Article

On High-Order Iterative Schemes for the Matrix pth Root Avoiding the Use of Inverses

Sergio Amat [1], Sonia Busquier [1], Miguel Ángel Hernández-Verón [2] and Ángel Alberto Magreñán [2,*]

[1] Department of Applied Mathematics and Statistics, Polytechnic University of Cartagena, 30203 Cartagena, Spain; sergio.amat@upct.es (S.A.); sonia.busquier@upct.es (S.B.)
[2] Department of Mathematics and Computation, University of La Rioja, 26006 Logroño, Spain; mahernan@unirioja.es
[*] Correspondence: angel-alberto.magrenan@unirioja.es

Abstract: This paper is devoted to the approximation of matrix pth roots. We present and analyze a family of algorithms free of inverses. The method is a combination of two families of iterative methods. The first one gives an approximation of the matrix inverse. The second family computes, using the first method, an approximation of the matrix pth root. We analyze the computational cost and the convergence of this family of methods. Finally, we introduce several numerical examples in order to check the performance of this combination of schemes. We conclude that the method without inverse emerges as a good alternative since a similar numerical behavior with smaller computational cost is obtained.

1. Introduction

The computation of operators on a matrix appears in many applications. The use of iterative methods has emerged as a useful technique to approximate it. This paper is devoted to the approximation of the matrix pth root. We recall that the principal pth root of a matrix $A \in \mathbb{C}^{r \times r}$, where $\mathbb{C}$ is a set of complex numbers, without positive eigenvalues in $\mathbb{R}$, as the unique solution X of

$$X^p - A = 0,$$

with an argument of the eigenvalues in modulus lower than

$$\frac{\pi}{p}.$$

For this kind of problem, the well-known and famous Newton method takes the following form

$$X_0 \text{ given}, \quad X_{n+1} = \frac{(p-1)X_n + AX_n^{1-p}}{p}, \quad n \geq 0.$$

This formula is a direct extension of the method applied to the scalar equation $x^p - a = 0$. In this form, the method achieves quadratic convergence, but unfortunately, it is unstable [1]. However, as can be in seen in [2], a stable version can be found.

In fact, several applications of the Newton method exist for finding the pth root matrix, see this incomplete list of references [1–6].

Similarly, we can derive the third-order Chebyshev method as

$$X_0 \in \Theta, \quad X_{n+1} = \frac{2p^2 - 3p + 1}{2p^2} X_n + \frac{2p - 1}{p^2} A X_n^{1-p} - \frac{p-1}{2p^2} A^2 X_n^{1-2p}, \quad n \geq 0, \quad (1)$$

where $\Theta = B \in \in \mathbb{C}^{r \times r}$ s.t. B has no nonpositive real eigenvalues.

In our paper [7], we proposed stable versions of this algorithm, we presented some numerical advantages of it with respect to Newton and other third-order methods such as Halley's method [8,9], and finally, we developed a general family of any order that includes both the Newton and Chebyshev methods.

In order to develop a method avoiding the use of inverses of the different iterates, we can consider the approximation of this less natural equation $\frac{1}{x^p} - \frac{1}{a} = 0$. In this case, the method presented in [7] has the form

$$\begin{cases} X_0 \text{ given,} \\ X_{n+1} = X_n \sum_{k=0}^{m} \frac{d_k}{k!} \left(I - A^{-1}X_n^p\right)^k, \quad n \geq 0, \end{cases} \quad (2)$$

where

$$d_0 = 1 \quad \text{and} \quad d_k = \prod_{i=0}^{k-1} \left(\frac{1}{p} + i\right), \quad k \geq 1.$$

This method has order m, and in particular, for $m = 2$ and $m = 3$, we recover the Newton and Chebyshev methods.

