Applied Mathematics E-Notes, 23(2023), 229-236 © ISSN 1607-2510
Available free at mirror sites of http://www.math.nthu.edu.tw/∼amen/

On A High Order Algorithm Converging To The mth Root Of A Positive Matrix And Application For Computing The Matrix Logarithm*

Nawal Helal Al-harbi†, Mustapha Raïssouli‡, Daoud Suleiman Mashat§

Received 18 April 2022

Abstract

In this paper, a high order algorithm converging to the mth root of a positive square matrix is investigated. An application for the computation of the logarithm of an invertible positive matrix is provided as well. Numerical simulation illustrating the theoretical study and showing the interest of our approach is also discussed.

1 Introduction

Numerical resolution of equations/systems arises in various context and contributes as a tool for solving many scientific problems. Iterative algorithms for approximating the roots of a given equation are usually used as useful way in theoretical point of view as well as in practical purposes. It has been proved, throughout a lot of studies, that the so-called Newton's algorithm is one of the most interesting methods for finding the roots of a nonlinear equation $f(x) = 0$. Two facts were discovered for Newton's method. Positively, such method converges much more rapidly than a simple iterative method. Negatively, the starting point should be chosen close to the unknown zero of $f(x) = 0$. Practically, this imposes a serious difficulty even for real equations with one unknown, and such difficulty increases when we have to solve a system with multiple unknowns. Let us discuss in detail our situation which turns into finding the principal mth roots of a matrix $A \in \mathbb{R}^{p\times p}$ [3, 4, 5, 6] whose eigenvalues lie in the sector

$$S_m = \Big\{ z \in \mathbb{C},\ z \neq 0,\ \ -\frac{\pi}{m} < \arg z < \frac{\pi}{m} \Big\}.$$

More precisely, let $m \geq 2$ be an integer and consider the nonlinear matrix equation

$$\text{Find a matrix } X \text{ such that } \quad F(X) := X^m - A = 0. \tag{1}$$

As in the general case, the Newton's method associated to (1) is described by the following iterative scheme

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

with given initial guess X_0, provided that $F'(X_k)$ is invertible at each iteration X_k. Here, F' refers to the Fréchet derivative of F. If we write the Taylor series for the matrix function $F(X)$ given by (1) we obtain

$$F(X+H) = (X+H)^m - A = (X^m - A) + \sum_{i=0}^{m-1} X^{m-1-i} H X^i + o\left(H^2\right), \tag{3}$$

*Mathematics Subject Classifications: 15A24, 65F45.

†Department of Mathematics, Faculty of Science, King Abdulaziz University, P.O. Box 80203, Jeddah 21589, Saudi Arabia, and Department of Mathematics, College of Sciences, King Khalid University, Abha 61413, Saudi Arabia
‡Department of Mathematics, Faculty of Science, University of Moulay Ismail, Meknes, Morocco
§Department of Mathematics, Faculty of Science, King Abdulaziz University, P.O. Box 80203, Jeddah 21589, Saudi Arabia

where $o(H^2)$ is a neglected expression in the sense that

$$\lim_{H\to 0}\frac{\|o(H^2)\|}{\|H\|}=0.$$

We then observe that $F'(X)$ is the linear mapping given by

$$F'(X)H=\sum_{i=0}^{m-1}X^{m-1-i}HX^i.$$

After some simple algebraic manipulations, the classical Newton's method, starting at X_0, can be written as

$$\begin{cases}\sum_{i=0}^{m-1}X_k^{m-1-i}H_kX_k^i=A-X_k^m,\\ H_k:=X_{k+1}-X_k.\end{cases}$$

Otherwise, in [1] the authors presented a modified algorithm of (2) given by

$$X_{k+1}=X_k-F'\left(X_k;I\right)^{-1}F\left(X_k\right),\quad k=0,1,\ldots, \tag{4}$$

where $F'\left(X_k;I\right)$ stands for the directional derivative of F at X_k in the direction I, identity matrix, namely

$$F'\left(X_k;I\right):=\lim_{t\downarrow 0}\frac{F(X_k+tI)-F(X_k)}{t}. \tag{5}$$

