mathematics

MDPI

Article

Fourth-Order Iterative Algorithms for the Simultaneous Calculation of Matrix Square Roots and Their Inverses

Jiameihui Zhu [1], Yutong Li [1], Yilin Li [1], Tao Liu [1,*] and Qiang Ma [2]

[1] School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
[2] Department of Mathematics, Harbin Institute of Technology at Weihai, Weihai 264209, China
* Correspondence: liutao@neuq.edu.cn or math.taoliu@gmail.com

Abstract

This paper develops and analyzes new high-order iterative schemes for the effective evaluation of the matrix square root (MSR). By leveraging connections between the matrix sign function and the MSR, we design stable algorithms that exhibit fourth-order convergence under mild spectral conditions. Detailed error bounds and convergence analyses are provided, ensuring both theoretical rigor and numerical reliability. A comprehensive set of numerical experiments, conducted across structured and large-scale test matrices, demonstrates the superior performance of the proposed methods compared to classical approaches, both in terms of computational efficiency and accuracy. The results confirm that the proposed iterative strategies provide robust and scalable tools for practical applications requiring repeated computation of matrix square roots.

Keywords: matrix square root; iterative methods; convergence analysis; computational efficiency; Jordan canonical form

MSC: 65F60; 41A25

check for updates

Academic Editors: Juan Ramón Torregrosa Sánchez, Alicia Cordero and Francisco I. Chicharro

Received: 27 September 2025
Revised: 17 October 2025
Accepted: 21 October 2025
Published: 22 October 2025

Citation: Zhu, J.; Li, Y.; Li, Y.; Liu, T.; Ma, Q. Fourth-Order Iterative Algorithms for the Simultaneous Calculation of Matrix Square Roots and Their Inverses. *Mathematics* **2025**, *13*, 3370. https://doi.org/10.3390/math13213370

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introductory Notes

The study of matrix functions occupies a central role in modern computational mathematics and its applications across diverse scientific domains. A matrix function is defined as a rule that assigns to a square matrix $A \in \mathbb{C}^{n\times n}$ another matrix of the same dimension, based on some functional relation that extends scalar functions to the matrix setting. Classical definitions rely on the Jordan canonical form, the integral of Cauchy representation, or Hermite matrix polynomial expansions, each of which provides an equivalent but distinct perspective on how functions act on matrices [1]. These formulations are not merely of theoretical interest; rather, they constitute the essential building blocks for developing computational algorithms that can handle large-scale problems in engineering and data science.

Traditional techniques such as diagonalization or contour integration have long been used to evaluate matrix functions. However, their practical implementation is often limited by the need for spectral decompositions or by sensitivity to conditioning when dealing with non-normal or nearly defective matrices. In contrast, iterative schemes have gained increasing prominence [2], especially in large-scale or real-time applications where data evolves dynamically or when good-quality initial approximations are available [3,4]. Iterative methods improve computational tractability while offering flexible trade-offs between accuracy and efficiency.

The matrix square root (MSR) X of a provided matrix A is defined such that

$$X^2 = A.$$

In practical numerical applications, one may encounter the need to compute both X and its inverse X^{-1} simultaneously [5]. This coupled problem can be reformulated as finding a solution X that minimizes the residual

$$R(X) = X^2 - A,$$

with the corresponding cost function given by

$$\Phi(X) = \frac{1}{2}\|X^2 - A\|^2.$$

The MSR of A, shown by $A^{1/2}$, is any matrix X such that $X^2 = A$. Unless otherwise specified, $A^{1/2}$ refers to the principal square root. Note that $A^{-1/2}$ denotes the inverse of the principal square root, i.e., $(A^{1/2})^{-1}$; see also [6]. In this formulation, the goal is to drive the residual $R(X)$ to zero. Similarly to the nonlinear least-squares problem discussed earlier, a natural approach is to adopt a Newton-type iteration. However, matrices that are ill-conditioned or which exhibit clustered eigenvalues may impair the performance of the classical Newton or even Gauss–Newton iterations, thus necessitating higher-order corrections.

The present work focuses on the iterative computation of the MSR and its inverse. This problem is of considerable importance both theoretically and practically. For example, in the resolution of nonlinear and linear matrix differential equations, MSRs naturally arise in closed-form solutions. Consider the first-order system

$$\frac{d}{dt}u(t) = \sqrt{A}\,u(t) + v(t), \tag{1}$$

where A is a matrix with a well-defined square root and $v(t)$ is a prescribed source term. The solution can be expressed explicitly as

$$u(t) = \exp(t\sqrt{A})u(0) + \int_0^t \exp\big((t-\tau)\sqrt{A}\big)v(\tau)\,d\tau. \tag{2}$$

Here, the exponential of the MSR governs the dynamical evolution. Similarly, in second-order matrix differential Equations [1] of oscillatory type,

$$\begin{cases} w''(t) + Aw(t) = 0, \\ w(0) = w_0, \quad w'(0) = w_0', \end{cases} \tag{3}$$

the exact solution is given by

$$w(t) = \cos(\sqrt{A}t)w_0 + A^{-1/2}\sin(\sqrt{A}t)w_0', \tag{4}$$

whereas both the principal square root and its inverse play crucial roles. The MSR is tightly linked to a family of related matrix functions, such as the matrix exponential, logarithm, cosine, and sine. These interconnections provide both theoretical insight and computational pathways [7].

1.1. Exponential and Logarithm

For a nonsingular matrix A with no eigenvalues on the closed negative real axis, one can define [8]

$$A^{1/2} = \exp\Big(\tfrac{1}{2}\log(A)\Big), \tag{5}$$

wherein $\log(A)$ is the principal matrix logarithm. This relation emphasizes that efficient algorithms for the matrix logarithm can indirectly yield algorithms for the square root.

1.2. Trigonometric Functions

In oscillatory systems, one often encounters

$$\cos(\sqrt{A}t) = \tfrac{1}{2}\big(\exp(i\sqrt{A}t) + \exp(-i\sqrt{A}t)\big), \tag{6}$$

$$A^{-1/2}\sin(\sqrt{A}t) = \tfrac{1}{2i}A^{-1/2}\big(\exp(i\sqrt{A}t) - \exp(-i\sqrt{A}t)\big). \tag{7}$$

Here, the inverse square root $A^{-1/2}$ plays an indispensable role in the evaluation of the sine function, as in the resolution of second-order differential equations.

1.3. The Sign of a Matrix

The connection between the MSR and the sign function is fundamental. For a nonsingular A,

$$\operatorname{sign}(A) = A(A^2)^{-1/2}, \tag{8}$$

which shows that an efficient algorithm for the sign function can be adapted to calculate the square root, and vice versa.

1.4. Control Theory

In linear quadratic regulator (LQR) problems, the Riccati (algebraic) equation

$$C - XBX + A^\top X + XA = 0, \tag{9}$$

is central. Numerical solvers frequently exploit the Hamiltonian matrix

$$\mathcal{H} = \begin{bmatrix} A & -B \\ -C & -A^\top \end{bmatrix},$$

whose stable invariant subspace is determined via the matrix sign function. Since $\operatorname{sign}(\mathcal{H})$ computation depends on MSR algorithms, iterative schemes for the square root are vital in control applications [9].