In the above Formula (2), we only need to compute the inverse of A. In the present paper, we propose to approximate the inverse of A by another iterative method. We can use our family introduced in [10] that has the form

$$\begin{cases} Y_0 \text{ given,} \\ Y_{n+1} = \sum_{k=0}^{m} \binom{m+1}{k+1}(-1)^k Y_n (AY_n)^k, \quad n \geq 0. \end{cases} \quad (3)$$

Our goal is to see that the new approach yields a similar numerical behavior but with the advantage that it is free of inverse operators, and in particular, has a smaller computational cost, which means that it requires fewer operations.

The computation of the pth root of a matrix appears, for example, in fractional differential equations, discrete representations of norms corresponding to finite element discretizations of fractional Sobolev spaces, and the computation of geodesic-midpoints in neural networks (see [11] and the references therein).

2. A General Method for Approximating the Matrix pth Root Free of Inverse Operators

As we mentioned in the introduction, for approximating $A^{\frac{1}{p}}$, we propose the use of a combination of two families. One for the approximation of A^{-1} and other, that use this approximation, for the approximation of the matrix pth root.

First step: for approximating A^{-1}

$$\begin{cases} Y_0 \text{ given,} \\ Y_{n+1} = \sum_{k=0}^{m} \binom{m+1}{k+1}(-1)^k Y_n (AY_n)^k, \quad n = 0, \dots, L. \end{cases} \quad (4)$$

Second step: for approximating $A^{\frac{1}{p}}$

$$
\begin{cases}
X_0 \text{ given,} \\
X_{n+1} = X_n \displaystyle\sum_{k=0}^{m} \frac{d_k}{k!} \left(I - Y^L X_n^p \right)^k, \quad n \geq 0,
\end{cases}
\tag{5}
$$

where Y^L denotes the final iteration computed by the above method (4) in the first step.

2.1. Convergence

To establish the convergence of the general method (5), we observe that

$$
\begin{aligned}
I - A^{-1} X_n^p &= I - Y^L X_n^p + Y^L X_n^p - A^{-1} X_n^p \\
&= (I - Y^L X_n^p) + (Y^L - A^{-1}) X_n^p.
\end{aligned}
\tag{6}
$$

Now, suppose that, from method (4), we consider $n_0 \in \mathbb{N}$ such that $Y_{n_0} = Y^L$. Then, for the iterative process (4), using Theorem 3.2 in [10], we obtain

$$
\|Y^L - A^{-1}\| \leq \|A\|^m \|Y_{n_0-1} - A^{-1}\|^{m+1} \leq \dots \|A\|^{(m+1)^{n_0}-1} \|Y_0 - A^{-1}\|^{(m+1)^{n_0}}.
\tag{7}
$$

On the other hand, for the iterative process (2), from Theorem 4.1 in [7], we have

$$
\|I - Y^L X_n^p\| < \|I - Y^L X_{n-1}^p\|^{m+1} < \dots < \|I - Y^L X_0^p\|^{(m+1)^n}.
\tag{8}
$$

Finally, fixed Y^L we take X_0 such that $\|I - Y^L X_0^p\| < 1$. Then, from Theorem 4.1 in [7], the sequence $\{X_n\}$, given by the general method (5), converges to $(Y^L)^{\frac{1}{p}}$; therefore, $\{X_n\}$ is bounded. So, there exists $M > 0$ such that $\|X_n\| \leq M$ for all $n \in \mathbb{N}$.

Theorem 1. *Suppose Y_0 with $\|Y_0 - A^{-1}\| < 1$ and $n_0 \in \mathbb{N}$ such that, from (4), we consider $Y^L = Y_{n_0}$. If X_0 is such that $\|I - Y^L X_0^p\| < 1$, then for all tolerance Tol > 0 given, there exists $n^* \in \mathbb{N}$ such that $\|I - A^{-1} X_n^p\| < $ Tol for $n \geqslant n^*$.*

Proof. Given $Tol > 0$, from (8), there exists $n_1 \in \mathbb{N}$ such that $\|I - Y^L X_n^p\| < \frac{Tol}{2}$ for all $n \geqslant n_1$.

