fractal and fractional

MDPI

Article

Accelerated Sign-Function-Based Iterations for Matrix Square Roots with Fourth-Order Convergence

Shuai Wang [1], Zhanmeng Yang [2], Sommayeh Salehi [3], Taher Lotfi [3], Xiaoxi Hu [4], Yakun Li [5], Wenhao Kang [6] and Tao Liu [7,*]

[1] Department of Mathematics and Physics, Changchun Guanghua University, Changchun 130033, China
[2] Sydney Smart Technology College, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
[3] Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan 65181-15743, Iran
[4] State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing 100044, China
[5] School of Economics and Management, Yanshan University, Qinhuangdao 066104, China
[6] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China
[7] School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
* Correspondence: liutao@neuq.edu.cn

Abstract

Motivated by the close relationship between the matrix square root and the matrix sign function, this paper develops a new high-order iterative framework for computing the principal square root of a matrix and its inverse. The proposed approach is derived from a rational fixed-point iteration associated with a scalar nonlinear equation and is extended consistently to the matrix setting. The method is shown to be globally convergent and to achieve fourth-order convergence. Numerical experiments demonstrate that the new scheme outperforms several classical and Padé-based methods.

Keywords: matrix sign; rational iterations; Hermite interpolation; basin of attraction; Jordan block

MSC: 65F60; 15A24; 41A25

1. Introduction

A matrix function refers to a mathematical operation that accepts a matrix—typically a square matrix—as its input and produces another matrix of identical dimensions as output. Multiple equivalent frameworks exist in the literature to define functions of matrices, with approaches including the Jordan canonical form, the Cauchy integral representation, and formulations based on Hermite matrix polynomials; see the details in [1]. Let $B \in \mathbb{C}^{n\times n}$ and assume that ψ is the minimal polynomial of B. Suppose that [2]

$$\psi(z) = \prod_{i=1}^{s}(z-\lambda_i)^{m_i},$$

where $\lambda_1, \lambda_2, \ldots, \lambda_s$ are the eigenvalues of B and $m_i \geq 1$ is the size of the largest Jordan block of B associated with λ_i (equivalently, the multiplicity of $(z-\lambda_i)$ in the minimal polynomial). Also let f be *holomorphic* (analytic) on an open set $\mathcal{D} \subset \mathbb{C}$ containing $\sigma(B)$. In particular, f has derivatives of all orders on $\mathcal{D}$, so the quantities $f^{(k)}(\lambda_i)$ required below

Check for updates

Academic Editors: Ljubiša D. R. Kočinac and Tayyab Kamran

Received: 20 January 2026
Revised: 8 February 2026
Accepted: 11 February 2026
Published: 14 February 2026

are well defined. Define p to be the unique polynomial of degree $< \deg(\psi) = \sum_{i=1}^{s} m_i$ satisfying the Hermite interpolation conditions [3]:

$$p^{(k)}(\lambda_i) = f^{(k)}(\lambda_i), \qquad i = 1, 2, \ldots, s, \quad k = 0, 1, \ldots, m_i - 1. \tag{1}$$

Then, the matrix function $f(B)$ is expressed by the Hermite interpolation rule

$$f(B) := p(B). \tag{2}$$

This definition yields a primary matrix function and coincides with the Jordan canonical form definition. In the special case that B is diagonalizable (so that all $m_i = 1$), the conditions (1) reduce to standard Lagrange interpolation at the eigenvalues. As an example, consider the principal matrix square root (MSR), where $f(z) = z^{1/2}$ denotes the principal branch (so that $\Re(z^{1/2}) \geq 0$ for $z \notin \mathbb{R}_{\leq 0}$) [4]. Consider

$$B = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix}.$$

The spectrum is $\sigma(B) = \{4\}$ and B is a Jordan block, hence the minimal polynomial is $\psi(z) = (z-4)^2$. Therefore, $s = 1$, $\lambda_1 = 4$, and $m_1 = 2$, so p must satisfy the Hermite conditions

$$p(4) = f(4) = 2, \qquad p'(4) = f'(4) = \frac{1}{2\sqrt{4}} = \frac{1}{4}.$$

Since $\deg(p) < 2$, write $p(z) = a + b(z-4)$. Enforcing the above conditions gives $a = 2$, $b = \frac{1}{4}$, hence

$$p(z) = 2 + \frac{1}{4}(z-4). \tag{3}$$

By (2), the principal square root is $B^{1/2} = p(B)$. Using (3),

$$B^{1/2} = 2I + \frac{1}{4}(B - 4I) = 2I + \frac{1}{4}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \frac{1}{4} \\ 0 & 2 \end{pmatrix}.$$