1.5. Quantum Mechanics

The time evolution of a quantum system is controlled by the Schrödinger equation

$$i\hbar\frac{d}{dt}\psi(t) = H\psi(t),$$

with Hamiltonian H. When considering reduced models, one often computes

$$\exp(-it\sqrt{H}) \quad \text{or} \quad \cos(\sqrt{H}t),$$

for propagator approximations. Hence, square root operators are crucial for both exact and approximate quantum dynamics.

1.6. Data Science and Machine Learning

In covariance estimation, whitening transformations, and kernel methods, the MSR is indispensable. Given a positive semidefinite covariance matrix Σ,

$$\Sigma^{-1/2}X, \tag{10}$$

produces whitened data with identity covariance. Efficient computation of $\Sigma^{\pm 1/2}$ is therefore a fundamental preprocessing step in statistical learning. See also Table 1.

Table 1. Applications of the MSR and its inverse across disciplines.

Discipline	Problem	Role of MSR / Inverse
Control Theory	Algebraic Riccati equations	Computation of stable subspaces via sign function
Quantum Mechanics	Propagation of wave functions	Evaluation of $\exp(-it\sqrt{H})$, oscillatory propagators
Data Science	Whitening and PCA	Transformation $\Sigma^{-1/2}$ to normalize covariance
Finance	Monte Carlo simulations	Generate correlated samples $X = \Sigma^{1/2}Z$
Engineering	Structural dynamics	Oscillatory system solutions via $\cos(\sqrt{A}t)$, $\sin(\sqrt{A}t)$
Numerical PDEs	Fractional diffusion operators	Discretized forms require $(-\Delta)^{-1/2}$ or related roots

Direct methods for computing MSRs—such as Schur decomposition followed by block square rooting—are numerically reliable but often computationally prohibitive for large matrices. Iterative methods provide several advantages:

(a) Scalability: iterative schemes can handle very large matrices by exploiting sparsity and structure.
(b) Simultaneity: coupled iterations like Denman-Beavers [10] yield both $A^{1/2}$ and $A^{-1/2}$ simultaneously, saving computation.
(c) High-Order Convergence: fourth-order or higher iterations significantly reduce the number of matrix inversions required, improving efficiency.
(d) Connections to Root-Finding: iterations can be systematically designed by analogy with scalar nonlinear solvers, allowing one to import convergence theory from classical numerical analysis.

Recall that the sign of a matrix A with no eigenvalues on the imaginary axis is defined by the Cauchy integral formulation [1]

$$\operatorname{sign}(A) = \Omega = \frac{2}{\pi} A \int_0^\infty (l^2 I + A^2)^{-1} dl. \tag{11}$$

The matrix sign function satisfies $\Omega^2 = I$ but, unlike the identity, it encodes the spectral distribution of A relative to the imaginary axis. It thus acts as a bridge between nonlinear scalar equations, such as $g(x) = x^2 - 1 = 0$, and matrix iterations for the square root.

The central idea pursued here is that many seemingly distinct iterative matrix algorithms can be derived by adapting classical nonlinear root-finding methods—Newton-type, fixed-point, or Padé-based approaches—to the matrix context [11–13]. This perspective highlights the deep interplay between scalar root solvers and matrix function evaluations.

In this article, we introduce a novel higher-order iterative solver for simultaneously obtaining the MSR and its inverse. The proposed solver achieves a fourth-order rate of convergence, which improves computational efficiency while retaining robustness. Moreover, it establishes a direct connection with the matrix sign function, an object of central importance in numerical linear algebra. The proposed iteration is constructed by enhancing

a scalar root-finding model [14] with an additional correction phase that raises the convergence order from two to four, while preserving numerical stability. Its matrix formulation, developed later in Section 3, retains the commutativity between the matrix A and the iterates X_k, which ensures structural stability and facilitates convergence analysis. A rigorous proof of fourth-order convergence is provided for both the scalar and matrix cases, and the method exhibits wide empirical basins of attraction, indicating strong global behavior. In terms of computational complexity, the new solver achieves the same per-iteration cost as established Padé-type schemes but requires fewer iterations to reach the desired accuracy.

The rest of this manuscript is organized as follows. In Section 2, we review established iterative solvers to calculate the MSR and the sign function, emphasizing their convergence properties and limitations. Section 3 presents our proposed fourth-order algorithm with global convergence type for the simultaneous computation of the square root and its inverse. Section 4 provides a series of numerical experiments, demonstrating the efficiency and stability of the new solver in comparison with classical schemes. Finally, Section 5 summarizes the main findings and outlines directions for further works.

2. Existing Methods

The numerical computation of MSRs and sign functions has been addressed by a wide variety of iterative methods. These algorithms can be broadly categorized into Newton-type schemes, coupled iterative solvers, cyclic reduction methods, and Padé-based rational approximants. Each class of methods has advantages in terms of convergence order, numerical stability, and computational cost.

2.1. Newton-Type Methods

Employing Newton's scheme to the nonlinear scalar Equation $g(x) = x^2 - 1 = 0$ yields, in the matrix case, the famous iteration for the function of sign [15]:

$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-1}\right), \tag{12}$$

where the sequence $\{X_k\}$ tends quadratically to $\operatorname{sign}(A)$ provided X_0 is suitably chosen. Similarly, Newton's scheme for the MSR of A leads to

$$X_{k+1} = \frac{1}{2}\left(X_k + AX_k^{-1}\right). \tag{13}$$

While elegant and conceptually straightforward, iteration (13) can exhibit instability when the starting point X_0 is poorly selected or when the underlying matrix A has challenging spectral properties.

2.2. Denman–Beavers Iteration

To overcome numerical instabilities, Denman and Beavers proposed a coupled two-sequence iteration [10], given by

$$\begin{cases} Z_0 = I, \quad X_0 = A, \\ X_{k+1} = \frac{1}{2}\left(X_k + Z_k^{-1}\right), \\ Z_{k+1} = \frac{1}{2}\left(Z_k + X_k^{-1}\right). \end{cases} \tag{14}$$

Here, $\{X_k\}$ tends to the principal square root $A^{1/2}$, while $\{Z_k\}$ tends to its inverse $A^{-1/2}$. The simultaneous computation of both quantities makes the method particularly attractive for problems where both appear naturally, such as in oscillatory differential equations.

2.3. Cyclic Reduction

An alternative approach relies on cyclic reduction techniques, which originate from iterative methods for solving matrix Equations [16]. For the MSR, the scheme takes the form

$$\begin{cases} Z_0 = 2(A+I), \quad X_0 = -A+I, \\ X_{k+1} = -X_k Z_k^{-1} X_k, \\ Z_{k+1} = Z_k - 2X_k Z_k^{-1} X_k. \end{cases} \tag{15}$$

The sequence $\{Z_k\}$ is constructed to converge toward $4A^{1/2}$, from which the desired root is recovered after normalization. Although less commonly used than Newton-type methods, cyclic reduction can offer stability advantages in certain structured problems.

