Mathematics Interdisciplinary Research **11** (1) (2026) 65 − 92

Original Scientific Paper

Developing an Efficient Interval Iterative Method for Computing Enclosures for the Matrix pth Root

Mandana Moccari⋆

Abstract

This paper introduces a verified interval iterative method for computing the principal pth root of a square matrix along with rigorous interval enclosures. Leveraging the epsilon inflation technique, the proposed algorithm is reformulated as an inclusion method, enabling robust control over approximation and rounding errors in finite-precision arithmetic. The method exhibits quadratic convergence and does not require an initial enclosure containing the exact root, which is a common limitation in existing interval approaches. We further demonstrate that the midpoint matrix sequence generated by the iteration is well-behaved and numerically stable. Theoretical analysis confirms the convergence of the interval enclosures to the exact matrix root, and numerical experiments validate the method's efficiency for large-scale matrices and high values of p. As a practical contribution, we provide implementable Mathematica code for the proposed algorithm, facilitating reproducibility and further exploration.

Keywords: Iterative method, pth Root of a matrix, Interval matrix, Order of convergence, Inclusion method.

2020 Mathematics Subject Classification: 15A24; 65F30.

How to cite this article
M. Moccari, Developing an efficient interval iterative method for computing enclosures for the matrix pth root, *Math. Interdisc. Res.* **11** (1) (2026) 65-92.

⋆Corresponding author (E-mail: m-moccari@iau.ac.ir)
Academic Editor: Davod Khojasteh Salkuyeh
Received 4 June 2025, Accepted 24 August 2025
DOI: 10.22052/MIR.2025.256982.1523

1. Introduction

Interval analysis [1] is a vital branch of numerical computation that provides mathematically rigorous enclosures for solving equations and controlling rounding, propagation, and approximation errors [2, 3]. Interval methods have found widespread applications in various scientific and engineering domains, including structural mechanics, beam physics, computer graphics, and computer-assisted proofs. [4–6].

Let $A \in \mathbb{C}^{n\times n}$ be a square matrix and $p \geq 2$ be an integer. A matrix X is called a pth root of A if $X^p = A$ [7]. Among all such roots, the principal pth root is of particular interest.

Theorem 1.1. *Assume that A is an $n \times n$ matrix with no eigenvalue in the negative real part of the imaginary axis. Then the principal pth root of A is the unique pth root with eigenvalue arguments in the open interval $(-\pi/p, \pi/p)$, and it is a primary matrix function of A. The principal pth root of A is denoted by $A^{1/p}$.*

Proof. The proof of this theorem is detailed in [7], Theorem 7.2. □

Computing the pth root of a matrix A is equivalent to solving the matrix equation $X^p - A = 0$ [7]. The matrix pth root problem is significant in numerous applications, such as computing the sign of a matrix, polar decomposition, geometric mean, and matrix p-sector functions [8–13].

Newton's iterative method is widely used for finding the pth root of a square matrix A with no eigenvalue on the left half-plane. This method is derived from a generalized Sylvester equation [7] and is given by:

$$X_{k+1} = \frac{1}{p}\left[(p-1)X_k + X_k^{1-p}A\right], \quad X_0A = AX_0. \tag{1}$$

The convergence of Newton's iterative method is discussed in [7], Corollary 7.8. To enhance the stability of Newton's method, N. Higham [7] introduced a symmetric form, leading to the stable iterative method:

$$\begin{cases} X_{k+1} = X_k\left(\frac{(p-1)I+M_k}{p}\right), & X_0 = I, \\ M_{k+1} = \left(\frac{(p-1)I+M_k}{p}\right)^{-p} M_k, & M_0 = A. \end{cases} \tag{2}$$

Since interval methods can control and enclose rounding errors, this study aims to generalize method (2) into an interval method. Before proceeding, we introduce the necessary notation and concepts related to interval matrices. An interval matrix $\mathcal{X}$ is defined as a matrix whose elements are intervals, i.e., $\mathcal{X} = (\mathcal{X}_{ij})$, where $\mathcal{X}_{ij} = [a_{ij}, b_{ij}]$ for each i, j. The midpoint matrix function $m(\mathcal{X})$ is defined as the matrix whose elements are the midpoints of the corresponding intervals in $\mathcal{X}$.

The extended interval method for computing the pth root of a matrix A is defined as:

$$\begin{cases} \mathcal{X}_{k+1} = m(\mathcal{X}_k) - \frac{\mathcal{X}_k}{p}(I - G_k), \\ G_{k+1} = \left(\frac{(p-1)I+G_k}{p}\right)^{-p} G_k, \end{cases} \tag{3}$$

for $k = 0, 1, \ldots$. Here, m denotes the midpoint matrix function, and all interval operations are performed with outward rounding to ensure that the computed intervals contain the exact results. The initial matrices $\mathcal{X}_0$ and G_0 should be chosen such that $m(\mathcal{X}_0) = I$ and $G_0 = A$. Thus:

$$\mathcal{X}_0 = (\mathcal{X}_{ij}), \quad \text{where} \quad \mathcal{X}_{ij} = [I_{ij} - \Delta_{ij}, I_{ij} + \Delta_{ij}], \quad 1 \le i, j \le n, \tag{4}$$

where the matrix (Δ_{ij}) represents a positive arbitrary matrix in $M_{n\times n}(\mathbb{R})$.

The midpoint iterative method of (3) is:

$$\begin{cases} m(\mathcal{X}_{k+1}) = m(\mathcal{X}_k)\left(\frac{(p-1)I+G_k}{p}\right), \quad m(\mathcal{X}_0) = I, \\ G_{k+1} = \left(\frac{(p-1)I+G_k}{p}\right)^{-p} G_k, \quad G_0 = A. \end{cases} \tag{5}$$

If the matrix A has no eigenvalues in the left half-plane, the sequences $\{m(\mathcal{X}_k)\}$ and $\{G_k\}$ converge quadratically to $A^{1/p}$ and I, respectively, according to N. Higham [7] and B. Iannazzo [14].

Inspired by the epsilon inflation technique [15, 16], the interval iterative method is adapted to an inclusion method. Additionally, intervals are rounded to a few-digit precision to control rounding errors. This adaptation represents the novelty and primary goal of this study. Furthermore, the implementation of the matrix pth root via interval extension is provided in an elegant Mathematica code, included in the Appendix, to support the numerical demonstration of the proposed iteration. Also, this paper demonstrates the well-behavior of the iteration (2) through its interval extension. An advantage of the proposed interval method is that it does not require an initial matrix containing the pth root of the given matrix, which is a complex problem for some interval methods [17].

The paper is organized as follows: Section 2 reviews definitions and results related to interval matrices and the epsilon inflation technique. Section 3 presents the convergence analysis of the proposed interval iterative method, including its order of convergence and the well-behaved nature of the midpoint matrix method. Section 4 confirms the theoretical results through numerical examples, where the Mathematica implementation referred to as pAlgorithm is used to compute the principal pth root of a matrix. The corresponding code is provided in the Appendix.

2. Preliminaries

In this section, definitions and results related to interval matrices and the epsilon inflation technique are adopted primarily from [18, 19].

2.1 Interval matrices and their operations

Definition 2.1. The set of all real numbers defined as follows is called the interval $[a, b]$:

$$[a, b] = \{x \in \mathbb{R};\ a \leq x \leq b\}. \tag{6}$$

If $a = b$, then the interval $[a, b]$ is called a point interval or a degenerated interval. For two intervals $[a_1, a_2]$ and $[b_1, b_2]$, interval operations are calculated as follows:

$$\begin{aligned} &[a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2], \\ &[a_1, a_2] - [b_1, b_2] = [a_1 - b_2, a_2 - b_1], \\ &[a_1, a_2] \cdot [b_1, b_2] = [\min\{a_1b_1, a_1b_2, a_2b_1, a_2b_2\}, \max\{a_1b_1, a_1b_2, a_2b_1, a_2b_2\}]. \end{aligned} \tag{7}$$

These definitions and operations are based on [18, 19].

$M_n(I(\mathbb{R}))$ is introduced as the set of all $n \times n$ matrices whose components are real intervals. In this study, we assume that all matrices are square matrices, i.e., $m = n$. This assumption is necessary to ensure that all operations, such as matrix multiplication, are well-defined. An interval matrix is shown as $\mathcal{X} = (\mathcal{X}_{ij})$, where $\mathcal{X}_{ij} \in I(\mathbb{R})$, $1 \leq i \leq n$, and $1 \leq j \leq n$. For $A = (A_{ij})$, if all components of A, A_{ij}, are point intervals, A is a point matrix which belongs to $M_n(\mathbb{R})$.

Definition 2.2. Let $\mathcal{A} = (\mathcal{A}_{ij})$ and $\mathcal{B} = (\mathcal{B}_{ij})$ be two interval matrices and belong to $M_n(I(\mathbb{R}))$, then the following statements are valid:

$$\mathcal{A} \subseteq \mathcal{B} \Longleftrightarrow \mathcal{A}_{ij} \subseteq \mathcal{B}_{ij}, \quad 1 \leq i \leq n, 1 \leq j \leq n, \tag{8}$$

and, if $A = (a_{ij})$ is a point matrix, then

$$A \in \mathcal{B} \Longleftrightarrow a_{ij} \in \mathcal{B}_{ij}, \quad 1 \leq i \leq n, 1 \leq j \leq n. \tag{9}$$

Also,

$$\mathcal{A} \pm \mathcal{B} = (\mathcal{A}_{ij} \pm \mathcal{B}_{ij}), \tag{10}$$

and if $\mathcal{C} \in M_n(I(\mathbb{R}))$ and $D \in M_n(I(\mathbb{R}))$, then

$$\mathcal{C}\mathcal{D} = (\sum_{v=1}^{n} \mathcal{C}_{iv}\mathcal{D}_{vj}). \tag{11}$$

Let $A = (a_{ij})$ and $B = (b_{ij})$, belong to $M_n(\mathbb{R})$. The partial order is defined by

$$A \leq B \Longleftrightarrow a_{ij} \leq b_{ij}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq n, \tag{12}$$

for $n \times n$ matrices (see [18, 19]).