A direct verification confirms this is a square root:

$$\left(\begin{pmatrix} 2 & \frac{1}{4} \\ 0 & 2 \end{pmatrix}\right)^2 = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix} = B.$$

Therefore, the Hermite interpolating polynomial definition (2) yields the correct principal MSR for this non-diagonalizable 2×2 example.

In terms of formal definitions of matrix functions, now if f is analytic on an open set $\mathcal{D} \subset \mathbb{C}$ containing $\sigma(B)$, then a standard analytic functional calculus defines [5]:

$$f(B) = \frac{1}{2\pi i} \int_{\Gamma} f(z)\,(zI - B)^{-1}\,dz, \tag{4}$$

where Γ is any positively oriented contour enclosing $\sigma(B)$ and contained in $\mathcal{D}$.

A matrix function $f(B)$ is called primary if it is defined by applying a single-valued scalar function f to the Jordan canonical form of B (equivalently, by Hermite interpolation or by an analytic functional calculus), and then $f(B)$ commutes with B and depends only on the eigenvalues and Jordan structure [5]. Typical examples include the matrix exponential and, under $\sigma(B) \cap \mathbb{R}_{\leq 0} = \varnothing$, the principal MSR. In contrast, a non-primary matrix function is any matrix satisfying an algebraic relation (e.g., $X^2 = I$) without being representable as

$f(B)$ for a single scalar function applied consistently across all Jordan blocks; such functions can depend on additional choices and need not commute with B. The matrix sign function, defined for $\sigma(B) \cap i\mathbb{R} = \varnothing$ by mapping eigenvalues in the open right/left half-planes to ± 1, fits naturally within the primary matrix-function framework on a suitable open set containing $\sigma(B)$, and it plays a central role in computing other matrix functions, including the principal square root, through block formulations [6].

A standard sufficient condition ensuring the existence and uniqueness of the principal MSR is [7]

$$\sigma(B) \cap \mathbb{R}_{\leq 0} = \varnothing, \tag{5}$$

in which case, there exists a unique root $B^{1/2}$ whose eigenvalues lie in the open right half-plane. This principal root is a primary matrix function and depends analytically on B. Moreover, it admits an integral representation that is particularly useful for analysis as follows [1]:

$$B^{1/2} = \frac{2}{\pi} B \int_0^{\infty} (t^2 I + B)^{-1} \, dt. \tag{6}$$

Iterative techniques have gained prominence, particularly in scenarios where matrix entries evolve dynamically over time or where high-quality initial approximations are available [8,9]. In the present study, we introduce a novel higher-order iteration solver specifically designed to compute both the square root and its inverse for matrices that satisfy certain admissibility conditions. We demonstrate that, under appropriate assumptions, the proposed algorithm exhibits global convergence towards the principal square root. Moreover, the method achieves a fourth-order rate of convergence, thereby offering an efficient tool for practical computations.

A central numerical issue is sensitivity of the MSR to perturbations in the data. Let B satisfy (5) and let $B^{1/2}$ denote its principal square root. The Fréchet derivative of the square-root map at B in direction E, denoted by $L_{\sqrt{\cdot}}(B, E)$, is characterized as the unique solution X of the Sylvester equation

$$B^{1/2} X + X B^{1/2} = E. \tag{7}$$

This relation underpins perturbation bounds and conditioning analysis. In particular, one can define a relative condition number for the MSR problem by [10]

$$\kappa_{\sqrt{\cdot}}(B) = \lim_{\varepsilon \to 0} \sup_{\|E\| \leq \varepsilon \|B\|} \frac{\|L_{\sqrt{\cdot}}(B, E)\|}{\varepsilon \|B^{1/2}\|}, \tag{8}$$

which quantifies the worst-case relative amplification of small relative perturbations in B into relative perturbations in $B^{1/2}$. The norm $\|\cdot\|$ used throughout the paper is norm-2 unless it is stated otherwise. This perspective is important because it clarifies when high-order iterations translate into improved accuracy, and when the attainable accuracy is dominated by intrinsic ill-conditioning.