2.4. Padé-Based Methods

A major advance in the design of iterative algorithms for the matrix sign function was introduced by Kenney and Laub [17], who constructed optimal iterations from Padé approximants to the scalar function

$$h(\zeta) = \frac{1}{\sqrt{1-\zeta}}, \qquad \zeta = 1 - u^2.$$

This leads to the family of rational iterations

$$u_{k+1} = \frac{u_k P_{\kappa_1,\kappa_2}(1-u_k^2)}{Q_{\kappa_1,\kappa_2}(1-u_k^2)}, \tag{16}$$

where P and Q are the Padé numerator and denominator polynomials of order $[\kappa_1, \kappa_2]$. The convergence order is $1 + \kappa_1 + \kappa_2$, allowing systematic construction of high-order globally convergent iterations.

Of particular interest are the fourth-order globally convergent schemes:

$$X_{k+1} = \left(I + 6X_k^2 + X_k^4\right)\left(4X_k(I + X_k^2)\right)^{-1}, \tag{17}$$

$$X_{k+1} = \left(4X_k(I + X_k^2)\right)\left(I + 6X_k^2 + X_k^4\right)^{-1}, \tag{18}$$

corresponding to the Padé $[1,2]$ approximant and its reciprocal. These methods balance rapid convergence with numerical robustness, making them leading competitors among modern schemes [18]. See also Table 2.

Although several iterative algorithms have been developed for computing matrix square roots–most notably Newton-type schemes, the Denman–Beavers coupled iteration, and Padé-based rational approximants–each of these approaches exhibits limitations that restrict their efficiency in practice. Newton's method, while conceptually simple, is only quadratically convergent and highly sensitive to the choice of initial guess, often leading to instability for ill-conditioned matrices or those with clustered eigenvalues. The Denman–Beavers iteration achieves simultaneous computation of $A^{1/2}$ and $A^{-1/2}$, but its quadratic rate and requirement of two matrix inversions per iteration make it computationally expensive for large-scale problems. These limitations motivate the development of a new algorithm that combines higher-order convergence with simultaneous evaluation of $A^{1/2}$ and $A^{-1/2}$, while reducing the total computational burden. The present work fills this gap by constructing a rational, fourth-order iteration that maintains the advantages of Padé schemes but achieves lower iteration counts in both theoretical and numerical contexts.

Table 2. Convergence orders of representative iterative methods for the MSR and sign function.

Method	Iterates Computed	Order of Convergence	Notes
Newton Iteration	MSR only	Quadratic (second-order)	Sensitive to choice of X_0
Denman–Beavers	MSR and inverse	Quadratic (second-order)	Stable for wide range of A
Cyclic Reduction	MSR (scaled)	Quadratic	Requires normalization
Padé $[1,1]$	Sign function	Cubic (third-order)	Rational iteration
Padé $[1,2]$	Sign function	Fourth-order	Globally convergent
Proposed Method	MSR and inverse	Fourth-order	Combines efficiency and simultaneity

3. Construction of a Novel Solver

One of the primary challenges in computing matrix square roots arises when the matrix A is poorly conditioned. In practice, a poorly conditioned matrix will have a high condition number, which implies that small perturbations in A or numerical errors during computation can lead to large inaccuracies in the calculated square root X. Furthermore, when the eigenvalues of A are clustered closely together, the sensitivity of the MSR function increases. Under these circumstances, standard iterative techniques such as the basic Newton method may converge very slowly or might not converge at all due to the inability of a first- or second-order model to capture the local behavior of the function accurately.

Higher-order methods tend to require fewer iterations, even though each iteration may be computationally more involved. The reduction in the number of costly evaluations can more than compensate for the increased per-iteration cost. Hence, in situations where the cost of an individual evaluation is high, using a higher-order approach—such as the tensor–Newton method—can lead to a net gain in computational efficiency. Empirical results from tests on nonlinear least-squares problems indicate that although the tensor–Newton method involves a higher per-iteration cost, it succeeds in reducing the total number of function evaluations and yields a more robust performance profile.

Although fourth-order Padé-based schemes for the matrix sign and square root functions are well established, the new solver proposed in this work is motivated by a distinct theoretical and practical consideration. The present method is not obtained from Padé rational approximation but rather from a carefully constructed nonlinear scalar root-finding model whose correction terms yield a specific set of numerical coefficients. These coefficients preserve the same asymptotic convergence order and computational cost as the Padé $[1,2]$ schemes, yet they produce noticeably larger attraction basins in practice. As a consequence, the proposed iteration typically enters the final convergence phase one cycle earlier than its Padé counterparts, leading to higher practical efficiency without sacrificing numerical stability or simplicity. Therefore, the novelty lies not merely in reproducing a fourth-order behavior but in optimizing the iteration coefficients through analytical design, resulting in improved global convergence characteristics and faster empirical performance under identical computational effort.

3.1. Formulation of an Enhanced Iterative Scheme

To improve both the accuracy and the rate of convergence of Newton's method [19], an additional correction phase is introduced that incorporates a divided-difference approximation [20]. This refinement leads to the design of a more advanced algorithm tailored for solving the nonlinear equation $g(x) = 0$:

$$\begin{cases} y_k = x_k - \dfrac{g(x_k)}{g'(x_k)}, & x_0 \geq 0, \\ h_k = x_k - \dfrac{-5000g(x_k) + 5001g(y_k)}{-5000g(x_k) + 10001g(y_k)} \dfrac{g(x_k)}{g'(x_k)}, \\ x_{k+1} = h_k - \dfrac{g(h_k)}{g[h_k, x_k]}, \end{cases} \tag{19}$$

where the notation $g[a_1, a_2] := (a_1 - a_2)^{-1}\big(g(a_1) - g(a_2)\big)$ is adopted.

The iterative process described in (19) has been carefully designed to accomplish two key objectives. First, it represents an extension of Newton's classical iteration, providing a higher order of convergence. Second, and perhaps more importantly, the scheme is constructed in such a way as to guarantee global convergence properties, thereby ensuring its applicability to a much wider family of nonlinear matrix equations without compromising numerical robustness. The proposed procedure thus achieves a balanced compromise between efficiency and stability, rendering it an attractive tool for tackling nonlinear problems in matrix computations with both reliability and computational economy.

Theorem 1. *Let $g : D \subseteq \mathbb{C} \to \mathbb{C}$ be a function of class $C^4(D)$ (that is, four times continuously differentiable on D) or, equivalently, analytic in a neighborhood of a simple zero $\eta \in D$ such that $g(\eta) = 0$ and $g'(\eta) \neq 0$. If the initial point x_0 is chosen sufficiently close to η, then the sequence $\{x_k\}$ generated by iteration (19) converges to η with order four, that is,*

$$|x_{k+1} - \eta| \leq C_1 \, |x_k - \eta|^4,$$

for some constant $C_1 > 0$ and all k large enough.