If F is as in (1), it is easy to check that (3) when applied to (5) yields $F'\left(X_k;I\right)=mX_k^{m-1}$. Substituting this in (4) we get

$$X_{k+1}=\frac{(m-1)X_k+X_k^{1-m}A}{m},\quad k\geq 0, \tag{6}$$

with X_0 conveniently chosen. Observe that (6) is a matrix version of the Newton's algorithm for computing the mth root of a real number $a\geq 0$, namely

$$x_{k+1}=\frac{(m-1)x_k+ax_k^{1-m}}{m},\quad k\geq 0, \tag{7}$$

with $x_0>0$ given. The algorithm (7) converges to $a^{1/m}$ for each given $x_0>0$, see Proposition 1 below. It is worth mentioning that this latter result does not need $x_0>0$ to be close to the exact root $a^{1/m}$. Furthermore, the convergence of (7) to $a^{1/m}$ is quadratic in the sense that, there is $\alpha:=\alpha_{m,a}>0$ such that for all $k\geq 0$ we have

$$\left|x_{k+1}-a^{1/m}\right|\leq\alpha\left|x_k-a^{1/m}\right|^2. \tag{8}$$

About (6), if X_0 commutes with A then the study of the convergence of the matrix sequence (X_k) is reduced to that of (7), and so (X_k) converges quadratically to $A^{1/m}$, namely

$$\left\|X_{k+1}-A^{1/m}\right\|\leq\beta\left\|X_k-A^{1/m}\right\|^2 \tag{9}$$

for some $\beta>0$ depending only on m and A. Here, and in what follows, $\|.\|$ refers to any norm among the equivalent norms of the Banach algebra of $n\times n$ matrices. For instance, we can choose the standard matrix norm defined by

$$\|A\|:=\sup_{x\neq 0}\frac{\|Ax\|}{\|x\|}=\sup_{\|x\|=1}\|Ax\|, \tag{10}$$

where $\|x\|$ is the Euclidian norm of $x\in\mathbb{R}^n$. The matrix norm defined by (10) satisfies

$$\|AB\|\leq\|A\|\|B\| \tag{11}$$

for any square matrices A and B.

The fundamental purpose of this paper is to accelerate the convergence of (X_k) to $A^{1/m}$. We then obtain some algorithms converging to $A^{1/m}$ with rates of convergence equal to $2^2,2^3,\ldots$. Afterwards, we give an application to the computation of the logarithm of an invertible positive matrix. In order to illustrate our theoretical approach, a numerical simulation is discussed.

2 High Order Algorithm Approximating $A^{1/m}$

We preserve the same notations as in the previous section. Before stating our main result, we will establish the following result already mentioned in the previous section.

Proposition 1 *Let $a > 0$ be a real number and $m \geq 2$ be an integer. For any $x_0 > 0$, the sequence (x_k) defined by (7) converges quadratically to $a^{1/m}$.*

Proof. From (7), it is clear that if $x_0 > 0$ then $x_k > 0$ for all $k \geq 0$. Since the real function $x \longmapsto \log x$ is concave on $(0, \infty)$ then (7) yields

$$\begin{aligned} \log x_{k+1} &\geq \frac{m-1}{m} \log x_k + \frac{1}{m} \log(a x_k^{1-m}) \\ &= \frac{m-1}{m} \log x_k + \frac{1}{m} \log a + \frac{1-m}{m} \log x_k = \frac{1}{m} \log a = \log a^{1/m}. \end{aligned}$$

Thus, $x_{k+1} \geq a^{1/m}$ for any $k \geq 0$. Using again (7), we get

$$0 \leq x_{k+1} - a^{1/m} = \frac{m-1}{m}\big(x_k - a^{1/m}\big) + \frac{1}{m}\big(a x_k^{1-m} - a^{1/m}\big). \tag{12}$$

From $x_k \geq a^{1/m}$, for all $k \geq 1$, we easily deduce that $a x_k^{1-m} - a^{1/m} \leq 0$. Therefore, (12) leads to

$$0 \leq x_{k+1} - a^{1/m} \leq \frac{m-1}{m}\big(x_k - a^{1/m}\big),$$

which by a simple mathematical induction yields

$$0 \leq x_{k+1} - a^{1/m} \leq \Big(\frac{m-1}{m}\Big)^k \big(x_1 - a^{1/m}\big).$$

This, with the fact that $0 < \frac{m-1}{m} < 1$, implies that (x_k) converges to $a^{1/m}$. Since (7) is a Newton type convergent algorithm the convergence of (x_k) to $a^{1/m}$ is quadratic. ■

We need to recall the following result, see [1].