Definition 2.3. The absolute value of an interval matrix $\mathcal{A} = (\mathcal{A}_{ij}) \in M_n(I(\mathbb{R}))$ is

$$|\mathcal{A}| = (|\mathcal{A}_{ij}|) = (\max_{a \in \mathcal{A}_{ij}} |a|), \tag{13}$$

and the width of the interval matrix $\mathcal{A}$ is the following nonnegative matrix:

$$d(\mathcal{A}) = (d(\mathcal{A}_{ij})) = (\max(\mathcal{A}_{ij}) - \min(\mathcal{A}_{ij})). \tag{14}$$

The following properties are derived from the preceding relations and hold for the absolute value and width of real interval matrices (see [18]):

1. $\mathcal{A} \subseteq \mathcal{B} \Rightarrow d(\mathcal{A}) \leq d(\mathcal{B})$,
2. $d(\mathcal{A} \pm \mathcal{B}) = d(\mathcal{A}) + d(\mathcal{B})$,
3. $d(\mathcal{A}) = \sup_{A,B\in\mathcal{A}} |A - B|$,
4. $|\mathcal{A}| = \sup_{A\in\mathcal{A}} |A|$,
5. $\mathcal{A} \subseteq \mathcal{B} \Rightarrow |\mathcal{A}| \leq |\mathcal{B}|$,
6. $d(A\mathcal{B}) = |A|d(\mathcal{B}), \quad d(\mathcal{B}A) = d(\mathcal{B})|A|$,
7. $d(\mathcal{A}\mathcal{B}) \leq d(\mathcal{A})|\mathcal{B}| + |\mathcal{A}|d(\mathcal{B})$.

Definition 2.4. The Hausdorff distance between two interval matrices $\mathcal{A}$ and $\mathcal{B}$ in $M_n(I(\mathbb{R}))$ is denoted by (see [18]):

$$q(\mathcal{A}, \mathcal{B}) = (q(\mathcal{A}_{ij}, \mathcal{B}_{ij})) \in M_n(\mathbb{R}). \tag{15}$$

If $\mathcal{A}_{ij} = [a_1, a_2]$ and $\mathcal{B}_{ij} = [b_1, b_2]$, then

$$q(\mathcal{A}_{ij}, \mathcal{B}_{ij}) = \max\{|a_1 - b_1|, |a_2 - b_2|\}. \tag{16}$$

Definition 2.5. The point matrix $m(\mathcal{A})$ is the midpoint matrix of the interval matrix $\mathcal{A} = (\mathcal{A}_{ij})$ and is obtained by (see [18]):

$$m(\mathcal{A}) = (m(\mathcal{A}_{ij})) = \left(\frac{max(\mathcal{A}_{ij}) + min(\mathcal{A}_{ij})}{2}\right). \tag{17}$$

The following properties hold for $\mathcal{A}, \mathcal{B} \in M_n(I(\mathbb{R}))$ and $A \in M_n(\mathbb{R})$:

1. $m(\mathcal{A} \pm \mathcal{B}) = m(\mathcal{A}) \pm m(\mathcal{B})$,
2. $m(A\mathcal{B}) = Am(\mathcal{B})$, $m(\mathcal{B}A) = m(\mathcal{B})A$,
3. $m(A) = A$.

2.2 Introducing the inclusion interval method inspired by the epsilon inflation technique

Digital devices cannot store or perform computations with an infinite set of real numbers. They approximate real numbers using a finite set of floating-point numbers, represented in the form $m \cdot b^e$, where m is the mantissa, b is the base, and e is the exponent [17]. The set of such machine-representable numbers is denoted by $\mathbb{R}_M$.

Directed rounding: Directed rounding is an essential concept in numerical computations. If x is a real number, $\downarrow x$ represents downward-directed rounding, and $\uparrow x$ represents upward-directed rounding [17]. Therefore, the following inequality holds:

$$\downarrow x \leq fl(x) \leq \uparrow x. \tag{18}$$

In this paper, the following Mathematica codes are used to implement directed rounding with s-digit precision:

```
down[x_, s_] := N[x - 10^-s (x*10^s - Floor[x*10^s]), s];
up[x_, s_] := N[x + 10^-s (Ceiling[x*10^s] - x*10^s), s];
```

Machine intervals: Intervals are also rounded in digital devices. Consider a real interval $[a, b]$. The machine rounding of $[a, b]$ can be represented as:

$$\updownarrow [a, b] = \{x \in \mathbb{R}; \downarrow a \leq x \leq \uparrow b\}, \quad a \leq b. \tag{19}$$

The set of all machine intervals is denoted by $I(\mathbb{R}_M)$. The interval rounding operation has the following properties:

$$[a, b] \subseteq \updownarrow [a, b], \quad \text{and} \quad [a, b] \subseteq [c, d] \Rightarrow \updownarrow [a, b] \subseteq \updownarrow [c, d]. \tag{20}$$

For an interval matrix, the rounding is applied element-wise (see [17]).

Controlling rounding errors: Let $x, y \in \mathbb{R}$, and suppose $x \in [a, b]$ and $y \in [c, d]$. For any arithmetic operation $* \in \{+, -, \cdot, /\}$, the following inclusion holds:

$$x * y \in [a, b] * [c, d] \subseteq [a, b]_M * [c, d]_M \subseteq ([a, b]_M * [c, d]_M)_M, \tag{21}$$

where $[a, b]_M$ denotes a machine interval. Thus, while x and y are unknown real numbers, the operation $x * y$ belongs to a known machine interval. This implies that rounding errors are well-controlled, making interval methods highly useful in such scenarios [20].

To show the convergence of the interval method (3), we need a nested sequence of interval matrices that contain the solution. We will derive such a sequence by employing the epsilon inflation technique, which is inspired by Definition 4 in [16].

Definition 2.6. Let $\mathcal{X}$ be an $n \times n$ real interval matrix. The *epsilon inflation* of $\mathcal{X}$, denoted by $\mathcal{X} \circ \epsilon$, is defined as

$$\mathcal{X} \circ \epsilon = \mathcal{X} + U_\epsilon(0), \quad \text{where} \quad 0 < \epsilon \in \mathbb{R}, \tag{22}$$

and $U_\epsilon(0)$ is an $n \times n$ interval matrix of the form

$$U_\epsilon(0) = \left([-\epsilon_{ij}, \epsilon_{ij}]\right), \quad 0 < \epsilon_{ij} \in \mathbb{R}, \quad 1 \leq i, j \leq n. \tag{23}$$

In the context of the interval iteration method (3), let $\mathcal{X}_{k+1} = (\mathcal{X}_{k+1}^{ij}) = \left([a_{k+1}^{ij}, b_{k+1}^{ij}]\right)$ be the computed interval matrix at iteration $k+1$. To apply a uniform inflation, we define a scalar ϵ_{k+1} as

$$\epsilon_{k+1} = \max_{1 \leq i,j \leq n} \left\{ \max \left\{ a_{k+1}^{ij} - \downarrow a_{k+1}^{ij}, \uparrow b_{k+1}^{ij} - b_{k+1}^{ij} \right\} \right\}, \tag{24}$$

where $\downarrow a$ and $\uparrow b$ denote the outward rounding of a and b, respectively. Then, the matrix $U_\epsilon^{k+1}(0)$ is defined as

$$U_\epsilon^{k+1}(0) := \left([-\epsilon_{k+1}, \epsilon_{k+1}]\right)_{n \times n},$$

which means that we set $\epsilon_{ij} := \epsilon_{k+1}$ for all $1 \leq i, j \leq n$ to ensure a consistent inflation across all entries. As a result, the inflated matrix is given by

$$\tilde{\mathcal{X}}_{k+1} := \mathcal{X}_{k+1} + U_{\epsilon_{k+1}}(0), \tag{25}$$

and it satisfies the inclusion relation

$$\mathcal{X}_{k+1} \subseteq\updownarrow \mathcal{X}_{k+1} \subseteq \tilde{\mathcal{X}}_{k+1}. \tag{26}$$

Moreover, the midpoint matrix remains unchanged under this inflation:

$$m(\mathcal{X}_{k+1}) = m(\tilde{\mathcal{X}}_{k+1}), \quad \text{for all } k \geq 0. \tag{27}$$

Lemma 2.7. *The sequence $\{U_{\epsilon_k}(0)\}$ converges to the zero matrix.*

Proof. The bounds of each interval matrix $\mathcal{X}_k$ are assumed to be rounded using directed rounding with δ-digit precision, where $\delta \geq 2$. Based on the update rule

$$\epsilon_{k+1} = \min \left\{ \epsilon_{k+1}, \frac{\epsilon_k}{\mu} \right\}, \quad \mu \geq 2,$$

we have $\epsilon_{k+1} < \epsilon_k$ for all k. From Equation (24) and this monotonic decrease, it follows that

$$U_{\epsilon_{k+1}}(0) \subset U_{\epsilon_k}(0), \quad \text{for all } k. \tag{28}$$

Hence, the sequence $\{U_{\epsilon_k}(0)\}$ is nested. Since $0 \in U_{\epsilon_k}(0)$ for every k, and applying Corollary 8.10 of [17] together with Theorem 3.10 of [21], we conclude that the intersection of this nested sequence converges to the zero matrix. □

Corollary 8.10 of [17]. Let $\{\mathcal{A}_k\}_{k=0}^{\infty}$ be a sequence of $m \times n$ interval matrices such that

$$\mathcal{A}_0 \supseteq \mathcal{A}_1 \supseteq \mathcal{A}_2 \supseteq \cdots \tag{29}$$

Then the sequence converges to an interval matrix $\mathcal{A} = \bigcap_{k=0}^{\infty} \mathcal{A}_k$.