Proof. Suppose η is a simple root for g and $e_k = x_k - \eta$ and $b_i = \dfrac{g^{(i)}(\eta)}{i!\, g'(\eta)}$, $1 \leq i \leq 4$. Assuming that g is adequately smooth, we expand both $g(x_k)$ and its derivative $g'(x_k)$ in a Taylor series about the point η, which leads to the following representations:

$$g(x_k) = g'(\eta)\Big[e_k + b_2 e_k^2 + b_3 e_k^3 + b_4 e_k^4 + O(e_k^5)\Big], \tag{20}$$

as well as

$$g'(x_k) = g'(\eta)\Big[1 + 2b_2 e_k + 3b_3 e_k^2 + 4b_4 e_k^3 + O(e_k^4)\Big]. \tag{21}$$

Employing the relations given in (20) and (21), the following formulation can be obtained:

$$\frac{g(x_k)}{g'(x_k)} = e_k - b_2 e_k^2 + 2(b_2^2 - b_3)e_k^3 + \Big(-4b_2^3 + 7b_2 b_3 - 3b_4\Big)e_k^4 + O(e_k^5). \tag{22}$$

By inserting Equation (22) into the definition of y_k from (19), the resulting expression becomes

$$y_k = \eta + b_2 e_k^2 + \Big(-2b_2^2 + 2b_3\Big)e_k^3 - \Big(-4b_2^3 + 7b_2 b_3 - 3b_4\Big)e_k^4 + O(e_k^5). \tag{23}$$

Utilizing Equations (20)–(23), we obtain

$$\frac{-5000g(x_k) + 5001g(y_k)}{-5000g(x_k) + 10001g(y_k)} = 1 + b_2 e_k + \left(2b_3 - \frac{4999b_2^2}{5000}\right)e_k^2$$

$$+\left(-\frac{9999b_2^3}{25000000}-\frac{2499b_3b_2}{1250}+3b_4\right)e_k^3+O(e_k^4). \tag{24}$$

Inserting (24) into (19) and applying a Taylor expansion followed by appropriate simplifications, we arrive at

$$z_k=\eta-\frac{1}{5000}\left(b_2^2e_k^3\right)+\frac{3\left(8338333b_2^3-8340000b_2b_3\right)}{25000000}e_k^4+O(e_k^5). \tag{25}$$

Proceeding with a Taylor series expansion of $g(z_k)$ around η and substituting Equation (19), one obtains

$$e_{k+1}=-\frac{b_2^3}{5000}e_k^4+\mathcal{O}(e_k^5). \tag{26}$$

This completes the proof, establishing fourth-order convergence of the proposed scheme. □

The scalar iteration presented in Equation (19) was derived from a modified Newton-type method that incorporates an additional correction phase. In particular, it follows the general principle of constructing high-order iterative processes through carefully chosen divided-difference approximations. The specific numerical coefficients in (19)—notably the constants 5000, 5001, and 10001—were obtained by enforcing algebraic cancelation of lower-order error terms in the Taylor expansion of the iterative map. This coefficient selection ensures that the third-order error components vanish, thereby achieving a fourth-order error term with minimal computational complexity. Unlike schemes derived by direct Padé approximation or ad hoc parameter tuning, the coefficients here arise analytically from satisfying the fourth-order convergence condition of the scalar prototype $g(x)=0$.

3.2. Extension to the Matrix Sign

By adapting the scalar scheme (19) to the matrix context, one arrives at a high-order solver for the matrix sign associated with $X^2=I$. The corresponding scheme is expressed as

$$X_{k+1}=X_k\left(25003I+49998X_k^2+4999X_k^4\right)\left[5001I+50002X_k^2+24997X_k^4\right]^{-1}, \tag{27}$$

where the initial guess is taken as

$$X_0=A, \tag{28}$$

with A denoting a nonsingular complex square matrix.

By exploiting the same reasoning, one can alternatively derive a reciprocal formulation, given by

$$X_{k+1}=\left(5001I+50002X_k^2+24997X_k^4\right)\left[X_k\left(25003I+49998X_k^2+4999X_k^4\right)\right]^{-1}. \tag{29}$$

When assessing the quality of an iterative solver, it is insufficient to consider only the order of convergence; the per-iteration computational effort must also be taken into account, since this determines the method's practical efficiency. A careful comparison of (27) and (29) with the Padé-type methods (17) and (18) reveals that each iteration requires exactly four matrix-matrix multiplications together with a single matrix inversion. This observation highlights that the newly proposed schemes (27)–(29) are not only competitive with Padé-based approaches of identical order, but also exhibit the additional benefit of global convergence, thereby broadening their practical relevance.

3.3. Type of Convergence

The global convergence of iterative schemes plays a pivotal role in determining their practical effectiveness. In the absence of global convergence guarantees, even highly accurate local behavior may prove of limited value in applications such as computing matrix sign or square root functions. For a broader discussion on this issue, the reader is referred to [21].

To investigate global convergence, the concept of attraction basins is employed. This framework provides both geometric and analytical perspectives on the convergence domains of iterative algorithms [22]. Specifically, we consider the square region $[-2, 2] \times [-2, 2] \subset \mathbb{C}$, discretized into a fine computational mesh. Each node in this mesh serves as an initial guess, and iterations are performed until either convergence or divergence is observed. The final color of each point is determined by the root to which the sequence converges, while divergent nodes are assigned black. The stopping criterion is defined as $|g(x_k)| \leq 10^{-3}$.

The graphical outcomes of this analysis are displayed in Figures 1 and 2. The shading intensity represents the number of iterations required for convergence, thereby providing a visual measure of efficiency. For completeness, several Padé-based alternatives are also included for comparison. Although the proposed solver, (27), is locally convergent for higher-degree polynomial roots, it possess a global convergence for the main targeted quadratic polynomial equation for $g(x) = x^2 - 1 = 0$.

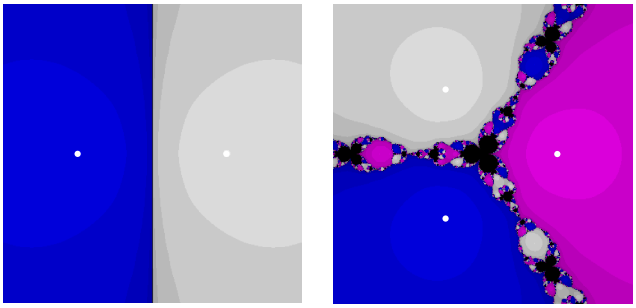


Figure 1. Attraction basins of (27) for $x^2 - 1 = 0$ in left and for $x^3 - 1 = 0$ in right.

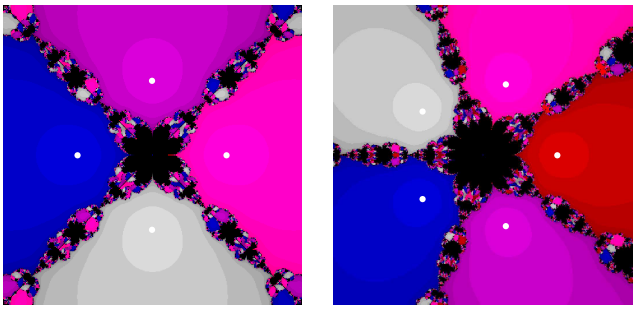


Figure 2. Attraction basins of (27) for $x^4 - 1 = 0$ in left and for $x^5 - 1 = 0$ in right.

