_{m-1} ... Q_1 A R_1 R_2 ... R_m$ .As is standard, the QR algorithm is numerically stable, so converges for positive definite matrices. On the other hand, the matrix needed to be inverted could be split into Singular Value Decomposition (SVD). For instance, if $A = U \sum V^T$, then we compute $A^{-1} = V \sum^{-1} U^T$. The components of decomposed A are that U and V are left and right hand vectors corresponding to the singular values $\sum = diag(\sigma_1, \sigma_2, .., .\sigma_n)$, where $\sigma_1 > \sigma_2 > ... > \sigma_n$ are ordered in increasing magnitudes.

5. Conclusion

The paper presented some methods based on Newton and Halley's methods and their variants for computing the principal p-th root of a matrix whose eigenvalues are not on the R^- axis. In particular, it is showed that Halley's method converges faster than Newton method as was demonstrated with numerical example whose computed results are displayed in Table 1 where p is taken to be 10. In Table 1, the Halley's method converged to the required solution in the second step of iteration, whereas, the Newton method converged at the third iteration. We also gave their condition numbers corresponding to each stage in their iteration cycle for the obtained results. In Table 1 also, the results for the inverse 10^{th} root for matrix A obtained from Inverse Newton iteration for the case p=10 was reported. We used the LU–Cholesky Factorization and Givens–QR decomposition for the inversion of resulting matrices appearing in the computation for the Halley's iteration method. Especially, the QR decomposition is numerically backward stable. Furthermore, the Givens-QR algorithm when run to completion also provides the eigenvalues of the matrix free of charge without additional task of cost in computation time.

In addition, we also reported in Table 2, computed results for the modified Newton method due to Iannazzo (2006) as represented. It revalidated that method is actually the Stable form of Newton iteration for computing principal p-th root of a positive definite matrix. Furthermore, while $X_k \to A^{\frac{1}{10}}$, the corresponding $M_k \to I$ an indication that there is a good reasonable approximate results to the already known values which can be obtained by any other methods. As a way of comparison, we also give the results obtained by Guo and Higham (2006) for the cases p=12 and 52 respectively where they used method of Polar decomposition in their approach as shown below:

$$p = \frac{1}{12}, \quad X = \begin{bmatrix} 0.9518 & 0.0384 & 0.0098 \\ 0.0253 & 0.9649 & 0.0098 \\ 0.0106 & 0.0089 & 0.9805 \end{bmatrix}, \quad P = \frac{1}{52}, \; X = \begin{bmatrix} 0.9886 & 0.0092 & 0.0023 \\ 0.0060 & 0.9917 & 0.0023 \\ 0.0025 & 0.0021 & 0.9954 \end{bmatrix}.$$

It can be shown that with either methods of Newton or Halley's for the case $p \geq 10$, considering the scalar test matrix as given by the example above, the results with our methods are in close agreement with those of Guo and Higham (2006). In our methods, we used both LU decomposition and Givens QR-Cholesky type Factorization. In either case, the results we obtained are accurate within the precision, epsilon.

We also reported that we used Euler-Chevbyshev method due to Lakic and Petkovic to compute the square root and its corresponding inverse matrix square root with results displayed in Table 3. Convergence for this method was achieved just in the second iteration to the desired root. All computations were carried out with the aid of MATLAB Windows 7 version.

We pointed out that Newton method may become highly unstable sometimes. To remedy this defect, we recommend that for convergence all eigenvalues of matrix A must be bounded by the quantity (Innazzo, 2006):

$$\frac{1}{p}\left| p - \sum_{k=1}^{p} \left(\frac{\lambda_i}{\lambda_j} \right)^{\frac{k}{p}} \right| \leq 1, \;\; (i = 1,2,..., n; \; j = 1,2,..., n) \tag{64}$$

References

Altman, M. (1960): An optimum cubically convergent iterative method of inverting a linear bounded operator in Hilbert space. *Pacific Journal Math*ematics. **10**, 1107-1113.

Bini, D. and Pan, V. Y. (1994): *Polynomial and matrix computation*: *Fundamental Algorithms*. Birkhauser, Boston.

Bjorck, A. (2009): *Numerical methods in Scientific Computing*. **2**, SIAM, Philadelphia.

Breass, D. and Hadeler, K. (1973): Simultaneous inclusion of the zeros of a polynomial. *Numerische Mathematik.* **21**, 161-165.

Cheng, S., Higham, N., Kenney, C. and Laub, A. (2001): Approximating the logarithm of a matrix to specified accuracy. *SIAM Journal of Mathematical Analysis and Applications*. **22** (2), 1112-1125.

Guo, C. (2010): On Newton's method and Halley's method for the principal pth root of a matrix. *Linear Algebra and its Applications*. **432** (8), 1905-1922.

Guo, C. and Higham, N. (2006): A Schur-Newton method for the matrix p-th root and its inverse. *SIAM Journal on Matrix Analysis and Applications.* **28** (3), 788-804.

Golub, G. and Van-Loan, F. (1983): *Matrix Computations*. The Johns Hopkins University Press, Baltimore Maryland.

Hansen, E. and Patrick, M. (1977): A family of root finding methods. *Numerische Mathematik.* **27**, 257-269.

Higham, N., Mackey, D., Mackey, N. and Tisseur, F. (2004): Computing the polar decomposition and the matrix sign decomposition in matrix groups. *SIAM Journal on Matrix Analysis and Applications.* **25**, 1178-192.

Higham, N., Mackey, D., Mackey, N. and Tisseur, F. (2005): Functions preserving Matrix groups and iterations for the matrix square root. *SIAM Journal on Matrix Analysis and Applications.* **26** (3), 849-877.

Higham, N. (2008): *Functions of matrices: Theory and Computation*. SIAM, Philadelphia.

Innazzo, B. (2006): On the Newton method for the Matrix p-th root. *SIAM Journal on Matrix Analysis and Applications.* **28**, 503-523.

Lakic, S. and Petkovic, M. (1998): On the matrix square root. Zeitschrift für angewandte Mathematik und Mechanik. **78** (3), 173 – 182.

Ostrowski, A. (1966): *Solutions of Equations and Systems of Equations*. Chapter 15, Academic Press, London, UK.

Petkovic, M. and Trikovic, S. (1995): On Zero finding method of the fourth order. *Journal of* Computational and Applied Mathematics. **64**, 291-294.

Sharp, W. and Allen, E. (2006): Stochastic Neutron transport equation for Rod and plane geometries. *Annals of Nuclear Energy*. **27**, 99-116.

Uwamusi, S. and Otunta, F. (2002): Computation of Eigenvalues of Hermittian matrix via Givens plane rotation. *Nigerian Journal of Applied Sciences*. **20**, 101-106.

Uwamusi, S. (2005a): A Class of algorithms for Zeros of Polynomial. *Pakistan Journal of Scientific and Industrial Research*. **48** (3), 149-153.

Uwamusi, S. (2005b): On methods derived from Hansen-Patrick formula for refining Zeros of Polynomial equation. *Pakistan Journal of Scientific and Industrial research*. **48** (5), 297-302.

Uwamusi, S. (2016): Jacobi Similarity Transformation for SVD and Tikhonov regularization

for Least squares problem: The theoretical foundation. *IOSR Journal of Mathematics*. **12** (5) Ver. VII, 76-85.

Yan, Y. (2005): Galerkin finite element methods for Stochastic parabolic partial differential equations. *SIAM Journal on Numerical Analysis*. **43** (4), 1363-1384.

Wasilkowski, W. (1980): Can any stationary iteration using linear information be globally convergent? *Journal of ACM*. **27**, 263-269.