Theorem 3.10 of [21]. Let $\{K_n\}$ be a sequence of compact sets in a topological space such that $K_{n+1} \subset K_n$ and $\lim_{n\to\infty} \operatorname{diam}(K_n) = 0$. Then $\bigcap_{n=1}^{\infty} K_n$ is a singleton.

Inclusion method:
Now, the interval method (3) can be rewritten as follows:

$$\begin{cases} \tilde{\mathcal{X}}_{k+1} = m(\mathcal{X}_k) - \dfrac{\tilde{\mathcal{X}}_k}{p}(I - G_k) + U_{\epsilon_{k+1}}, \quad m(\tilde{\mathcal{X}}_0) = m(\mathcal{X}_0) = I, \\ G_{k+1} = \left(\dfrac{(p-1)I + G_k}{p}\right)^{-p} G_k, \quad G_0 = A, \end{cases} \tag{30}$$

for every $k = 0, 1, \ldots$. By relation (27), the midpoint sequence of the method (30) is the same as (5). In the next section, we will prove that the method (30) is an inclusion interval method. Since the sequence $\{U_{\epsilon_k}(0)\}$ converges to zero, the method (30) eventually reduces to the method (3). Thus, the convergence of the method (3) is analogous to the convergence of (30).

3. Main results

In this section, the convergence analysis of the proposed method (30) is studied. Based on the relation (27), the midpoint iterative method of (30) is equivalent to (5). Therefore, the lemmas and theorem, which also hold for midpoint sequence of (30), are necessary to obtain the convergence of the method (30).

Lemma 3.1. *Let $\mathcal{X}_0$ be an interval matrix so that $m(\mathcal{X}_0) = I$, and A is a square matrix whose all eigenvalues are in $\mathbb{R}^+$. In addition, $G_0 = A$. For the sequences $\{\mathcal{X}_k\}$ and $\{G_k\}$ generated by the iterative method (3), the following statements are true for every $k \geq 0$,*

(a) $AG_k = G_k A$,

(b) $Am(\mathcal{X}_k) = m(\mathcal{X}_k)A$,

(c) $G_k = m(\mathcal{X}_k)^{-p} A$.

Proof. (a): For indicating this part, induction on k is used. For $k = 0$, (a) is

trivial, and for $k = n$, assume that it is true. For $k = n + 1$, it is obtained that

$$\begin{aligned} AG_{n+1}^{-1} &= AG_n^{-1}(\frac{(p-1)I + G_n}{p})^p \\ &= \frac{1}{p^p} AG_n^{-1} \sum_{i=0}^{i=p} \binom{p}{i} (p-1)^i G_n^{p-i} \\ &= \frac{1}{p^p} \sum_{i=0}^{i=p} \binom{p}{i} (p-1)^i G_n^{p-i-1} A \\ &= G_{n+1}^{-1} A. \end{aligned} \tag{31}$$

Based on the Corollary 1.41 of [7], if λ_i is an ordering of eigenvalues of G_k, for every k, then the following relation is obtained:

$$\lambda_i(G_{k+1}) = p^p \frac{\lambda_i(G_k)}{((p-1) + \lambda_i(G_k))^p}. \tag{32}$$

Thus, with induction on k, if all of the eigenvalues of A are non-zero, then G_k is invertible for every k.
(b): This study shows part (b) by induction on k. For $k = 0$, if $m(\mathcal{X}_0) = I$, then it will be clear. Then this study assumes that (b) is true for $k = n$. For $k = n+1$, it is deduced that

$$\begin{aligned} Am(\mathcal{X}_{n+1}) &= Am(\mathcal{X}_n) - \frac{Am(\mathcal{X}_n)}{p}(I - G_n) \\ &= m(\mathcal{X}_n)A - \frac{m(\mathcal{X}_n)}{p}(A - G_nA) \\ &= m(\mathcal{X}_n)A - \frac{m(\mathcal{X}_n)}{p}(I - G_n)A \\ &= m(\mathcal{X}_{n+1})A. \end{aligned} \tag{33}$$

(c): First we need to prove the relation $G_k m(\mathcal{X}_k) = m(\mathcal{X}_k)G_k$ for every $k \geq 0$. It can be shown easily by induction on k. Next, again by induction on k, it can be indicated that:

$$G_k = m(\mathcal{X}_k)^{-p} A. \tag{34}$$

Obviously, the induction holds for $k = 0$. For $k = n$, it can be assumed that relation (34) is true. For $k = n + 1$, it can be obtained that:

$$\begin{aligned} G_{n+1} &= (\frac{(p-1)I + G_n}{p})^{-p} G_n \\ &= (\frac{(p-1)I + G_n}{p})^{-p} m(\mathcal{X}_n)^{-p} A \\ &= m(\mathcal{X}_{n+1})^{-p} A. \end{aligned} \tag{35}$$

□

Lemma 3.2. *For a real $n \times n$ matrix A, $|A| = (|a_{ij}|)$ denotes nonnegative matrix ($|a_{ij}| \geq 0$ for every i, j). For $A, B \in \mathbb{R}^{n\times n}$, if $|A| \leq B$, then*

$$\rho(A) \leq \rho(|A|) \leq \rho(B) \leq ||B||, \tag{36}$$

where $||.||$ is any consistent matrix norm, and ρ denotes the spectral radius.

Proof. Refer to the book of N. Higham [7]. □

Corollary 3.3. *([22]). Let D be an invertible matrix. Then $||D^{-1}|| = \frac{1}{\sqrt{\lambda}}$, where λ represents the smallest eigenvalue of D^*D. D^* is the conjugate transpose of the matrix D.*

Theorem 3.4. *Let $\{G_k\}$ be the matrix sequence generated by (3). Assume $G_0 = A$, where A is a square matrix with eigenvalues only in $\mathbb{R}^+$. Hence, the sequence $\{G_k\}$ converges to the identity matrix I with order of convergence 2 and the spectral radius of $I - G_k$ is smaller than 1 for every k.*

Proof. The expression $I - G_{k+1}$ is simplified as follows:

$$\begin{aligned}
I - G_{k+1} &= (I - \frac{1}{p}(I - G_k))^{-p}((I - \frac{1}{p}(I - G_k))^p - G_k) \\
&= (I - \frac{1}{p}(I - G_k))^{-p}(\sum_{i=0}^{p}\binom{p}{i}(\frac{I - G_k}{-p})^i - G_k) \\
&= (I - \frac{1}{p}(I - G_k))^{-p}(\sum_{i=2}^{p}\binom{p}{i}(\frac{I - G_k}{-p})^{i-2})(\frac{I - G_k}{p})^2 \\
&= (I - \frac{1}{p}(I - G_k))^{-p}(\sum_{i=0}^{p-2}\binom{p}{i+2}(\frac{I - G_k}{-p})^i)(\frac{I - G_k}{p})^2 \\
&= (I - \frac{1}{p}(I - G_k))^{-p}(\sum_{i=0}^{p-2}\frac{p(p-1)}{(i+2)(i+1)}\binom{p-2}{i}(\frac{I - G_k}{-p})^i)(\frac{I - G_k}{p})^2.
\end{aligned} \tag{37}$$

Hence, by $\frac{1}{(i+2)(i+1)} \leq \frac{1}{2}$, it is obtained

$$I - G_{k+1} \leq \frac{1}{2}p(p-1)((p-1)I + G_k)^{-2}(I - G_k)^2. \tag{38}$$

Taking norm and considering the Euclidean norm, it is obtained that

$$||I - G_{k+1}|| \leq \frac{1}{2}p(p-1)||((p-1)I + G_k)^{-1}||^2||I - G_k||^2. \tag{39}$$

By the Corollary 3.3, it can be obtained:

$$||((p-1)I + G_k)^{-1}|| = \frac{1}{\sqrt{\lambda}}, \tag{40}$$

where λ is the smallest eigenvalue of $G = ((p-1)I + G_k)^*((p-1)I + G_k)$. It can be easily indicated that all of the eigenvalues of G are greater than $(p-1)^2$. Hence, it can be concluded that:

$$||I - G_{k+1}|| \leq \frac{p}{2(p-1)}||I - G_k||^2, \tag{41}$$

and for $k > 0$, we deduce

$$||I - G_{k+1}|| \leq (\frac{p}{2(p-1)})^k||I - G_0||^{2^k}. \tag{42}$$

Based on Lemma 3.2, it is concluded that

$$\rho(I - G_{k+1}) \leq ||I - G_{k+1}|| \leq (\frac{p}{2(p-1)})^k||I - G_0||^{2^k}, \tag{43}$$

therefore,

$$\rho(\prod_{k=i}^{n}(I-G_k)) \leq \prod_{k=i}^{n}\rho(I-G_k) \leq (\frac{p}{2(p-1)})^{\frac{(n-i)(n+i-1)}{2}+i-1}||I-G_0||^{2^n-2^{i-1}}. \tag{44}$$

If $||I-G_0|| < 1$, then $\rho(\prod_{k=i}^{n}(I-G_k)) < 1$ for every $n \geq 0$ and $i \geq 0$. In addition, it is concluded that $\{G_k\}$ converges quadratically to the matrix I.