A deeper link between the sign function and the MSR can be described through a structured block matrix. If B is nonsingular and satisfies (5), then the $2n \times 2n$ matrix [11]

$$\mathcal{H} = \begin{bmatrix} 0 & B \\ I & 0 \end{bmatrix}, \tag{9}$$

has no eigenvalues on the imaginary axis, and the sign of $\mathcal{H}$ encodes both $B^{1/2}$ and $B^{-1/2}$:

$$\operatorname{sign}(\mathcal{H}) = \begin{bmatrix} 0 & B^{1/2} \\ B^{-1/2} & 0 \end{bmatrix}. \tag{10}$$

Identity (10) provides a conceptual justification for designing algorithms that compute the principal MSR and its inverse simultaneously, and it also serves as a theoretical bridge between sign iterations and MSR iterations developed later in this manuscript.

The present study is fundamentally centered on iterative schemes [12], particularly in the context of matrix computations. These iterative procedures, when equipped with a suitably chosen initial matrix guess, operate as fixed-point or Newton-type algorithms [13] that generate sequences of matrix iterates converging to the desired solution. While the linkage between iterative solvers to tackle the matrix sign function and classical nonlinear root-finding approaches [14] might not be immediately apparent, it becomes evident upon closer inspection that many fixed-point algorithms designed for sign computation are, in fact, derived by adapting root-solving techniques to the nonlinear (scalar) equation below [15,16]:

$$g(x) = x^2 - 1 = 0. \tag{11}$$

High-order iterative schemes for the matrix sign function and sign-based constructions for matrix square roots have substantial literature. Beyond the classical Newton and Padé/Kenney–Laub families, many globally convergent methods of order four and higher have been proposed and analyzed; see, for example, ref. [17] and the references therein. Recently, Zhu et al. [18] developed fourth-order sign-function-based algorithms for the simultaneous computation of the principal MSR and its inverse. The present manuscript follows this broad and well-established paradigm, but contributes a different quartic rational iteration obtained from a divided-difference correction of a scalar root solver, leading to a small-coefficient matrix iteration. We make the connection between our coefficients and a one-parameter quartic family explicit in Section 3, and we motivate our parameter choice from a finite-precision viewpoint: smaller integer coefficients reduce coefficient growth in the evaluation of the associated matrix polynomials, which is beneficial for numerical robustness while preserving the same asymptotic order and the same per-iteration algebraic workload.

The rest of the manuscript is structured as follows. Section 2 offers a concise yet comprehensive overview of several established iterative algorithms of varying convergence orders that are utilized for attaining the MSR. In Section 3, we introduce a new iteration solver tailored for the simultaneous computation of both the MSR and its inverse. Section 4 investigates the type of convergence as well as the attraction basis for the fixed-point-type proposed solver. The rate of convergence of the solver is analyzed in detail to substantiate its robustness in Section 5. Section 6 is dedicated to evaluating the practical effectiveness of the proposed iterative framework. To this end, we conduct a suite of numerical experiments. Ultimately, Section 7 concludes the study by giving the findings and suggesting directions for further research.

2. Overview of Classical Schemes

Applying the classical Newton iteration to (11) yields the well-known iterative process for computing the matrix sign function [19,20]:

$$X_{j+1} = \frac{1}{2}\left(X_j + X_j^{-1}\right). \tag{12}$$

Here, $\{X_j\}$ denotes the sequence of matrix iterates generated by the method. When X_0 has no eigenvalues on the imaginary axis, iteration (12) converges quadratically to $\operatorname{sign}(X_0)$, and its convergence can be understood through the transformation of eigenvalues under

the scalar map $y \mapsto (y + y^{-1})/2$. In particular, if λ is an eigenvalue of X_j, then the corresponding eigenvalue at the next iteration satisfies

$$\lambda_{j+1} = \frac{1}{2}\left(\lambda_j + \lambda_j^{-1}\right), \tag{13}$$

which drives λ_j rapidly toward ± 1 depending on the sign of its real part. One of the most classical techniques for computing the MSR is Newton's scheme employed directly to the nonlinear matrix equation $X^2 - B = 0$. This leads to the iterative scheme

$$X_{j+1} = \frac{1}{2}\left[X_j + BX_j^{-1}\right], \tag{14}$$

where X_0 is a suitably chosen initial approximation. Under standard spectral assumptions, iteration (14) converges quadratically to the principal MSR $B^{1/2}$. However, from a numerical standpoint, this formulation may suffer from instability or slow convergence when X_0 is poorly scaled or when B has eigenvalues close to the negative real axis, leading to ill-conditioned intermediate inversions; see also [21].