Lemma 1 *Let $\phi_1, \phi_2, \ldots, \phi_s$ be functions corresponding to iterative methods of orders $r_1, r_2, \ldots, r_s$, respectively. Then the composition of iterative functions*

$$\phi(x) = \phi_1 \circ \phi_2 \circ \ldots \circ \phi_s(x) := \phi_1\Big(\phi_2\big(\cdots(\phi_s(x))\cdots\big)\Big)$$

defines an iterative method of order $r_1 r_2 \cdots r_s$.

Let us explain how to use Lemma 1. To fix our idea, let us take $s = 2$. Lemma 1 asserts that if the two iterative algorithms $x_{k+1} = \phi_1(x_k)$ and $x_{k+1} = \phi_2(x_k)$ are convergent of orders r_1 and r_2, respectively, then the following algorithm

$$x_{k+1} = \phi_1\big(\phi_2(x_k)\big) \tag{13}$$

is also convergent with order $r_1 r_2$. Setting $y_k = \phi_2(x_k)$, (13) is then equivalent to the coupled algorithm given by

$$\begin{cases} y_k = \phi_2(x_k), \\ x_{k+1} = \phi_1(y_k). \end{cases} \tag{14}$$

Applying (14) with the Newton's algorithm, that is,

$$\phi_1(x) = \phi_2(x) = x - \frac{f(x)}{f'(x)},$$

we then obtain the following coupled algorithm

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \dfrac{f(y_k)}{f'(y_k)}. \end{cases} \tag{15}$$

Lemma 1 tell us that, if the Newton's algorithm corresponding to the equation $f(x) = 0$ is convergent (with order 2) then (15) is also a convergent algorithm with order $2 \times 2 = 4$.

Summarizing, if we take $f(x) = x^m - a$ in (15) we are in the position to consider the following coupled algorithm

$$\begin{cases} y_k = \dfrac{(m-1)x_k + ax_k^{1-m}}{m}, \\ x_{k+1} = \dfrac{(m-1)y_k + ay_k^{1-m}}{m}, \end{cases} \tag{16}$$

which is convergent to $a^{1/m}$ with order four, provided that the initial guess x_0 satisfies $x_0 > 0$.

Inspired by (16), with the help of (6), we now consider the following coupled algorithm involving matrix arguments

$$\begin{cases} Y_k = \dfrac{(m-1)X_k + X_k^{1-m}A}{m}, \\ X_{k+1} = \dfrac{(m-1)Y_k + Y_k^{1-m}A}{m}. \end{cases} \tag{17}$$

We mention that, whenever X_0 is conveniently chosen the matrix sequence (X_k) is well defined. We are in the position to state the following main result.

Theorem 1 *Let X_0 be an invertible positive matrix such that $AX_0 = X_0A$. Then the sequence (X_k) defined through (17) satisfies the following properties:*

(i) X_k is invertible positive, for each $k \geq 0$.

(ii) (X_k) converges to $A^{1/m}$, for any $m \geq 2$.

(iii) The speed of convergence of (X_k) to $A^{1/m}$ is equal to four. That is, the following estimation

$$\forall k \geq 0 \quad \left\| X_{k+1} - A^{1/m} \right\| \leq \beta_m \left\| X_k - A^{1/m} \right\|^4 \tag{18}$$

holds for some constant $\beta_m > 0$ depending only on A and m.

Proof. Since $AX_0 = X_0A$ simple mathematical induction leads to show that $AX_k = X_kA$ for all $k \geq 0$. Then, (17) generates iterative matrices which are all analytical functions with respect to the given matrix A. The convergence of (X_k) is then easily examined by transforming the iterates to diagonal forms and so considering the convergence of the scalar version (16) by standard way. The details are straightforward and therefore omitted here. ■

We end this section by stating the following remarks which may be of interest for the reader.