□

Corollary 3.5. *Let $\{\tilde{\mathcal{X}}_k\}$ be the interval matrix sequence that generated by (30). If the hypotheses of Theorem 3.4 hold, then the sequence $\{m(\tilde{\mathcal{X}}_k)\}$ converges to $A^{\frac{1}{p}}$.*

Proof. Based on the 3th part of Lemma 3.1, when G_k converges to the identity matrix, the sequence $\{m(\tilde{\mathcal{X}}_k)^p\}$ converges to the matrix A. Using the Corollary 1.41 of [7] and an ordering of eigenvalues of G_k and $m(\tilde{\mathcal{X}}_k)$ $(m(\tilde{\mathcal{X}}_k) = m(\mathcal{X}_k))$, we have

$$\lambda_i(m(\mathcal{X}_{k+1})) = \lambda_i(m(\mathcal{X}_k))\frac{(p-1) + \lambda_i(G_k)}{p}. \tag{45}$$

Hence, by induction on k, because $m(\mathcal{X}_0)$ and G_0 have positive eigenvalues, $m(\mathcal{X}_k)$ has also positive eigenvalues for every k. Hence, by considering $m(\mathcal{X}_k)$ as the pth root of the matrix $m(\mathcal{X}_k)^p$, the sequence $\{m(\mathcal{X}_k)\}$ converges to the matrix $A^{\frac{1}{p}}$.

□

Note 1. If the hypotheses of the Theorem 3.4 hold, the matrix $((p-1)I + G_k)^p$ is well-condition for each $p = 2, 3, \cdots$.

Proof. Using relation (40) and (44), a bound for condition number of the matrix $((p-1)I+G_k)^p$ is calculated as follows:

$$\begin{aligned}\text{cond}(((p-1)I+G_k)^p) =& ||((p-1)I+G_k)^p||||(((p-1)I+G_k)^{-1})^p|| \\ \leq& ||(p-1)I+G_k||^p||((p-1)I+G_k)^{-1}||^p \\ \leq& ||pI+(I-G_k)||^p(\frac{1}{p-1})^p \\ \leq& (\frac{p+1}{p-1})^p.\end{aligned} \tag{46}$$

A simple computation shows that the sequence $(\frac{p+1}{p-1})^p$ for $p=2,3,\cdots$ is a decreasing sequence that converges to e^2. Therefore, the matrices $((p-1)I+G_k)^p$ are well-condition matrices. □

Note 2. Using the second step of the method (30), the norm of the point matrix G_{k+1} can be estimated as follows:

$$||G_{k+1}|| \leq p^p||((p-1)I+G_k)^{-1}||^p||G_k||.$$

By the convergence G_k to I, there is a natural number N such that for $k>N$, it can be concluded:

$$||G_{k+1}|| \leq ||G_k||.$$

3.1 Interval convergence

For proving the next theorem, the following relations are defined:

$$C_k = \frac{I-G_k}{p}, \quad f(\tilde{\mathcal{X}}_k) = m(\tilde{\mathcal{X}}_k) - \tilde{\mathcal{X}}_k C_k, \quad \tilde{\mathcal{X}}_{k+1} = f(\tilde{\mathcal{X}}_k) + U_{\epsilon_{k+1}}(0), \tag{47}$$

based on the relation (28) and the interval method (30). Since the midpoint map and multiplication of matrices are continuous, it is clear that the function f is continuous. Also, epsilon inflation method needs to continuous functions [15]. Thus, the following theorem can be explained.

Theorem 3.6. *Let $\{\tilde{\mathcal{X}}_k\}$ be an interval sequence generating by the interval method (30), and $m(\mathcal{X}_0)=I$, where I represents the identity matrix. Also, let A be a square matrix without any non-positive eigenvalues, $||I-A||<1$, $\epsilon_{k+1}<\frac{\epsilon_k}{2}$, and $U_{\epsilon_{k+1}}(0) \subset U_{\epsilon_k}(0)$. Thus, for some $t \geq 0$, there is an inclusion for the interval sequence $\{\tilde{\mathcal{X}}_k\}$ when $k \geq t$, that is*

$$\tilde{\mathcal{X}}_t \supset \tilde{\mathcal{X}}_{t+1} \supset \tilde{\mathcal{X}}_{t+2} \supset \ldots. \tag{48}$$

Proof. Based on the relations (47), for $k = 1$, it is clear that the following relation is true:

$$\tilde{\mathcal{X}}_k = m(\tilde{\mathcal{X}}_{k-1}) + U_{\epsilon_k}(0) + \sum_{i=0}^{k-2}(-1)^{i+1}(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{k-1} C_j + (-1)^k \tilde{\mathcal{X}}_0 \prod_{j=0}^{k-1} C_j. \tag{49}$$

Based on the induction on k, it is obtained that

$$\begin{aligned}\tilde{\mathcal{X}}_{k+1} = m(\tilde{\mathcal{X}}_k) - \Bigg\{ & m(\tilde{\mathcal{X}}_{k-1}) + U_{\epsilon_k}(0) + \sum_{i=0}^{k-2}(-1)^{i+1}\left(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)\right) \prod_{j=i+1}^{k-1} C_j \\ & + (-1)^k \tilde{\mathcal{X}}_0 \prod_{j=0}^{k-1} C_j \Bigg\} C_k + U_{\epsilon_{k+1}}(0).\end{aligned} \tag{50}$$

or,

$$\tilde{\mathcal{X}}_{k+1} = m(\tilde{\mathcal{X}}_k) + U_{\epsilon_{k+1}}(0) + \sum_{i=0}^{k-1}(-1)^{i}(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{k} C_j + (-1)^{k+1} \tilde{\mathcal{X}}_0 \prod_{j=0}^{k} C_j. \tag{51}$$

According to the method (2.5), the sequence $\{m(\tilde{\mathcal{X}}_k)\}$ is a convergent sequence, and based on the relation (44), $\prod_{i=j}^{k} C_i$ is a contracting matrix for each k and $0 \leq i \leq k$, hence there is $t \geq 0$ such that

$$\begin{aligned} & m(\tilde{\mathcal{X}}_t) - m(\tilde{\mathcal{X}}_{t-1}) + (-1)^{t-1}(m(\tilde{\mathcal{X}}_{t-1}) + U_{\epsilon_t}(0))C_t \\ & \quad + \sum_{i=0}^{t-2}(-1)^i\{(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{t} C_j + (m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=0}^{t-1} C_j\} \\ & \quad + (-1)^{t+1}(\tilde{\mathcal{X}}_0 \prod_{j=i+1}^{t} C_j + \tilde{\mathcal{X}}_0 \prod_{j=0}^{t-1} C_j) \\ & \quad \subseteq U_{\epsilon_{t+1}}(0), \end{aligned} \tag{52}$$

because $U_{\epsilon_{t+1}}(0)$ is a neighborhood of zero. Then, following S. M. Rump's result in [15] with $\epsilon_{t+1} < \frac{\epsilon_t}{2}$, and applying the general set inclusion principle (that for

any sets A, B, C, the relation $A - B \subseteq C$ implies that $A \subseteq B + C$), we obtain:

$$\begin{aligned} &U_{\epsilon_{t+1}}(0) + m(\tilde{\mathcal{X}}_t) + \sum_{i=0}^{t-1}(-1)^i(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{t} C_j + (-1)^{t+1}\tilde{\mathcal{X}}_0 \prod_{j=0}^{t} C_j \\ &\subseteq U_{\epsilon_{t+1}}(0) + U_{\epsilon_{t+1}}(0) + m(\tilde{\mathcal{X}}_{t-1}) + \sum_{i=0}^{t-2}(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{t-1} C_j + (-1)^{t}\tilde{\mathcal{X}}_0 \prod_{j=0}^{t-1} C_j \\ &\subset U_{\epsilon_t}(0) + m(\tilde{\mathcal{X}}_{t-1}) + \sum_{i=0}^{t-2}(m(\tilde{\mathcal{X}}_i) + U_{\epsilon_{i+1}}(0)) \prod_{j=i+1}^{t-1} C_j + (-1)^{t}\tilde{\mathcal{X}}_0 \prod_{j=0}^{t-1} C_j, \end{aligned} \tag{53}$$

or

$$\tilde{\mathcal{X}}_{t+1} \subset \tilde{\mathcal{X}}_t. \tag{54}$$

It is clear that this result can be true for every $k \geq t$. Thus, there is an inclusion for the interval sequence $\{\tilde{\mathcal{X}}_k\}$ when $k \geq t$. □

Lemma 3.7. *If the hypotheses of Theorem 3.6 hold, the interval sequence* $\{\tilde{\mathcal{X}}_k\}$ *converges to* $A^{\frac{1}{p}}$ *with order of convergence two.*

Proof. As there is an inclusion for the interval sequence $\{\tilde{\mathcal{X}}_k\}$ when $k \geq t$, then it converges to $\bigcap_{k=t}^{\infty} \tilde{\mathcal{X}}_k$ by Corollary 8.10 of [17]. Also, $m(\tilde{\mathcal{X}}_k)$ converges to $A^{\frac{1}{p}}$ by Corollary 3.5. Thus, there is $N > 0$ such that for $k \geq N$ and arbitrary $\epsilon_k \leq \frac{d(\tilde{\mathcal{X}}_k)}{2}$, the following estimate holds:

$$A^{\frac{1}{p}} \in m(\tilde{\mathcal{X}}_k) + (-\epsilon_k,\, \epsilon_k) \subseteq m(\tilde{\mathcal{X}}_k) + (-d(\tilde{\mathcal{X}}_k)/2,\, d(\tilde{\mathcal{X}}_k)/2).$$