On the other hand, as X_n converges to $(Y^L)^{\frac{1}{p}}$, from Theorem 4.1 in [7], there exists $n_2 \in \mathbb{N}$ such that $\|(Y^L)^{\frac{1}{p}} - X_n\| < Tol$, for all $n \geqslant n_2$. Then, $\|X_n\| \leq Tol + \|(Y^L)^{\frac{1}{p}}\|$ for all $n \geqslant n_2$.

Moreover, from (7), given $\dfrac{Tol}{2(Tol + \|(Y^L)^{\frac{1}{p}})} > 0$, there exists $n_3 \in \mathbb{N}$ such that $\|Y^L - A^{-1}\| < \frac{Tol}{2}$ for all $n \geqslant n_3$.

Therefore, from (6), there exists $n^* = max\{n_1, n_2, n_3\} \in \mathbb{N}$ such that $\|I - A^{-1} X_n^p\| < Tol$ for all $n \geqslant n^*$. $\quad\square$

2.2. Computational Cost and Efficiency

If A is a matrix $q \times q$, and taking into account that the computational cost of the product of two matrices is q^3 and that adding or subtracting two matrices has a computational cost of q^2, then the computational cost of

- the computation of matrix Y^L, given by means of using (4), is $CC_I(m, q) = mq^2 + (m+1)q^3$ and its computational efficiency is $CE_I(m, q) = (m+1)^{\frac{1}{mq^2+(m+1)q^3}}$
- the computation of method (2) has a computational cost of $CC_R(m, q) = mq^2 + (p+m)q^3$ and its computational efficiency is $CE_R(m, q) = (m+1)^{\frac{1}{mq^2+(p+m)q^3}}$

- the computational cost of method (5) is obtained directly by $CC(m,q) = 2mq^2 + (p + 2m + 1)q^3$ and its computational efficiency is $CE(m,q) = (m+1)^{\frac{1}{2mq^2+(p+2m+1)q^3}}$.

In Figures 1 and 2 the computational efficiency for fixed $p = 2$ and $p = 3$ respectively and different values of q and m are shown.
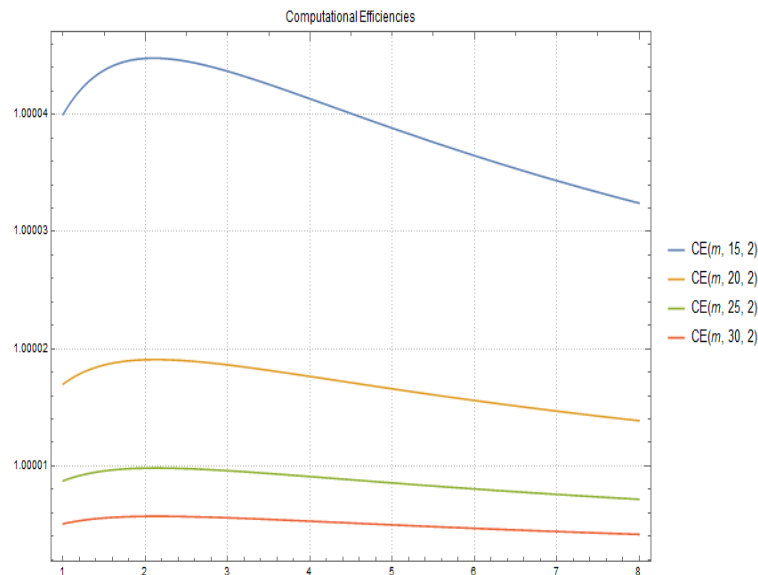