Remark 1 *Taking $s = 3, 4, ...$ in Lemma 1, and by the same previous way for $s = 2$, we can construct matrix algorithms converging to $A^{1/m}$ with orders $2^3, 2^4, ...$, respectively. We left to the reader the task for formulating the matrix algorithm that converges to $A^{1/m}$ with order equal to $2^3 = 8$. We notice that, the previous algorithm of order 2^2 is more than enough for obtaining quite good approximations for $A^{1/m}$ even from the first iterations. This latter claim will be discussed and justified by numerical examples in Section 4.*

Remark 2 *The previous study, and its related algorithms, are still valid for computing $A^{a/b}$ where a, b are integer numbers with $b \geq 2$. Indeed, instead of (1) we take $F(X) = X^b - A^a = 0$ or we simply write $A^{a/b} = M^{1/b}$ with $M = A^a$. By the same previous procedure, we obtain the following algorithm*

$$Y_k = \frac{(b-1)X_k + X_k^{1-b}A^a}{b}, \quad X_{k+1} = \frac{(b-1)Y_k + Y_k^{1-b}A^a}{b},$$

which for $a = 1$ and $b = m$ coincides with (17).

3 Application for Computing $\log A$

The computation of logarithm for a positive matrix arises in various numerical contexts. Before stating our claimed application about the approximation of $\log A$, we need the following lemma.

Lemma 2 *Let A be an invertible positive matrix. Then we have*

$$\log A = \lim_{n\uparrow\infty} 2^{n-1}\Big(A^{\frac{1}{2^n}} - A^{-\frac{1}{2^n}}\Big). \tag{19}$$

Proof. We present here two ways for proving this result.

Method 1: We first show (19) for real arguments. Let $a > 0$ be a real number. Setting $t = 1/2^n$ and using the standard l'Hopital rule, we get

$$\lim_{n\uparrow\infty} 2^{n-1}\Big(a^{1/2^n} - a^{-1/2^n}\Big) = \lim_{t\downarrow 0}\frac{a^t - a^{-t}}{2t} = \lim_{t\downarrow 0}\frac{a^t\log a - (-1)a^{-t}\log a}{2} = \log a.$$

To prove (19) for matrix arguments we then use the techniques of Functional Calculus (Cauchy's integral formula) or we simply use the process of matrix diagonalization.

Method 2: We can directly show (19) for matrix arguments. Indeed, the Taylor's series when applied for matrix exponential gives

$$A^{1/2^n} := \exp\Big(\frac{1}{2^n}\log A\Big) = \sum_{k=0}^{\infty}\frac{(\log A)^k}{2^{nk}k!},$$

$$A^{-1/2^n} := \exp\Big(-\frac{1}{2^n}\log A\Big) = \sum_{k=0}^{\infty}\frac{(\log A)^k}{(-1)^k 2^{nk}k!}.$$

It follows that

$$\begin{aligned} A^{\frac{1}{2^n}} - A^{-\frac{1}{2^n}} &= \sum_{k=0}^{\infty}\frac{(\log A)^k}{k!2^{nk}}\Big(1 - \frac{1}{(-1)^k}\Big) = \sum_{k=0}^{\infty}\frac{(\log A)^{2k+1}}{(2k+1)!2^{n(2k+1)-1}} \\ &= \frac{\log A}{2^{n-1}} + \sum_{k=1}^{\infty}\frac{(\log A)^{2k+1}}{(2k+1)!2^{n(2k+1)-1}}. \end{aligned} \tag{20}$$

Let $\|.\|$ be the matrix norm defined by (10). From (20), and using (11), we deduce that

$$\begin{aligned} \Big\|2^{n-1}\Big(A^{\frac{1}{2^n}} - A^{-\frac{1}{2^n}}\Big) - \log A\Big\| &= \Big\|\sum_{k=1}^{\infty}\frac{(\log A)^{2k+1}}{(2k+1)!2^{2nk}}\Big\| \\ &\leq \sum_{k=1}^{\infty}\frac{\|\log A\|^{2k+1}}{(2k+1)!2^{2nk}} \leq \frac{1}{2^{2n}}\sum_{k=1}^{\infty}\frac{\|\log A\|^{2k+1}}{(2k+1)!} < \frac{1}{2^{2n}}\exp\|\log A\|. \end{aligned}$$

The right expression of this latter inequality tends to 0 when $n \uparrow \infty$, so the desired result is proved. ■

According to (17), for fixed $n \geq 0$, we consider the following algorithm

$$\begin{cases} U_{k,n} = \left(1 - 2^{-n}\right) V_{k,n} + 2^{-n} V_{k,n}^{1-2^n} A, \\ V_{k+1,n} = \left(1 - 2^{-n}\right) U_{k,n} + 2^{-n} U_{k,n}^{1-2^n} A, \end{cases} \tag{21}$$

where $V_{0,n}$ is given for any fixed $n \geq 0$. Following Theorem 1, $V_{k,n}$ approximates $A^{1/2^n}$ when $k \uparrow \infty$. By a simple argument of continuity, $V_{k,n}^{-1}$ approximates $A^{-1/2^n}$ when $k \uparrow \infty$. Thanks to Lemma 2, we have established the following result.