So, for $k \geq N$ the matrix $A^{\frac{1}{p}}$ belongs to interval matrix $\tilde{\mathcal{X}}_k$. For t that obtained above, if $t \geq N$, then $A^{\frac{1}{p}} \in \tilde{\mathcal{X}}_t$. Also, if $N \geq t$, then $\tilde{\mathcal{X}}_t \supseteq \tilde{\mathcal{X}}_N \ni A^{\frac{1}{p}}$. Therefore, the iterate $\tilde{\mathcal{X}}_t$, which includes $A^{\frac{1}{p}}$, is easily achieved by examining the inclusion relationship. On the other hand, because

$$\tilde{\mathcal{X}}_t \supseteq \tilde{\mathcal{X}}_{t+1} \supseteq \cdots \supseteq \tilde{\mathcal{X}}_{N-1} \supseteq \tilde{\mathcal{X}}_N \ni A^{\frac{1}{p}},$$

the relation $A^{\frac{1}{p}} \in \bigcap_{k=t}^{\infty} \tilde{\mathcal{X}}_k$ is concluded, and so later iterates can be considered as more rigorous iterate including the matrix $A^{\frac{1}{p}}$.
Based on the first step of method (30) and relation (6) of Definition 2.3, the width of the $\mathcal{X}_{k+1}$ is computed as follows:

$$d(\tilde{\mathcal{X}}_{k+1}) = \frac{d(\tilde{\mathcal{X}}_k)}{p}|I - G_k| + 2\epsilon_{k+1}. \tag{55}$$

If Theorem 3.4 and Lemma 2.7 are used, it can be deduced that the sequence $\{d(\mathcal{X}_k)\}$ converges to the zero matrix. Also, for every $k \geq t$

$$d(\bigcap_{k=t}^{\infty} \tilde{\mathcal{X}}_k) \leq d(\tilde{\mathcal{X}}_k). \tag{56}$$

Hence, $d(\bigcap_{k=t}^{\infty} \tilde{\mathcal{X}}_k) = 0$ and $\bigcap_{k=t}^{\infty} \tilde{\mathcal{X}}_k = A^{\frac{1}{p}}$. So, it is proved that the sequence $\{\tilde{\mathcal{X}}_k\}$ converges to $A^{\frac{1}{p}}$.

For obtaining the order of convergence, relation (55) and Lemma 3.1 are used. Moreover, $A^{\frac{1}{p}} \in \tilde{\mathcal{X}}_k$ implies that $A \in \tilde{\mathcal{X}}_k^p$. Hence, by 7th part of Definition 2.3 and Corollary 3.5, for $k \geq t$

$$\begin{aligned}
||d(\tilde{\mathcal{X}}_{k+1})|| &= \frac{||d(\tilde{\mathcal{X}}_k)||}{p} |||I - m(\tilde{\mathcal{X}}_k)^{-p} A||| + 2\epsilon_{k+1} \\
&\leq \frac{||d(\tilde{\mathcal{X}}_k)||}{p} |||m(\tilde{\mathcal{X}}_k)^{-p}|||||m(\tilde{\mathcal{X}}_k)^p - A|| + 2\frac{\epsilon_k}{\mu} \\
&\leq \frac{||d(\tilde{\mathcal{X}}_k)||}{p} |||m(\tilde{\mathcal{X}}_k)^{-p}|||||m(\tilde{\mathcal{X}}_k)^p - \tilde{\mathcal{X}}_k^p|| + 2\frac{\epsilon_1}{\mu^k}, \qquad (\epsilon_1 < 0.5) \\
&\leq \frac{||d(\tilde{\mathcal{X}}_k)||}{p} |||m(\tilde{\mathcal{X}}_k)^{-p}|||||d(\tilde{\mathcal{X}}_k^p)|| + \frac{1}{\mu^k} \\
&\leq \frac{||d(\tilde{\mathcal{X}}_k)||}{p} |||(A^{\frac{1}{p}} + \epsilon)^{-p}|||||(|\tilde{\mathcal{X}}_k| + |\tilde{\mathcal{X}}_k|^2 + \cdots + |\tilde{\mathcal{X}}_k|^{p-1}|)d(\tilde{\mathcal{X}}_k)|| + \frac{1}{\mu^k}.
\end{aligned} \tag{57}$$

Because $k > t$ and $A^{\frac{1}{p}} \in \tilde{\mathcal{X}}_k$, we have $|\tilde{\mathcal{X}}_k| < A^{\frac{1}{p}} + d(\tilde{\mathcal{X}}_k)/2 < A^{\frac{1}{p}} + \epsilon$. Also, we set $C = \frac{1}{p}|||(A^{\frac{1}{p}} + \epsilon)^{-p}|||||((A^{\frac{1}{p}} + \epsilon) + (A^{\frac{1}{p}} + \epsilon)^2 + \cdots + (A^{\frac{1}{p}} + \epsilon)^{p-1}|||$. As k or μ gets larger, $\frac{1}{\mu^k}$ can be ignored, and

$$||d(\tilde{\mathcal{X}}_{k+1})|| \leq C||d(\tilde{\mathcal{X}}_k)||^2, \quad k > t. \tag{58}$$

If $||.||$ is the Euclidean norm, and $||.||'$ is another norm for matrix B,

$$||B||'\gamma_1 \leq ||B|| \leq ||B||'\gamma_2,$$

where γ_1 and γ_2 are positive constants, then

$$||d(\tilde{\mathcal{X}}_{k+1})||'\gamma_1 \leq C\gamma_2||d(\tilde{\mathcal{X}}_k)||'^2, \quad k > t. \tag{59}$$

Thus, by a Theorem of appendix [17], this study concludes that the order of convergence is almost 2. □

3.2 Well-behavior of iterations of the midpoint sequence of the interval method (30)

Definition 3.8 (well-behaved iterative method)**.** An iterative method (5) will be called well-behaved if there is the sequences $\{\delta X_k\}$ and $\{\delta d_k\}$ such that

$$\overline{\lim_k}||F(m(\mathcal{X}_k)+\delta X_k; A+\delta d_k)|| = O(\varrho^2), \tag{60}$$

and

$$||\delta X_k|| \leq k_1\varrho||m(\mathcal{X}_k)||, \quad ||\delta d_k|| \leq k_2\varrho||A||, \tag{61}$$

for large k, where k_1 and k_2 are constants, and can only depend on the dimension of the matrix A while ϱ is the relative computer precision.

Woźniakowski [23, 24] demonstrated that if an iterative method exhibits well-behaved characteristics, then it is numerically stable; however, the converse does not generally hold. Higham discussed the numerical stability of the matrix iterative method (2) in Definition 4.17 of [7]. In contrast, the present study establishes the well-behaved nature of the matrix iterative method (5) (or (2)), from which its numerical stability is subsequently inferred. In other words, the numerical stability achieved in this study is a consequence of the proposed method itself, and the approach is entirely distinct from and independent of Higham's framework.
Now, the equation $F(X;A) = X^p - A = 0$ is considered. Let A be a square matrix with eigenvalues on the right hand side of imaginary axis, and p be an integer number where $p \geq 2$. Let the matrix sequence $m(\mathcal{X}_k)$ be computed by the iterative method (5), and $m(\mathcal{X}_k)$ be successive approximations of $A^{\frac{1}{p}}$.

Theorem 3.9. *Let $\{m(\mathcal{X}_k)\}$ be the matrix sequence generated by the method (5). The iterative method (5) for the midpoints of the interval sequence $\{\tilde{\mathcal{X}}_k\}$ is a well-behaved iterative method, and therefore, is a numerical stable iterative method.*

Proof. Based on the Definition 3.8, this study should identify $\{\delta X_k\}$ and $\{\delta d_k\}$. This study defines $\delta X_k = \dfrac{d(\tilde{\mathcal{X}}_k)}{2}$ and $\delta d_k = p\varrho\dfrac{||G_k||}{2}$ where $d(\tilde{\mathcal{X}}_k)$ and $||G_k||$ are identified as before. Using the convergence the matrix sequence $\{d(\tilde{\mathcal{X}}_k)\}$ to zero matrix, there exists a natural number N for $\epsilon = \varrho||m(\mathcal{X}_k)||$ such that

$$||d(\tilde{\mathcal{X}}_k)|| < \varrho||m(\mathcal{X}_k)||, \quad k > N. \tag{62}$$

For matrix B, if we suppose $||B|| = \max_{i,j}|B_{ij}|$ and $k_1 = \frac{1}{2}$, then it can be obtained for $k > N$

$$||\delta X_k|| \leq k_1\varrho||m(\mathcal{X}_k)||. \tag{63}$$

Also, using Note 2 and $k_2 = \frac{p}{2}$, it can be deduced that:

$$||\delta d_k|| \leq k_2\varrho||G_0|| = k_2\varrho||A||. \tag{64}$$

Now, $F(m(\mathcal{X}_k)+\delta X_k; A+\delta d_k)$ can be computed as follows:

$$F(m(\mathcal{X}_k)+\delta X_k; A+\delta d_k) = (m(\mathcal{X}_k)+\delta X_k)^p - (A+\delta d_k). \tag{65}$$

For large k, using (3.6), $A^{\frac{1}{p}}$ belongs to the interval $\mathcal{X}_k$, and it can be deduced:

$$\begin{aligned} ||(m(\mathcal{X}_k)+\delta X_k)^p|| &\leq ||(m(\mathcal{X}_k)+\delta X_k)||^p \\ &\leq (||m(\mathcal{X}_k)|| + ||\delta X_k||)^p \\ &= ||m(\mathcal{X}_k)||^p + p||m(\mathcal{X}_k)||^{p-1}k_1\varrho||m(\mathcal{X}_k)|| + O(\varrho^2) \\ &= ||A|| + \varrho\frac{p}{2}||A|| + O(\varrho^2). \end{aligned} \tag{66}$$