Figure 1. Computational Efficiencies for $p = 2$ and different values of q and m.
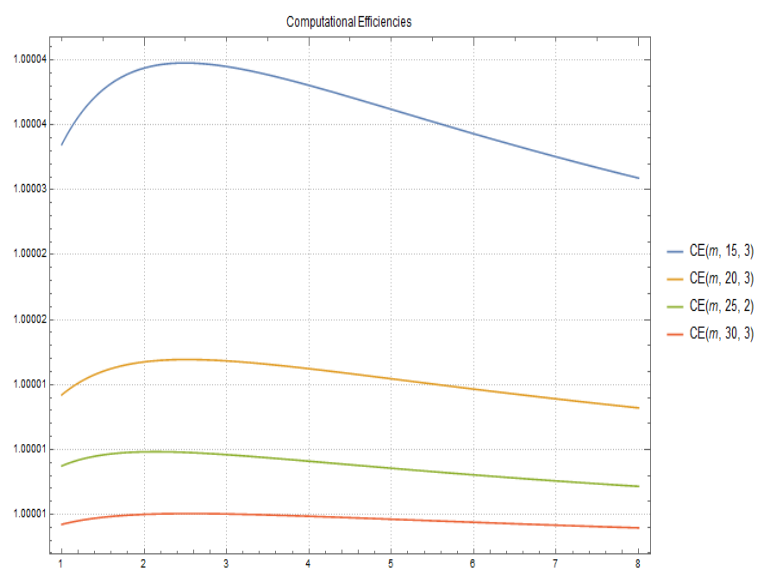

Figure 2. Computational Efficiencies for $p = 2$ and different values of q and m.

In most cases, when working with a family of methods, the best one is related to the problem considered (see the next figures) and the main way used to solve the problem. For example, the most efficient method (in terms of computation) for solving sparse problems, which appear when discretizing differential equations, is Chebyshev's method [7,10].

In Figures 3 and 4 the computational efficiency for fixed $q = 15$ and $q = 20$ respectively and different values of p and m are shown.
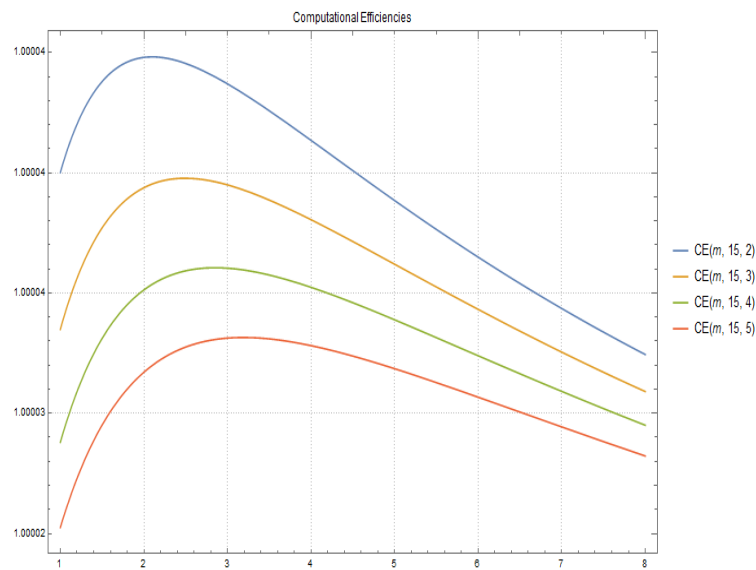
Figure 3. Computational Efficiencies for $q = 15$ and different values of p and m.
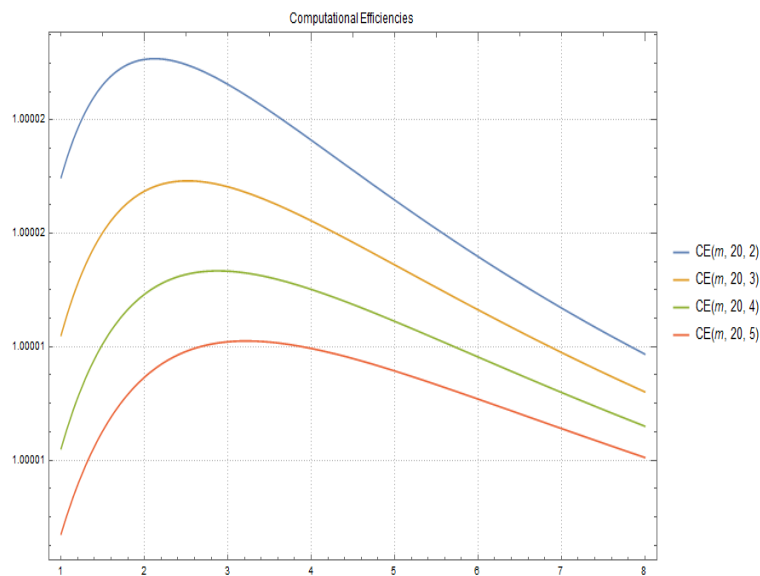