Proposition 2 *Let A be an invertible positive matrix. Then we have*

$$\log A = \lim_{k,n \uparrow \infty} 2^{n-1} \left(V_{k,n} - V_{k,n}^{-1} \right), \tag{22}$$

where $(V_{k,n})$ is defined by (21) when $V_{0,n}$ satisfies the condition $V_{0,n} A = A V_{0,n}$.

4 Concluding Remarks and Numerical Examples

In this paper we have constructed and studied an iterative matrix algorithm approximating $A^{1/m}$ with a rate of convergence equal to 2^2. As already pointed out in Remark 1, we can pursue the same procedure by constructing algorithms of orders $2^3, 2^4, \ldots$ also converging to $A^{1/m}$. These algorithms are all of Newton type whose convergence depends generally on the initial guess that should be chosen close to the exact solution which is in fact unknown. This is most inconvenient aspect of the Newton method, even for scalar equations and the difficulty increases when we have to solve matrix equations. However, in our previous study, the initial data X_0 needs only to satisfy the simple condition $AX_0 = X_0 A$, for example $X_0 = I$ or $X_0 = A$, which can be easily manipulable in the numerical situation when A is a given matrix. This latter point will be explained by the following numerical examples.

We notice that the numerical simulation here was done by using MATLAB 2021a with 3500 digits floating arithmetic. In the Banach algebra of square matrices, the computations were stopped when the estimation $\|X_{k+1} - X_k\| < \epsilon$ is satisfied, for some $\epsilon > 0$ small enough.

Example 1 *Let us consider the matrix*

$$A = \begin{pmatrix} 13 & 4 & -5 \\ 4 & 17 & 2 \\ -5 & 2 & 19 \end{pmatrix},$$

whose the 2-norm condition number is $Cond_2(A) := \|A\|_2 \|A^{-1}\|_2 \approx 2.744$. Taking $m = 2, 3, 4$, and $X_0 = I$, in algorithms (6) and (17) we obtain the following estimations for computing $A^{1/m}$. See Table 1.

Example 2 *Let us consider the so-called Wilson matrix [2]*

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}.$$

About this specific matrix we have $Cond_2(A) := \|A\|_2 \|A^{-1}\|_2 \approx 2984$. Here we take $m = 5, 6, 7$ and $X_0 = I$, in (6) and (17). We obtain the following estimations. See Table 2.

Example 3 *Let us find an approximation of $\log A$, when A is the Wilson matrix, by using Proposition 2. If we set*

$$L_{k,n} = 2^{n-1} \left(V_{k,n} - V_{k,n}^{-1} \right),$$

Table 1: Estimations for $A^{1/m}$.

	m=2		m=3		m=4	
	$\|X_{k+1}-X_k\|_2$		$\|X_{k+1}-X_k\|_2$		$\|X_{k+1}-X_k\|_2$	
k	Alg. (6)	Alg. (17)	Alg. (6)	Alg. (17)	Alg. (6)	Alg. (17)
0	10.424	5.6682	6.9493	4.4148	5.212	3.6818
1	4.7557	1.9851	2.5345	2.3534	1.5302	1.8878
2	1.6959	8.9384e-3	1.5565	0.26564	1.1172	0.6115
3	0.2892	7.7835e-12	0.79685	1.7779e-04	7.7054e-1	2.0525e-2
4	8.9229e-3	4.4926e-48	0.24341	4.5721e-17	4.4808e-1	5.7425e-8
5	8.5301e-6	4.9862e-193	2.2235e-2	2.0001e-67	1.6342e-1	3.6319e-30
6	7.7835e-12	7.5661e-773	1.7778e-4	7.3252e-269	2.0237e-2	5.8107e-119
7	6.4806e-24	4.0113e-3092	1.1306e-8	1.3178e-1074	2.8767e-4	3.8074e-474