Hence, by converging the interval sequence $\{\mathcal{X}_k\}$, it can be obtained:

$$\overline{\lim_k}||F(m(\mathcal{X}_k)+\delta X_k; A+\delta d_k)|| = \overline{\lim_k}(||(m(\mathcal{X}_k)+\delta X_k)^p|| - |||(A+\delta d_k)|||) = O(\varrho^2). \tag{67}$$

Thus, by Definition 3.8, the successive iterations of the sequence $\{m(\mathcal{X}_k)\}$ are well-behaved and based on Woźniakowski [23] is numerical stable. □

4. Numerical examples

This section justifies the validity of the obtained theoretical results by some numerical examples. In these examples, the pth root of a given matrix A is computed for various values of p using a Mathematica implementation referred to as pAlgorithm, which is provided in the Appendix. The calculated roots are the principal pth root. To this end, this study shows the eigenvalues of $A^{\frac{1}{p}}$ are in the segment $\{z \in \mathbb{C}; -\frac{\pi}{p} < \arg(z) < \frac{\pi}{p}\}$. In addition, the approximate computational order of convergence (ACOC) is calculated by this formula (see [25–27]):

$$ACOC = \frac{\ln(\frac{||X_{k+1}-X_k||}{||X_k-X_{k-1}||})}{\ln(\frac{||X_k-X_{k-1}||}{||X_{k-1}-X_{k-2}||})}. \tag{68}$$

Example 4.1. Consider the following matrix

```
A_n = Table[0.3/(i - j + 0.3), {i, 1, n}, {j, 1, n}],
```

which has the dimension n. The pth root of the matrix A_n is computed by the Mathematica code, pAlgorithm, for arbitrary p and n such as $p = 3, 5, 18, 2000, 2763, 3560$ and $n = 15, 40, 78, 500, 600, 1000$. The eigenvalues of A_n are in the right half-plane, and $||I - A_n||_2$ is computed. The Mathematica routine inverse, which is invoked in pAlgorithm can generate the inverse of well-condition matrices to the maximum precision given the input. Using Note 1, the matrix $((p-1)I + G_k)^p$ is well-condition for each $p = 2, 3, \cdots$.
Also, μ and δ which were introduced in Lemma 2.7 can be arbitrary considered 2 or greater than 2. The initial matrix $\mathcal{X}_0$ considered as follows:

```
Table[Interval[{IdentityMatrix[n][[i, j]]-u/2,IdentityMatrix[n][[i,j]]
+u/2}], {i, 1,Length[A_n]}, {j, 1, Length[A_n]}];
```

The arbitrary value u is considered the width of $\mathcal{X}_0$, and t the smallest iteration that $\tilde{\mathcal{X}}_{t+1} \subset \tilde{\mathcal{X}}_t$ is true. In Table 1, the error of computation and ACOC are also shown.The tolerance1 and tolerance2 are selected 10^{-10} and 10^{-5} arbitrarily. In formula (68), for the computation of $ACOC_d$, X_k is considered $d(\tilde{\mathcal{X}}_k)$ and for $ACOC_m$, X_k is considered $m(\tilde{\mathcal{X}}_k)$.

Table 1: Results for the computation of pth root of matrix A_n.

n	p	u	t	δ	k	ACOC_d	ACOC_m	$\|\|m(\mathcal{X}_k)^p - A\|\|_2$	$\|\|d(\mathcal{X}_k)\|\|_2$	ϵ_k	$\|\|I - A_n\|\|_2$	M
15	3	2	0	7	5	1.9954	2.0049	6.2506e-18	3.5740e-9	2.7269e-11	0.85992	50
15	3560	2	0	5	5	1.8782	2.0043	2.92042e-12	2.99579e-8	9.98596e-10	0.85991	50
40	5	10	0	5	5	1.5472	2.0009	5.2321e-14	8.0003e-8	9.99983e-10	0.93623	70
40	2000	3	3	5	5	1.5032	2.0008	6.71925e-11	7.99294e-8	9.99118e-10	0.93621	70
78	18	5	0	7	5	1.5939	2.0168	2.9296e-11	9.7494e-7	6.2496e-9	0.96190	60
500	3	2	0	9	6	1.6124	2.0015	4.9144e-13	3.1482e-6	5.7431e-11	0.98789	50
600	2763	5	2	9	6	1.6018	2.0010	1.4738e-13	5.2713e-6	4.3934e-11	0.98881	50
1000	5	2	3	9	8	1.5238	2.0061	5.3072e-12	2.1536e-6	3.5674e-11	0.99068	60

Table 1 presents the numerical results obtained for the computation of the p-th root of matrix A_n under various choices of parameters n, p, and u. This table reports key quantities such as the number of iterations k, contraction index t, error measures, and average computational order of convergence (ACOC), using both the diameter and midpoint of the computed interval enclosures. The following observations are made from the results:

- **Effect of matrix size n:** As the size of the matrix increases (e.g., from $n = 15$ to $n = 1000$), the method maintains a high level of accuracy, with the normed error $\|m(\tilde{\mathcal{X}}_k)^p - A\|_2$ remaining within acceptable limits (e.g., 10^{-12} or smaller). Notably, the values of ACOC_m remain close to 2 across different sizes, indicating consistent second-order convergence regardless of matrix dimension.

- **Effect of root order p:** In some cases, extremely high values of p were selected (e.g., $p = 2000$ and $p = 2763$). Despite this, both the convergence behavior and error metrics remain stable. For instance, the pair $n = 40$, $p = 5$ and $n = 40$, $p = 2000$ yield very similar errors and convergence orders, demonstrating that the method is robust with respect to variations in the root order p.

- **Behavior of the contraction index t:** In most entries, the contraction property $\tilde{\mathcal{X}}_{t+1} \subset \tilde{\mathcal{X}}_t$ is achieved at $t = 0$, suggesting rapid entry into the convergence regime. Only in a few cases, such as when $n = 40$, $p = 2000$ or $n = 1000$, $p = 5$, does t increase to 2 or 3. This indicates that for large matrices or high root orders, the onset of contraction may be slightly delayed, although convergence is still achieved reliably.

- **Control of numerical accuracy:** The computed errors ϵ_k are consistently smaller than the prescribed tolerance (10^{-5}), with many results showing errors as low as 10^{-11} or better. The number of decimal digits δ used in directed rounding varies between 5 and 9, suggesting a moderate precision requirement that scales with the problem size.

- **Role of initial width u:** The parameter u, representing the width of the initial set $\mathcal{X}_0$, varies between 2 and 10. Larger values of u sometimes lead to a slight increase in the iteration count k (e.g., $u = 10$ and $k = 8$), though the overall convergence behavior remains unaffected. This indicates that the method is not highly sensitive to the choice of u, provided it is chosen reasonably.

In conclusion, the method demonstrates robust and efficient convergence behavior across a wide range of matrix sizes and root orders. The consistent values of ACOC_m and low error norms support the reliability of the algorithm. These results highlight the method's potential for solving high-dimensional matrix root problems with controlled accuracy and predictable convergence properties.

4.1 Spectral symmetry in the principal matrix root

According to Theorem 1.1, the principal pth root of a matrix A is the unique pth root whose eigenvalues lie in the open sector $(-\pi/p, \pi/p)$ and is defined as a principal matrix function. This spectral constraint induces a characteristic symmetry in the eigenvalue arguments of real matrices.

In Table 2, it is observed that the maximum and minimum arguments of the eigenvalues of $A_n^{1/p}$ are equal in magnitude but opposite in sign. This behavior is both correct and theoretically expected. It arises from the conjugate symmetry property of complex eigenvalues in real matrices. Specifically, if A_n is a real matrix and $\lambda = re^{i\theta}$ is a complex eigenvalue, then its complex conjugate $\bar{\lambda} = re^{-i\theta}$ is also an eigenvalue. As a result, the eigenvalues of $A_n^{1/p}$ become $r^{1/p}e^{i\theta/p}$ and $r^{1/p}e^{-i\theta/p}$, respectively, reflecting symmetry of arguments about the real axis.

Hence, the values reported in Table 2, where the maximum and minimum arguments differ only in sign, are accurate and confirm the correctness of the numerical procedure. This symmetry is a direct consequence of the structure of real matrices and aligns with the theoretical behavior of the principal matrix pth root.

Example 4.2. (Ill-Conditioned Symmetric Matrix.) We consider the following matrix:

$$B = \frac{1}{30}\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}. \tag{69}$$

Table 2: Maximum and Minimum of the Arguments of eigenvalues of the pth root of the matrix A_n.

n	p	Maximum	Minimum	$\frac{\pi}{p}$
15	3	0.26777	-0.26777	1.0472
15	3560	0.000225	-0.000225	0.000882
40	5	0.176521	-0.176521	0.62831
40	2000	0.00044	-0.00044	0.00157
78	18	0.05051	-0.05051	0.17453
500	3	0.312106	-0.312106	1.0472
600	2763	0.000339	-0.000339	0.001137
1000	5	0.187844	-0.187844	0.62831

Matrix B is a scaled version of the Wilson matrix, which is symmetric, positive definite, and moderately ill-conditioned with condition number 2984. We compute the pth root of matrix B for various values of p. The eigenvalues of B are:

```
{1.00962, 0.128602, 0.0281036, 0.000338335},
```

and $\|I - B\|_2 = 0.999662 < 1$, which ensures that the matrix is sufficiently close to the identity to satisfy the assumptions of the algorithm.