Figure 4. Computational Efficiencies for $q = 20$ and different values of p and m.

3. Applications Related to Differential Equations

In this section, we present a comparison of the original method (2) using A^{-1} and the new proposal (5), avoiding the use of any inverse operator. We refer to our paper [7] in order to see the advantages of the original method in comparison with other methods that appear in the literature.

We consider the approximation of the pth root of two matrices related to the discretization of differential equations. This type of matrix operations appears in the approximation of space-fractional diffusion problems [12].

We start with the following matrix

$$
A = \begin{pmatrix}
C & D & & & & & \\
B & C & D & & & 0 & \\
 & B & C & D & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & 0 & & B & C & D & \\
 & & & & B & C &
\end{pmatrix},
\tag{9}
$$

where $B = -1 - \frac{h}{2}w$, $C = 2 + h^2 v$ and $D = -1 + \frac{h}{2}w$.

The matrix (9) can be seen as the result of applying a discretization process, using finite differences to the boundary value problem defined as

$$
x''(t) = v \cdot x(t) + w \cdot x'(t),
$$

where

$$
x(a) = x_a \text{ and } x(b) = x_b,
$$

Since the matrix A is sparse, the most efficient methods in both families are Chebyshev-like methods.

In Table 1, we compare our original method using A^{-1} (2) and the new combination using the approximation Y^L of the inverse (5). We observe a similar numerical behavior of both methods. Thus, the approximation of the inverse seems a good alternative since has similar errors with smaller computational cost.

Table 1. Error for the approximation of $A^{1/p}$, taking $(m, h, v, w) = (100,\ 0.01,\ 20{,}000,\ -10)$ and four iterations.

p	Original Method Computing A^{-1}	New Combined Method with Smaller Cost
2	1.4845×10^{-11}	1.4845×10^{-11}
4	3.6643×10^{-13}	3.6639×10^{-13}
6	3.1660×10^{-13}	3.1655×10^{-13}
8	3.2326×10^{-13}	3.2326×10^{-13}

Finally, we compute the pth matrix roots for the matrix

$$
A = \begin{pmatrix}
1 - 2\lambda & \lambda & & & & & \\
\lambda & 1 - 2\lambda & \lambda & & & 0 & \\
 & \lambda & 1 - 2\lambda & \lambda & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & 0 & & \lambda & 1 - 2\lambda & \lambda & \\
 & & & & \lambda & 1 - 2\lambda &
\end{pmatrix}
$$

This matrix is related to the discretization, using finite differences, of the laplacian operator that appear in many mathematical models. In Table 2, we observe again a similar numerical behavior of both schemes.