Table 2: Computations for $A^{1/m}$.

	m=5		m=6		m=7	
	$\|X_{k+1}-X_k\|_2$		$\|X_{k+1}-X_k\|_2$		$\|X_{k+1}-X_k\|_2$	
k	Alg. (6)	Alg. (17)	Alg. (6)	Alg. (17)	Alg. (6)	Alg. (17)
0	5.8577	4.4889	4.8814	3.9019	4.1841	3.4437
1	1.3688	1.9545	0.97952	1.4919	0.74036	1.1771
2	1.0911	1.1516	0.8152	1.0061	0.63426	0.85471
3	0.86337	0.39254	0.67669	0.54806	0.5428	0.56907
4	0.66807	1.2166e-2	0.55739	9.016e-2	0.46311	0.2086
5	0.48354	2.2092e-8	0.44872	1.4817e-4	0.3916	64704e-3
6	0.28867	2.4619e-31	0.33777	1.3679e-15	0.32259	1.0741e-8
7	0.10387	3.7968e-123	0.21029	9.9415e-60	0.24647	8.3297e-32
8	1.2018e-2	2.1477e-490	8.0009e-2	2.7732e-236	0.15282	3.0133e-124
9	1.4781e-4	2.1989e-1959	1.0151e-2	1.6793e-942	5.5774e-2	5.1605e-494

Table 3: Approximations for $\log A$.

	n=2	n=3	n=4	n=5	n=6
k	$\|L_{k+1,n}-L_{k,n}\|_2$	$\|L_{k+1,n}-L_{k,n}\|_2$	$\|L_{k+1,n}-L_{k,n}\|_2$	$\|L_{k+1,n}-L_{k,n}\|_2$	$\|L_{k+1,n}-L_{k,n}\|_2$
10	7.0127e-639	3.4187e-339	2.3253e-12	0.56649	1.9215
11	4.789e-2555	8.7774e-1356	2.8144e-48	0.0082624	1.3993
12	8.9121e-3468	4.4601e-3463	6.0393e-192	5.18e-10	0.27858
13	1.1928e-3463	2.0803e-3458	1.2805e-766	8.0443e-39	0.00059755
14	1.5965e-3459	9.7027e-3454	2.588e-3065	4.6785e-154	1.5131e-14
15	-	-	-	5.3531e-615	6.2236e-57
16	-	-	-	9.1742e-2459	1.7811e-226
17	-	-	-	1.9155e-3437	1.1949e-904
18	-	-	-	-	7.2529e-3432

then (22) *tells us that* $L_{k,n}$ *approximates* $\log A$ *when* $k, n \uparrow \infty$. *This, with* $V_{0,n} = I$ *in* (21), *leads to the following estimations. See Table 3. As an approximative value of* $\log A$ *we can take*

$$\log A \approx L_{14,2} \approx \begin{pmatrix} 0.151871071925 & 3.47189006352 & 0.336791127306 & 0.850370892666 \\ 3.47189006352 & -3.13785518545 & 1.6269720327 & -0.153505935774 \\ 0.336791127306 & 1.6269720327 & 0.926046956447 & 1.55749530144 \\ 0.850370892666 & -0.153505935774 & 1.55749530144 & 1.43832101511 \end{pmatrix}.$$

Acknowledgment. The authors would like to express their thanks to Professor Ibrahim A. Al-Subaihi from Taibah University for his helpful discussions. They would also like to thank the Editor-in-chief, Professor Sui-Sun Cheng, and the anonymous referee for their valuable comments and suggestions which have been included in final version of this manuscript.

References

[1] N. Alharbi and M. Raïssouli, On an iterative algorithm converging to the solution of $XCX = D$, Afr. Mat., 31(2020), 997–1007.

[2] C. E. Froberg, Introduction to Numerical Analysis, Second edition Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969.

[3] N. J. Higham, Newton's method for the matrix square root, Math. Comp., 46(1986), 537–549.

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

[5] B. Iannazzo, On the Newton method for the matrix pth root, SIAM J. Matrix Anal. Appl. 28/2 (2006), 503–523.

[6] E. Jarman, Nth Roots of Matrices, Thesis (2011), Univ. Central Missouri.