Within the Mathematica code referred to as pAlgorithm, the initial interval matrix $\mathcal{X}_0$ is initialized as:

```
Table[Interval[{IdentityMatrix[4][[i, j]] - u/2,
IdentityMatrix[4][[i, j]] + u/2}],{i, 1, 4}, {j, 1, 4}];
```

This allows us to initialize a neighborhood around the identity matrix, which facilitates convergence.

We compute the average computational order of convergence (ACOC) for the sequences $\{m(\tilde{\mathcal{X}}_k)\}$ and $\{d(\tilde{\mathcal{X}}_k)\}$, and the 2-norm of $d(\tilde{\mathcal{X}}_k)$ as presented in Table 3. The tolerance1 and tolerance2 are selected 10^{-10} and 10^{-5} arbitrarily. In addition, we examine the arguments of the eigenvalues of the computed pth root. Since the matrix B is symmetric positive definite, all its eigenvalues are real and positive. Therefore, all arguments of the eigenvalues of its pth roots are zero, confirming that the computed roots are indeed principal. The results in Table 3 highlight the effectiveness and robustness of the proposed method for computing the pth root of matrices, even for ill-conditioned cases such as matrix B. Key observations include:

- **Effect of increasing p:** As the root order p increases from 2 to 2560, we observe that the number of iterations k remains moderate (between 9 and

Table 3: Results for the computation of pth root of matrix B by k iterations ($i+\delta$ digits are used for rounding in ith iteration).

p	k	u	t	δ	$ACOC_d$	$ACOC_m$	$\|\|m(\tilde{\mathcal{X}}_k)^p - B\|\|_2$	$\|\|d(\tilde{\mathcal{X}}_k)\|\|_2$	ϵ_k	Timing	M
2	9	2	1	10	2.08	1.99	8.92075e-12	5.85220e-8	9.55832e-19	0.015625	30
8	14	2	14	10	1.96	2.00	0.0725e-48	7.51587e-23	9.39483e-24	0.046875	50
22	14	5	14	5	1.92	2.00	0.0085e-47	7.29347e-18	9.11684e-19	0.046875	50
560	15	2	14	20	1.80	2.00	2.52918e-37	7.30765e-34	9.13457e-35	0.0625	50
2560	14	2	14	10	1.76	2.00	0.0093e-43	6.76083e-23	8.45104e-24	0.046875	50

15), suggesting good scalability of the method. Additionally, the norm of the residual error $\|m(\tilde{\mathcal{X}}_k)^p - B\|_2$ stays at extremely low levels, confirming the effectiveness of the method for large p.

- **Role of initial interval width u:** The parameter u controls the width of the initial interval matrix $\mathcal{X}_0$. Smaller values are used in most cases to ensure that the initial guess is close to the identity, thus facilitating rapid convergence.

- **Convergence behavior:** The approximate computational orders of convergence ($ACOC_d$ and $ACOC_m$) remain consistently close to 2 for all tested values of p, confirming near quadratic convergence of the algorithm. This highlights the efficiency and robustness of the method even for large and ill-conditioned matrices.

- **Error metrics and stability:** The residual error $\|m(\tilde{\mathcal{X}}_k)^p - B\|_2$ achieves values as small as 10^{-48} or even 10^{-43}, and the diameter norm $\|d(\tilde{\mathcal{X}}_k)\|_2$ also remains small across all values of p. These metrics confirm both the accuracy of the computed root and the tightness of the interval enclosures, thus verifying the robustness of the interval arithmetic.

- **Tolerance and precision settings:** The tolerances `tolerance1` $= 10^{-10}$ and `tolerance2` $= 10^{-5}$ are fixed for all experiments. To compensate for increased complexity at higher p, the rounding parameter δ is adjusted dynamically. This trade-off between precision and computational cost ensures reliable convergence without excessive iterations or numerical instability.

- **Principal root verification:** For all tested values of p, the eigenvalues of the computed matrix roots are real and positive, with arguments equal to zero. This guarantees that the method consistently yields the principal pth root, as theoretically expected for symmetric positive definite matrices like B.

In summary, Table 3 demonstrates that the proposed interval-based iterative algorithm is capable of accurately computing the pth root of an ill-conditioned matrix

for a wide range of p, with controlled precision, consistent convergence, and minimal error. The input parameters u, δ, and the stopping tolerances play key roles in balancing convergence speed and numerical robustness.

4.2 Comparison with existing methods for matrix pth root computation

Among the test examples considered, we specifically select the symmetric positive definite matrix defined in the relation (69) (previously discussed in Example 4.2 with emphasis on its ill–conditioning) to further demonstrate the accuracy, efficiency, and generalizability of the proposed interval-based method through comparative analysis.

We first compute the principal square root $B^{1/2}$ using several classical methods discussed in Ref. [13], including Newton iteration, Denman–Beavers (DB) iteration, Product Denman–Beavers (PDB), and Newton–Schulz iteration. The results obtained from these methods are then compared with those of the proposed interval-based method.

It is worth noting that these classical techniques are specifically designed for computing square roots of matrices and are not directly applicable to general pth roots. Therefore, to evaluate the flexibility and scalability of our method, we extend the computations to higher-order arbitrary roots with $p = 22$, $p = 560$, and $p = 2560$. For this purpose, we implement a generalized Newton iteration as described in Equation (1), and compare its performance with our interval-based approach Table 4 presents the spectral norm of the residual error $\|X^p - B\|_2$, the number of iterations performed, and the precision level M (used in `SetAccuracy`) employed in each computation. Here, I denotes the 4×4 identity matrix. The

Table 4: Comparison of methods for computing $B^{1/p}$ for $p = 2, 22, 560, 2560$.

p	Initial Guess X_0	Method	Iterations	$\|X^p - B\|_2$	Timing	M
2	B	Newton [13]	9	8.92075e-12	0.015625	50
2	B	Denman–Beavers [13]	9	8.92075e-12	0.015625	50
2	B	Product Denman–Beavers [13]	9	8.92075e-12	0.046875	50
2	B	Newton–Schulz [13]	14	3.41231e-13	0.03125	50
22	I	Generalized Newton method (1)	12	8.21646e-16	0.046875	50
560	I	Generalized Newton method (1)	12	3.12008e-14	0.015625	50
2560	I	Generalized Newton method (1)	12	3.47158e-14	0.046875	50

comparison between Tables 3 and 4 highlights several advantages of the proposed interval-based method. In the special case of $p = 2$, i.e., computing the matrix square root, the residual error achieved by our interval approach is comparable to that of classical methods reported in [13]. However, unlike point-based algorithms that return a single approximate matrix, our method computes an interval matrix that rigorously encloses the exact root. This feature ensures controlled round-off error and enhances numerical reliability.

Importantly, this advantage is not limited to the case $p = 2$. For higher values such as $p = 22$, $p = 560$, and $p = 2560$, the same level of accuracy and reliability is maintained. By employing a generalized Newton iteration within an interval framework, we successfully compute the pth root for arbitrary $p > 1$ with moderate computational cost. As recorded in Table 3, the execution times of our method are comparable to those of non-interval methods and do not incur significant additional overhead.

In summary, the proposed method achieves both high accuracy and verified enclosures for matrix pth roots, while maintaining computational efficiency and controlling rounding errors–a combination that is difficult to achieve using classical floating-point approaches. These features make our method particularly well-suited for applications where both reliability and generality are critical, such as verified computing, control theory, and the numerical solution of matrix equations.

Note 3. [Application of matrix pth root in solving differential equations]

In this study, the matrix pth root plays a crucial role in solving certain initial value problems involving matrix functions. As recalled from the theorem by Davies and Higham [28], the solution to the initial value problem

$$\frac{dy}{dw} = \alpha(A - I)[w(A - I) + I]^{-1}y, \quad y(0) = b, \quad 0 \leq w \leq 1, \tag{70}$$

is given by

$$y(w) = [w(A - I) + I]^{\alpha}b,$$

with the unique solution $y(w) = [w(A - I) + I]^{\alpha}b$, where $y(1) = A^{\alpha}b$..

This problem arises in various applications, including control theory, signal processing, and systems governed by fractional dynamics, where matrix functions such as fractional powers and roots of matrices are fundamental.

Computing the matrix pth root $A^{1/p}$ accurately and efficiently is essential in these contexts because it enables direct evaluation of solutions involving fractional powers of matrices without resorting to expensive numerical integration or approximation methods. Moreover, such problems can often be stiff, especially for large matrices or when high accuracy is demanded. Interval iterative methods, as proposed in this work, provide guaranteed enclosures of the solution and enhance the reliability and robustness of computations by controlling rounding errors and uncertainties.

Therefore, the method developed here not only contributes to the theoretical understanding of matrix root computations but also has practical significance by improving the numerical stability and precision of solutions to matrix-function-based differential equations encountered in real-life applications.

Note 4. This study worked based on Wolfram Mathematica for numerical results. The computer specifications were Intel(R) Xeon(R), CPU E7-4870 2.40 GHz (2 processors), with 16 GB of RAM.

5. Conclusions

In this study, an inclusion interval iterative method was developed based on a stable iterative approach for computing the pth root of a given square matrix, inspired by the epsilon inflation technique. The proposed interval method effectively calculates enclosure solutions, thereby controlling and containing rounding errors. The method achieves second-order convergence, and importantly, does not require an initial matrix that contains the pth root of the matrix A. Additionally, this study demonstrated the robust performance of the midpoint interval iterative method by utilizing the properties of the interval method. Finally, several numerical examples were provided to validate the theoretical findings.

Conflicts of Interest. The author declares that she has no conflicts of interest regarding the publication of this article.

Acknowledgment. The author wish to express their deep appreciation to the anonymous reviewers for their insightful comments and constructive suggestions, which have significantly contributed to improving the clarity and quality of this work.