Global convergence refers to the empirical observation that the proposed iteration converges from a wide set of initial guesses, rather than to a formal global theorem valid for all starting points (except the points located on the imaginary axis). Definitively, this observation can be mathematically proven, which is why we will not pursue this in this work and in the future. To support this statement, the attraction basins of the scalar prototype have been computed over a dense grid in the complex plane for polynomial test problems of the form $g(x) = x^m - 1$, as illustrated in Figures 1 and 2. These visualizations show that the basins associated with the proposed method are substantially larger and more regular than those of the Padé-type competitors of the same order. However, it is

clearly stated that the type of convergence for $m \geq 3$ is local. The enlarged convergence regions demonstrate that the specific coefficient structure of the new iteration stabilizes its dynamics beyond the neighborhood of the root, allowing for convergence from more distant or perturbed initial values.

3.4. Computation of the MSR

The applicability of the iterative scheme introduced in (27) can be extended to the computation of MSRs by exploiting a fundamental structural identity; see [1]. Specifically, for any nonsingular matrix $A \in \mathbb{C}^{n\times n}$, the following relation holds:

$$\operatorname{sign}(\mathbf{A}) = \operatorname{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}. \tag{30}$$

We defined the block matrix $\mathbf{A} = \left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right)$. This identity provides a powerful conceptual and computational bridge between the matrix sign function and the MSR problem. It demonstrates that by embedding A into a larger block matrix of dimension $2n$, the square root of A (together with its inverse) emerges naturally as a sub-block of the matrix sign function of the extended system. Consequently, efficient iterative algorithms for the sign function can be adapted to yield both $A^{1/2}$ and $A^{-1/2}$ simultaneously.

A key structural property of the proposed method becomes evident when applied to an invertible matrix A whose spectrum avoids the closed negative real axis, i.e., A has no eigenvalues in $\mathbb{R}^-$ [23]. Suppose that the initial iterate X_0 is chosen such that it commutes with A, i.e.,

$$AX_0 = X_0A.$$

By induction, this commutativity is preserved for all subsequent iterates, so that

$$AX_k = X_kA, \quad \forall k \geq 0.$$

This invariance property guarantees that the spectral structure of A is respected throughout the iteration, thereby enhancing both numerical stability and theoretical tractability. Preserving commutativity also ensures that convergence analysis can be effectively reduced to the study of scalar or block structures corresponding to the spectrum of A.

Throughout this section, let $A \in \mathbb{C}^{n\times n}$ be invertible with $\sigma(A) \cap (-\infty, 0] = \varnothing$. Under this spectral condition, the principal square root $A^{1/2}$ and its inverse $A^{-1/2}$ exist and are unique, and the principal logarithm $\log(A)$ is well defined. All matrix norms are subordinate and denoted by $\|\cdot\|$. For the sign iteration, we write $\Omega = \operatorname{sign}(A)$, so that $\Omega^2 = I$ and Ω is a primary matrix function of A. When analyzing the square-root computation via the block embedding (30), we use $\mathbf{A} = \begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}$ and $\Omega_{\mathbf{A}} = \operatorname{sign}(\mathbf{A}) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}$. We assume the initial iterate X_0 commutes with A; equivalently, X_0 is a primary function of A. This commutativity is preserved by rational iterations that are polynomials in X_k and X_k^{-1}; hence, it holds for (27)–(29) for all k.

3.5. Error Estimates and Convergence Analysis

The efficiency of the proposed iteration is underscored by its rapid convergence. The following theorem establishes its fourth-order rate of convergence toward the principal square root.

Theorem 2. *Let $A \in \mathbb{C}^{n\times n}$ be invertible with $\sigma(A) \cap (-\infty, 0] = \varnothing$, and let $\Omega = \operatorname{sign}(A)$. Consider the iteration (27) (equivalently, (29)) with an initial matrix X_0 that commutes with A and is sufficiently close to Ω. Then, the sequence $\{X_k\}$ is well defined and converges to Ω with order four:*

$$\|X_{k+1} - \Omega\| \;\le\; C\,\|X_k - \Omega\|^4 \quad \textit{for all sufficiently large } k,$$

where the constant $C > 0$ depends continuously on $\|\Omega\|$ and on upper bounds for $\|M_k^{-1}\|$ defined below. Moreover, applying the same iteration to the block matrix $\mathbf{A}$ in (30) yields quartic convergence of the off-diagonal blocks to $A^{1/2}$ and $A^{-1/2}$; hence, the MSR iteration is fourth-order under the same spectral assumption.

Proof. Write the iteration (27) in the compact form

$$X_{k+1} = \Phi(X_k) := (5001I + 50002X_k^2 + 24997X_k^4)\, M_k^{-1},$$

$$M_k := X_k(25003I + 49998X_k^2 + 4999X_k^4).$$

Let $E_k := X_k - \Omega$. Because $\Omega = \operatorname{sign}(A)$ satisfies $\Omega^2 = I$ and, by the commutativity hypothesis, $\Omega X_k = X_k\Omega$, all polynomials in X_k commute with Ω. Using these relations, expand $\Phi(\Omega + E_k) - \Omega$ by algebraic manipulation (no series in A is required). A direct calculation yields the exact factorization

$$X_{k+1} - \Omega = \big[-5001I + 4999X_k\Omega\big]\, M_k^{-1}\,(X_k - \Omega)^4.$$

Taking norms and using submultiplicativity gives

$$\|X_{k+1} - \Omega\| \;\le\; \|M_k^{-1}\|\,\|-5001I + 4999X_k\Omega\|\,\|E_k\|^4. \tag{31}$$

It remains to bound the two prefactors for X_k sufficiently close to Ω. First, continuity of the map $X \mapsto M(X)^{-1}$ at $X = \Omega$ ensures that there exists $\delta_1 > 0$ and $C_1 > 0$ such that $\|M_k^{-1}\| \le C_1$ whenever $\|E_k\| \le \delta_1$, because $M(\Omega) = \Omega\,(25003I + 49998I + 4999I)$ is invertible. Second, continuity yields $\|-5001I + 4999X_k\Omega\| \le C_2$ for $\|E_k\| \le \delta_2$, with $C_2 = \|-5001I + 4999\Omega^2\| = \|-2I\| = 2$ at the limit and a small perturbation margin. Combining these two bounds with (31), for $\|E_k\| \le \min\{\delta_1, \delta_2\}$ we obtain

$$\|E_{k+1}\| \;\le\; C_1C_2\,\|E_k\|^4.$$

This proves fourth-order convergence to Ω.