Table 2. Error for the approximation of $A^{1/p}$ taking $m = 100$, $k = 2 \times 10^{-6}$, $k/h^2 = 2 \times 10^{-2}$ and three iterations.

p	Original Method Computing A^{-1}	New Combined Method with Smaller Cost
2	2.1204×10^{-14}	2.1208×10^{-14}
4	1.8486×10^{-14}	1.8484×10^{-14}
6	1.7260×10^{-14}	1.7261×10^{-14}
8	2.0723×10^{-14}	2.0724×10^{-14}

4. Conclusions

The function evaluation of a matrix appears in a large and growing number of applications. We have presented and studied a general family of iterative methods without using inverse operators, for the approximation of the matrix pth root. The family incorporates the approximation of A^{-1}, by an iterative method, into another iterative method for approximating $A^{1/p}$. The family includes methods of every order of convergence.

As it appears in [7,10], the most efficient method, in computational terms, for solving sparse problems, which appears when discretizing differential equations, is Chebyshev's method.

Some numerical examples related to the discretization of differential equations have been presented. We have concluded that the new approach (5) has a similar numerical behavior to that of (2) but with the advantage that it is free of inverse operators, and in particular, has a lower computational cost. Finally, we care to mention the existence of other strategies for computing fractional powers of a matrix, which do not use matrix iterations [13,14] or even take into account other techniques such as those appearing in [15].

References

1. Higham, N.J. Newton's method for the matrix square root. *Math. Comp.* **1986**, *46*, 537–549.
2. Iannazzo, B. On the Newton method for the matrix pth root. *SIAM J. Matrix Anal. Appl.* **2006**, *28*, 503–523. [CrossRef]
3. Guo, C.-H.; Higham, N.J. A Schur-Newton method for the matrix pth root and its inverse. *IAM J. Matrix Anal. Appl.* **2006**, *28*, 788–804. [CrossRef]
4. Iannazzo, B. A note on computing the matrix square root. *Calcolo* **2003**, *40*, 273–283. [CrossRef]
5. Petkov, M.; Boršukova, S. A modified Newton method for obtaining roots of matrices. *Annu. Univ. Sofia Fac. Math.* **1974**, *66*, 341–347. (In Bulgarian)

6. Smith, M.I. A Schur algorithm for computing matrix *p*th roots. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 971–989. [CrossRef]
7. Amat, S.; Ezquerro, J.A.; Hernández-Verón, M.A. On a new family of high-order iterative methods for the matrix*p*th root. *Numer. Linear Algebra Appl.* **2015**, *22*, 585–595. [CrossRef]
8. Guo, C.-H. On Newton's method and Halley's method for the principal *p*th root of a matrix. *Linear Algebra Appl.* **2010**, *432*, 1905–1922. [CrossRef]
9. Iannazzo, B. A family of rational iterations and its application to the computation of the matrix *p*th root. *SIAM J. Matrix Anal. Appl.* **2008**, *30*, 1445–1462. [CrossRef]
10. Amat, S.; Ezquerro, J.A.; Hernández-Verón, M.A. Approximation of inverse operators by a new family of high-order iterative methods. *Numer. Linear Algebra Appl.* **2014**, *21*, 629–644. [CrossRef]
11. Higham, N.J.; Al-Mohy, A.H. *Computing Matrix Functions*; The University of Manchester, MIMS EPrint: Manchester, UK, 2010; Volume 18.
12. Szekeres, B.J.; Izsák, F. Finite difference approximation of space-fractional diffusion problems: The matrix transformation method. *Comput. Math. Appl.* **2017**, *73*, 261–269. [CrossRef]
13. Iannazzo, B.; Manasse, C. A Schur logarithmic algorithm for fractional powers of matrices. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 794–813. [CrossRef]
14. Higham, N.J.; Lin, L. An improved Schur–Pade algorithm for fractional powers of a matrix and their Fréchet derivatives. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 1341–1360. [CrossRef]
15. Marino, G.; Scardamaglia, B.; Karapinar, E. Strong convergence theorem for strict pseudo-contractions in Hilbert spaces. *J. Inequalities Appl.* **2016**, *2016*, 134. [CrossRef]