References

[1] R. E. Moore, R. B. Kearfott and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, 2009.

[2] A. H. Bhrawy, E. Tohidi and F. Soleymani, A new Bernoulli matrix method for solving high-order linear and nonlinear Fredholm integro-differential equations with piecewise intervals, *Appl. Math. Comput.* **219** (2012) 482 – 497, https://doi.org/10.1016/j.amc.2012.06.020.

[3] P. Bakhtiari, T. Lotfi, K. Mahdiani and F. Soleymani, Interval Ostrowski–type methods with guaranteed convergence, *Ann. Univ. Ferrara Sez. VII Sci. Mat.* **59** (2013) 221 – 234, https://doi.org/10.1007/s11565-012-0174-4.

[4] J. M. Snyder, Interval analysis for computer graphics, *ACM SIGGRAPH Comput. Graph* **26** (1992) 121 – 130, https://doi.org/10.1145/142920.134024.

[5] M. Berz and G. Hoffstätter, Computation and application of Taylor polynomials with interval remainder bounds, *Reliab. Comput.* **4** (1998) 83 – 97, https://doi.org/10.1023/A:1009958918582.

[6] S. P. Mudur and P. A. Koparkar, Interval methods for processing geometric objects, *IEEE Comput. Graph. Appl.* **4** (1984) 7 − 17, https://doi.org/10.1109/MCG.1984.275931.

[7] N. J. Higham, *Functions of matrices: Theory and computation*, **104**, SIAM, 2008.

[8] L. S. Shieh, Y. T. Tsay and C. T. Wang, Matrix sector functions and their applications to systems theory, *IEE Proc.* **131** (1984) 171 − 181, https://doi.org/10.1049/ip-d.1984.0029.

[9] J. A. Marrero, R. B. Taher, Y. El Khatabi and M. Rachidi, On explicit formulas of the principal matrix pth root by polynomial decompositions, *Appl. Math. Comput.* **242** (2014) 435 − 443, https://doi.org/10.1016/j.amc.2014.05.110.

[10] J. S.-H. Tsai, L. S. Shieh and R. E. Yates, Fast and stable algorithms for computing the principal nth root of a complex matrix and the matrix sector function, *Comput. Math. Appl.* **15** (1988) 903−913, https://doi.org/10.1016/0898-1221(88)90034-X.

[11] S. Miyajima, Fast verified computation for the matrix principal pth root, *J. Comput. Appl. Math.* **330** (2018) 276 − 288, https://doi.org/10.1016/j.cam.2017.08.018.

[12] M. Moccari and T. Lotfi, On a two-step optimal Steffensen-type method: Relaxed local and semi-local convergence analysis and dynamical stability, *J. Math. Anal. Appl.* **468** (2018) 240 − 269, https://doi.org/10.1016/j.jmaa.2018.08.007.

[13] M. Moccari, Computing the matrix square root: A problem-solving approach using Mathematica and Pólya's strategies, *AUT J. Math. Comput.* (2025) Articles in Press, https://doi.org/ 10.22060/AJMC.2025.23508.1262.

[14] B. Iannazzo, On the Newton method for the matrix pth root, *SIAM J. Matrix Anal. Appl.* **28** (2006) 503 − 523, https://doi.org/10.1137/050624790.

[15] S. M. Rump, New results on verified inclusions, in: W. L. Miranker, R. A. Toupin (eds)*Accurate Scientific Computations*, Springer, Berlin, Heidelberg, (1986), 31 − 69.

[16] S. M. Rump, On the solution of interval linear systems, *Computing* **47** (1992) 337 − 353, https://doi.org/10.1007/BF02320201.

[17] G. Alefeld and J. Herzberger, *Introduction to Interval Computation*, Academic Press, 2012.

[18] G. Alefeld and G. Mayer, Interval analysis: Theory and applications, *J. Comput. Appl. Math.* **121** (2000) 421 – 464, https://doi.org/10.1016/S0377-0427(00)00342-3.

[19] S. Singh and G. Panda, SVD enclosure of a class of interval matrices, *Inf. Sci.* **666** (2024) #120386, https://doi.org/10.1016/j.ins.2024.120386.

[20] H. Ratschek and J. Rokne, *Geometric Computations with Interval and New Robust Methods: Applications in Computer Graphics, GIS and Computational Geometry*, Elsevier, 2003.

[21] W. Rudin, *Real and Complex Analysis*, Tata McGraw-Hill, 2006.

[22] S. H. Friedberg, A. J. Insel and L. E. Spence, *Elementary Linear Algebra*, Prentice Hall, 1997.

[23] H. Woźniakowski, Numerical stability for solving nonlinear equations, *Numer. Math.* **27** (1976) 373 – 390, https://doi.org/10.1007/BF01399601.

[24] M. Moccari, Smoothing the absolute value equations by the componentwise analysis, *J. Math. Model.* **13** (2025) 251 – 262, https://doi.org/10.22124/JMM.2024.28343.2501.

[25] M. Grau-Sánchez, M. Noguera and J. M. Gutiérrez, On some computational orders of convergence, *Appl. Math. Lett.* **23** (2010) 472 – 478, https://doi.org/10.1016/j.aml.2009.12.006.

[26] M. Moccari and T. Lotfi, Using majorizing sequences for the semi-local convergence of a high-order and multipoint iterative method along with stability analysis, *J. Math. Ext.* **15** (2021) 1 – 32.

[27] M. Moccari, T. Lotfi and V. Torkashvand, On the stability of a two-step method for a fourth-degree family by computer designs along with applications, *Int. J. Nonlinear Anal. Appl.* **14** (2023) 261 – 282, https://doi.org/10.22075/IJNAA.2022.25849.3142.

[28] P. I. Davies and N. J. Higham, Computing $f(A)b$ for matrix functions f, in: *QCD and Numerical Analysis III*, Springer Berlin Heidelberg, 2005.

Appendix: pAlgorithm Code

pAlgorithm: Mathematica code for calculating the pth root of a given matrix

```
diameter[X_] :=
Table[(Abs[Max[X[[i, j]]] - Min[X[[i, j]]]]),
{i, 1, Length[X]}, {j,1,Length[X]}]
midpoint[X_] :=
Table[(Max[X[[i, j]]] + Min[X[[i, j]]])/2,
{i, 1, Length[X]}, {j, 1, Length[X]}]
down[x_, n_] := N[x - 10^-n (x*10^n - Floor[x*10^n]), n];
up[x_, n_] := N[x + 10^-n (Ceiling[x*10^n] - x*10^n), n];
M = Input["M"];
u = Input["u"];
delta = Input["delta"];
p = Input[pth_Root];
tolerance1 = Input["tolerance1"];
tolerance2 = Input["tolerance2"];
L = IdentityMatrix[Length[A]];
subset[Y_, Z_] :=
Table[If[Min[Z[[i, j]]] < Min[Y[[i, j]]] &&
Max[Y[[i, j]]] < Max[Z[[i, j]]],True, False],
{i, 1, Length[Z]}, {j, 1, Length[Y]}];
subsetTrue = Table[True, {i, 1, Length[A]}, {j, 1, Length[A]}];
Subscript[X, 0] =
Table[Interval[{L[[i, j]] - u/2, L[[i, j]] + u/2}], {i, 1,
Length[A]}, {j, 1, Length[A]}];
Subscript[X, -1] =
Table[Interval[{L[[i, j]] - u, L[[i, j]] + u}], {i, 1,
Length[A]}, {j, 1, Length[A]}];
Subscript[G, 0] = SetAccuracy[A, M];
k = 0;
Subscript[epsilon, 0] = 2;
diam = SetAccuracy[Norm[diameter[Subscript[X, k]], 2], M];
normProot = Norm[MatrixPower[midpoint[Subscript[X, k]], p] - A, 2];
If[subset[Subscript[X, k], Subscript[X, k - 1]] == subsetTrue,
subseting = True, subseting = False];

While[(normProot > tolerance1) || (diam > tolerance2) ||
(subseting ==False),
Subscript[a, k + 1] =
SetAccuracy[midpoint[Subscript[X, k]] -
1/p Subscript[X, k].(IdentityMatrix[Length[A]] - Subscript[G, k]),M];
Subscript[epsilon, k + 1] =
SetAccuracy[Min[Max[Table[Max[Min[Subscript[a, k + 1][[i, j]]]
-down[Min[Subscript[a, k + 1][[i, j]]], k+delta],
up[Max[Subscript[a, k + 1][[i, j]]], k+delta]-
Max[Subscript[a, k + 1][[i, j]]]], {i, 1,
```

```
Length[A]}, {j, 1, Length[A]}]], Subscript[epsilon, k]/2], M];
Subscript[X, k + 1] = SetAccuracy[
Subscript[a, k + 1] +
Table[Interval[{-Subscript[epsilon, k + 1], Subscript[epsilon, k + 1]}],
{i, 1, Length[A]}, {j, 1, Length[A]}], M];
Subscript[Y, k] =
SetAccuracy[((p - 1) IdentityMatrix[Length[A]] + Subscript[G, k])/p,M];
Subscript[G, k + 1] = SetAccuracy[
Inverse[MatrixPower[Subscript[Y, k], p]].Subscript[G, k], M]; k++;
normProot = Norm[MatrixPower[midpoint[Subscript[X, k]], p] - A, 2];
diam = SetAccuracy[Norm[diameter[Subscript[X, k]], 2], M];
If[subset[Subscript[X, k], Subscript[X, k - 1]] == subsetTrue,
subseting = True, subseting = False]]
```

Mandana Moccari
Department of Mathematics,
Ha. C., Islamic Azad University,
Hamedan, Iran
e-mail: m-moccari@iau.ac.ir