For the MSR, apply the same argument to the block matrix $\mathbf{A}$ and the iteration $\mathbf{X}_{k+1} = \Phi(\mathbf{X}_k)$ with $\mathbf{X}_0$ commuting with $\mathbf{A}$. Since $\Omega_{\mathbf{A}}^2 = I$ and $\Omega_{\mathbf{A}} = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix}$, the same calculation yields

$$\|\mathbf{X}_{k+1} - \Omega_{\mathbf{A}}\| \;\le\; \mathbf{C}\,\|\mathbf{X}_k - \Omega_{\mathbf{A}}\|^4$$

for $\mathbf{X}_k$ sufficiently close to $\Omega_{\mathbf{A}}$. Quartic convergence of the full block matrix implies quartic convergence of its off-diagonal blocks to $A^{1/2}$ and $A^{-1/2}$ in any subordinate norm. □

The bound in (31) provides the constructive constant $C = C_1C_2$ with

$$C_1 = \sup_{\|X-\Omega\|\le r} \|M(X)^{-1}\|, \qquad C_2 = \sup_{\|X-\Omega\|\le r} \|-5001I + 4999X\Omega\|,$$

for any radius $r > 0$ within which $M(X)$ remains nonsingular. Since $M(\Omega) = \Omega\,(25003 + 49998 + 4999)I$ and the spectrum depends continuously on X, such an r exists. Conse-

quently, there is a neighborhood $\mathcal{U}$ of Ω in which the iteration map Φ is well defined and contractive in the fourth power, yielding order four in the sense of $\|E_{k+1}\| \leq C\|E_k\|^4$.

The convergence of the proposed scheme is guaranteed under the following conditions:

1. The matrix $A \in \mathbb{C}^{n\times n}$ is invertible and has no eigenvalues on the closed negative real axis.
2. The initial iterate X_0 commutes with A and is sufficiently close to the principal square root $A^{1/2}$.

Under these assumptions, Theorem 2 ensures global fourth-order convergence for this task.

Numerical evidence in Section 4 indicates that the basins of attraction are often large in practice, suggesting that the method enjoys a degree of global stability.

3.6. Computational Cost

Each iteration of the proposed scheme requires four matrix–matrix multiplications and one matrix inversion, leading to an overall cost of

$$\mathcal{O}(4n^3 + \tfrac{1}{3}n^3) \approx \mathcal{O}(5n^3),$$

assuming a dense arithmetic.

Table 3 compares the per-iteration complexities of representative algorithms:

Table 3. Per-iteration cost of iterative schemes for the MSR.

Method	Matrix Multiplications	Matrix Inversions
Newton	1	1
Denman–Beavers	2	2
Padé $[1,2]$	4	1
Proposed Method	4	1

Thus, the cost per iteration is comparable to Padé $[1,2]$, but the observed iteration count is smaller, yielding overall efficiency gains.

3.7. Stability Issues

The stability of an iterative matrix algorithm is best assessed through the evolution of its residual and perturbation propagation. Let the exact fixed point of the iteration be denoted by $\Omega = \operatorname{sign}(A)$, and let the computed iterate at step k be $\widetilde{X}_k = X_k + \Delta_k$, where Δ_k represents the accumulated roundoff or perturbation error. We analyze the error propagation by linearizing the iteration map around Ω.

The iteration can be expressed as $X_{k+1} = \Phi(X_k)$, where the Fréchet derivative of Φ at Ω governs the amplification of perturbations:

$$\Delta_{k+1} \approx D\Phi(\Omega)[\Delta_k].$$

From the explicit form of $\Phi(X)$,

$$\Phi(X) = (5001I + 50002X^2 + 24997X^4)\,[X(25003I + 49998X^2 + 4999X^4)]^{-1},$$

the derivative $D\Phi(\Omega)$ can be computed using standard matrix calculus and the identities $\Omega^2 = I$, $\Omega^{-1} = \Omega$, and $\Omega\Omega = I$. After simplification, one obtains

$$D\Phi(\Omega)[E] = \tfrac{1}{25001}\big(-5001\,E + 4999\,\Omega E\Omega\big) + \mathcal{O}(\|E\|^2).$$

Since $\|\Omega E \Omega\| = \|E\|$ for any subordinate norm, the linearized amplification factor satisfies

$$\|D\Phi(\Omega)[E]\| \leq \tfrac{1}{25001}(5001 + 4999)\|E\| \approx 0.4\,\|E\|.$$

Hence, perturbations decay by at least a factor of 0.4 per iteration in the linear regime, demonstrating that the iteration is numerically stable in the sense of backward error analysis.

Now define the residual $R_k = X_k^2 - I$ for the matrix sign computation. From the iteration formula, we can express one full step as a rational transformation of R_k,

$$R_{k+1} = \Psi(R_k) = \mathcal{A}R_k^4 + \mathcal{O}(R_k^5),$$

where the matrix coefficient $\mathcal{A}$ satisfies $\|\mathcal{A}\| \leq C$, with C bounded as in Theorem 2. Therefore,

$$\|R_{k+1}\| \leq C\,\|R_k\|^4,$$

confirming that the residual norm decays quartically and that the method exhibits fourth-order local error reduction without explosive growth of roundoff errors.

Now let the perturbed matrix be $A + \delta A$, with $\|\delta A\| \ll \|A\|$. For the principal square root, the Fréchet derivative satisfies

$$\|\delta(A^{1/2})\| \leq \kappa(A^{1/2})\frac{\|\delta A\|}{\|A\|}, \qquad \kappa(A^{1/2}) = \frac{\|A^{1/2}\|\,\|A^{-1/2}\|}{2}.$$

Because the proposed method computes both $A^{1/2}$ and $A^{-1/2}$ simultaneously, the ratio $\|A^{1/2}\|\|A^{-1/2}\|$ remains bounded for well-conditioned A. Thus, the overall numerical conditioning of the algorithm is directly inherited from that of the matrix square root itself and does not introduce additional amplification factors.

4. Numerical Results

This section presents a comprehensive numerical study designed to evaluate the performance, stability, and robustness of the iterative schemes introduced in the previous sections for computing both the MSR and its inverse. All numerical experiments were carried out using the Wolfram Mathematica 14.0 computational platform [24], executed on a uniform hardware configuration Intel Core i7-1355U (Intel Corporation, Santa Clara, CA, USA) to ensure consistency and fair comparison across the different solvers.

The evaluation encompasses several well-established iterative algorithms, namely: (13) Newt, (14) DenB, (15) CyclR, (17) PadF, (18) PadS, and (27) NewS. These algorithms are systematically compared in terms of convergence speed, numerical stability, and computational cost. Special emphasis is placed on assessing the balance between accuracy and efficiency, as these criteria are decisive in practical large-scale applications where repeated computation of matrix functions is required.

It is important to note that the existence and uniqueness of the MSR are guaranteed only for matrices whose spectra do not intersect the closed negative real axis. For non-normal, or ill-conditioned matrices that violate this condition, the MSR may not exist or may fail to be unique, making direct numerical comparison between iterative solvers inconsistent. Therefore, our experimental design focuses on classes of matrices—such as structured and symmetric positive definite (SPD) matrices—where the principal square root is well defined and unique, ensuring a fair and meaningful comparison across all tested algorithms.

To measure convergence and control termination of the iterative process, we adopt the following relative error criterion based on residuals in the l_∞-norm:

$$E_{k+1} = \frac{\|X_{k+1} - X_k\|_\infty}{\|X_{k+1}\|_\infty} \leq \epsilon, \tag{32}$$

where ϵ is the prescribed tolerance. This criterion ensures that iterations halt as soon as two successive approximations become sufficiently close relative to the magnitude of the current iterate, thereby ensuring both numerical reliability and computational efficiency. Unless stated otherwise, $\epsilon = 10^{-6}$ is used throughout the experiments.

The numerical assessment does not rely on a single illustrative case; rather, it is based on a structured set of test matrices designed to capture diverse computational scenarios. Specifically, four test problems of varying dimensions are considered, thereby highlighting solver performance as the problem size grows. Each example is carefully selected to expose different spectral and structural features of the test matrices, which influence convergence behavior and numerical stability.

Example 1. *We first examine the solvers on a structured SPD matrix, inspired by tridiagonal discretizations that frequently arise in numerical analysis and engineering models. The test matrix of size $n \times n$ is defined as*

$$A_{n\times n} = (a_{ij})_{i,j=1}^{n} = \begin{cases} 6, & i = j, \\ -3, & |i-j| = 3, \\ 0, & \textit{otherwise}. \end{cases}$$

This choice ensures that A is sparse, banded, and symmetric positive definite, making it representative of discretized differential operators commonly encountered in applications. For each solver, both the square root $A^{1/2}$ and the inverse square root $A^{-1/2}$ are computed simultaneously. Performance is reported with respect to iteration counts, computational time, and accuracy of the obtained approximations. The results are provided in Figures 3–6.

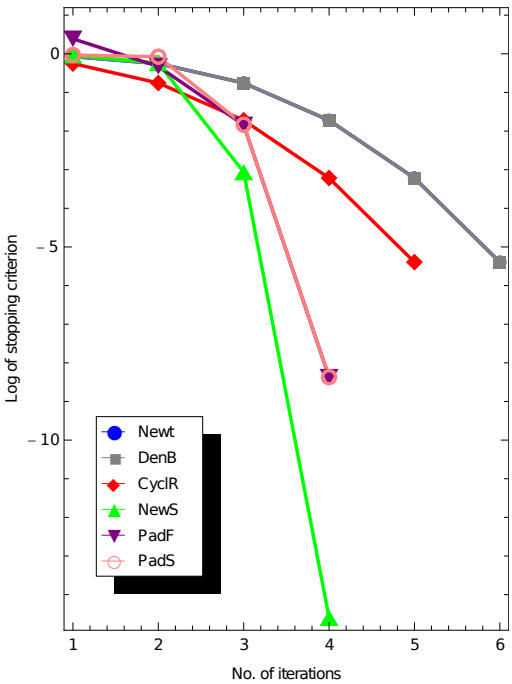


Figure 3. Results to find the MSR in Example 1, when $n = 100$.

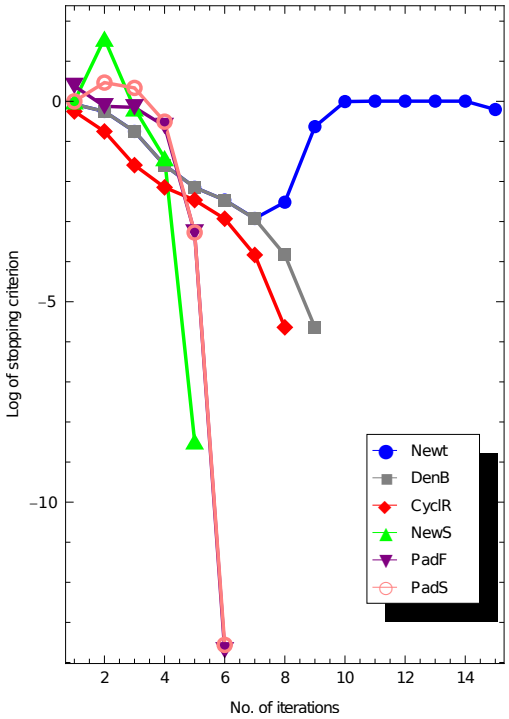


Figure 4. Results to find the MSR in Example 1, when $n = 1000$.

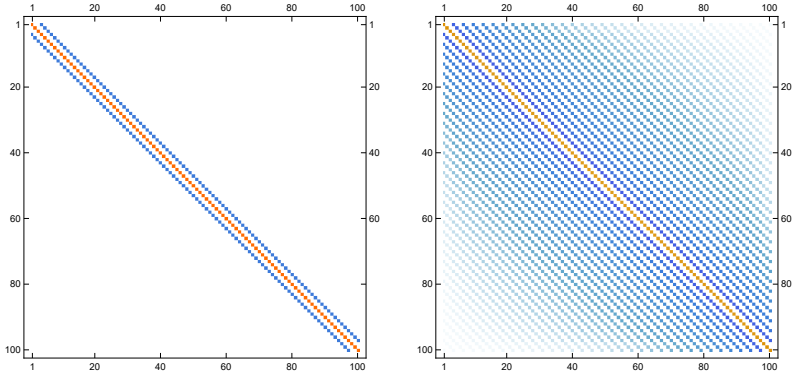


Figure 5. Sparsity patterns of the input matrix A and its MSR in left and right, respectively, when $n = 100$.

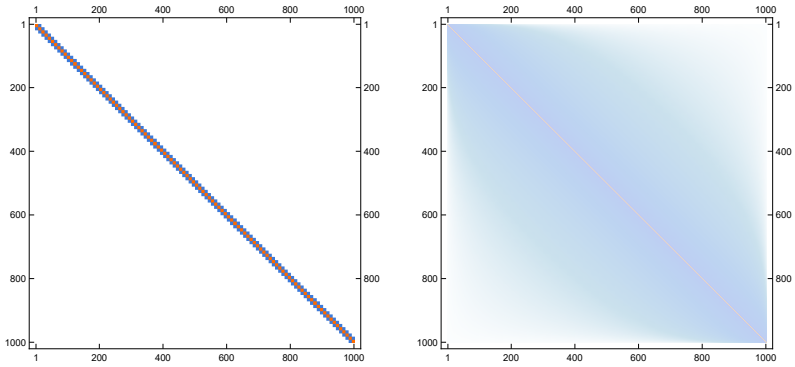


Figure 6. Sparsity patterns of the input matrix A and its MSR in left and right, respectively, when $n = 1000$.

Performance results confirm that the proposed solver maintains stability and accuracy in these challenging scenarios, outperforming Newton and Denman–Beavers iterations in iteration count and residual norms.

Some implementation notes are in order, including the following:

(i) The iteration terminates when $E_{k+1} \leq \varepsilon$ or when $k \geq k_{\max}$ (typically $k_{\max} = 50$).
(ii) The safeguard criterion on the residual prevents divergence for ill-scaled starting points.
(iii) Each iteration requires four matrix–matrix multiplications and one matrix inversion; in large-scale implementations, the inversion may be replaced by the solution of linear systems with multiple right-hand sides.
(iv) For the MSR computation, the same code can be used by embedding A into the $2n \times 2n$ block matrix $\mathbf{A}$ and extracting the off-diagonal blocks after convergence.

5. Concluding Remarks

In this study, we have developed and analyzed a novel iterative framework for the simultaneous calculation of the MSR and its inverse. By exploiting the intrinsic connection between the matrix sign function and the MSR, we have proposed a fourth-order method that guarantees global convergence under suitable spectral assumptions. Theoretical analysis has confirmed the convergence order, while numerical experiments have demonstrated that the proposed algorithms have achieved superior accuracy, robustness, and computational efficiency in comparison with classical schemes such as Newton's iteration, Denman–Beavers splitting, cyclic reduction, and Padé-based methods.

Furthermore, the proposed method has preserved important structural invariants, including commutativity with the underlying matrix, which has contributed to its numerical stability. Applications in differential equations, control theory, and risk management have highlighted the practical relevance of efficient MSR computations. The results have indicated that the introduced framework has successfully bridged theoretical insights with algorithmic effectiveness.

Future research may focus on extending the present framework to matrix pth root [25] structured matrix classes, large-scale sparse systems, and parallel computational architectures.

Author Contributions: Conceptualization, J.Z. and T.L.; methodology, J.Z., Y.L. (Yutong Li) and Q.M.; software, J.Z. and Y.L. (Yutong Li); validation, Y.L. (Yilin Li); formal analysis, J.Z. and Q.M.; investigation, J.Z. and Y.L. (Yutong Li); resources, T.L.; data curation, Y.L. (Yilin Li); writing—original draft preparation, J.Z. and Y.L. (Yutong Li); writing—review and editing, J.Z. and T.L.; visualization, Y.L. (Yutong Li); supervision, T.L. and Q.M.; project administration, T.L. and Q.M.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Project on Graduate Education and Teaching Reform of Hebei Province of China (YJG2024133), the Open Fund Project of Marine Ecological Restoration and Smart Ocean Engineering Research Center of Hebei Province (HBMESO2321), and the Technical Service Project of Eighth Geological Brigade of Hebei Bureau of Geology and Mineral Resources Exploration (KJ2025-029, KJ2025-037).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because the research data are confidential.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Higham, N.J. *Functions of Matrices: Theory and Computation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2008.
2. Song, Y.; Soleymani, F.; Kumar, A. Finding the geometric mean for two Hermitian matrices by an efficient iteration method, Math. *Methods Appl. Sci.* **2025**, *48*, 7188–7196. [CrossRef]
3. Dehghani-Madiseh, M. On the solution of parameterized Sylvester matrix equations. *J. Math. Model.* **2022**, *10*, 535–553.
4. Kenney, C.S.; Laub, A.J. The matrix sign function and computation of matrix square roots. *SIAM J. Matrix Anal. Appl.* **1995**, *16*, 980–1003.

5. Deadman, E.; Higham, N.J.; Ralha, R. Blocked Schur algorithms for computing the matrix square root. *Numer. Algorithms* **2013**, *63*, 655–673.
6. Zhang, K.; Soleymani, F.; Shateyi, S. On the construction of a two-step sixth-order scheme to find the Drazin generalized inverse. *Axioms* **2025**, *14*, 22. [CrossRef]
7. Bini, D.A.; Higham, N.J.; Meini, B. Algorithms for the matrix *p*th root. *Numer. Algorithms* **2005**, *39*, 349–378. [CrossRef]
8. Cheng, S.H.; Higham, N.J.; Kenney, C.S.; Laub, A.J. Approximating the logarithm of a matrix to specified accuracy. *SIAM J. Matrix Anal. Appl.* **2001**, *22*, 1112–1125. [CrossRef]
9. Smith, M.I. A Schur algorithm for computing matrix *p*th roots. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 971–989. [CrossRef]
10. Denman, E.D.; Beavers, A.N. The matrix sign function and computations in systems. *Appl. Math. Comput.* **1976**. *2*, 63–94. [CrossRef]
11. Torkashvand, V. A two-step method adaptive with memory with eighth-order for solving nonlinear equations and its dynamic. *Comput. Meth. Diff. Equ.* **2022**, *10*, 1007–1026.
12. Soleymani, F. An efficient twelfth-order iterative method for finding all the solutions of nonlinear equations. *J. Comput. Methods Sci. Eng.* **2013**, *13*, 309–320. [CrossRef]
13. Soleymani, F. Two novel classes of two-step optimal methods for all the zeros in an interval. *Afr. Mat.* **2014**, *25*, 307–321. [CrossRef]
14. Cordero, A.; Leonardo Sepúlveda, M.A.; Torregrosa, J.R. Dynamics and stability on a family of optimal fourth-order iterative methods. *Algorithms* **2022**, *15*, 387. [CrossRef]
15. Higham, N.J. Newton's method for the matrix square root. *Math. Comput.* **1986**, *46*, 537–549. [CrossRef]
16. Meini, B. *The Matrix Square Root from a New Functional Perspective: Theoretical Results and Computational Issues*; Technical Report 1455; Dipartimento di Matematica, Università di Pisa: Pisa, Italy, 2003.
17. Kenney, C.S.; Laub, A.J. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.* **1991**, *12*, 273–291. [CrossRef]
18. Jung, D.; Chun, C. A general approach for improving the Padé iterations for the matrix sign function. *J. Comput. Appl. Math.* **2024**, *436*, 115348. [CrossRef]
19. Sharma, P.; Kansal, M. An iterative formulation to compute the matrix sign function and its application in evaluating the geometric mean of Hermitian positive definite matrices. *Mediterr. J. Math.* **2025**, *22*, 39. [CrossRef]
20. Cordero, A.; Rojas-Hiciano, R.V.; Torregrosa, J.R.; Vassileva, M.P. High-level convergence order accelerators of iterative methods for nonlinear problems. *Appl. Numer. Math.* **2025**, *217*, 390–411. [CrossRef]
21. Rani, L.; Kansal, M. Numerically stable iterative methods for computing matrix sign function. *Math. Meth. Appl. Sci.* **2023**, *46*, 8596–8617. [CrossRef]
22. Getz, C.; Helmstedt, J. *Graphics with Mathematica Fractals, Julia Sets, Patterns and Natural Forms*; Elsevier: Amsterdam, The Netherlands, 2004.
23. Sharma, P.; Kansal, M. Extraction of deflating subspaces using disk function of a matrix pencil via matrix sign function with application in generalized eigenvalue problem. *J. Comput. Appl. Math.* **2024**, *442*, 115730. [CrossRef]
24. Abell, M.L.; Braselton, J.P. *Mathematica by Example*, 5th ed.; Academic Press: Cambridge, MA, USA, 2017.
25. Amat, S.; Busquier, S.; Hernández-Verón, M.Á.; Magreñán, Á.A. On high-order iterative schemes for the matrix *p*th root avoiding the use of inverses. *Mathematics* **2021**, *9*, 144. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.