To improve robustness and numerical stability, Denman and Beavers gave a coupled solver that simultaneously approximates the square root and its inverse [22]. Their method is given by

$$\begin{cases} Z_0 = I, \ \ X_0 = B, \ \ j = 0, 1, \cdots, \\ X_{j+1} = \frac{1}{2}[X_j + Z_j^{-1}], \\ Z_{j+1} = \frac{1}{2}[Z_j + X_j^{-1}]. \end{cases} \tag{15}$$

Here, X_j converges to $B^{1/2}$ while Z_j converges to $B^{-1/2}$. This symmetry leads to improved numerical behavior compared to the single-sequence Newton iteration, particularly for poorly scaled matrices.

Another important approach for computing MSRs is based on cyclic reduction, originally introduced by Meini [23]. The method is defined by the iteration

$$\begin{cases} Z_0 = 2(I + B), \ \ X_0 = I - B, \ \ j = 0, 1, \cdots, \\ X_{j+1} = -X_j Z_j^{-1} X_j, \\ Z_{j+1} = Z_j - 2X_j Z_j^{-1} X_j. \end{cases} \tag{16}$$

This algorithm arises naturally from block Gaussian elimination applied to structured matrix equations and exhibits quadratic convergence. During the asymptotic phase, the sequence $\{Z_j\}$ satisfies

$$\lim_{j \to \infty} Z_j = 4B^{1/2}, \tag{17}$$

from which the principal square root can be recovered by appropriate scaling; see also [24]. Cyclic reduction methods are especially attractive for structured matrices and large-scale problems, as they rely primarily on matrix multiplications and linear solves.

It is also necessary here to recall that a general and widely studied class of iterative methods for computing the matrix sign function was offered by Kenney and Laub [25]. These schemes are derived from Padé approximations of the scalar function

$$f(\zeta) = \frac{1}{\sqrt{1 - \zeta}},$$

combined with the transformation $\zeta = 1 - u^2$, which maps the problem of evaluating the sign function to rational approximation. Using this framework, the sign function can be written as

$$\operatorname{sign}(u) = s = \frac{u}{(u^2)^{1/2}} = \frac{u}{(1-\zeta)^{1/2}}. \tag{18}$$

Padé approximants of f provide rational functions that are near-optimal in approximating f over prescribed regions of the complex plane, leading to highly efficient matrix iterations.

Let P_{κ_1,κ_2} and Q_{κ_1,κ_2} denote the numerator and denominator polynomials of the $[\kappa_1, \kappa_2]$ Padé approximant of f. Assuming $\kappa_1 + \kappa_2 \geq 1$, the corresponding Padé-based iteration for the sign function is provided as

$$u_{j+1} = \frac{u_j P_{\kappa_1,\kappa_2}(1-u_j^2)}{Q_{\kappa_1,\kappa_2}(1-u_j^2)} := \varphi_{2\kappa_1+1,2\kappa_2}, \tag{19}$$

which converges to ± 1 with order $1 + \kappa_1 + \kappa_2$.

These Padé-based methods and their reciprocal forms encompass several well-known optimal iterations, including the globally convergent $[\kappa_1/\kappa_1]$ and $[(\kappa_1 - 1)/\kappa_1]$ members. Their global convergence properties stem from the fact that the underlying rational functions map the right and left half-planes into themselves, a feature that is crucial for numerical stability; see also [26].

Among these schemes, certain Padé iterations achieve fourth-order convergence while maintaining global stability. Two representative fourth-order methods are

$$X_{j+1} = [I + 6X_j^2 + X_j^4][4X_j(I + X_j^2)]^{-1}, \qquad \text{Padé [2,1]}, \tag{20}$$

and its reciprocal form

$$X_{j+1} = [4X_j(I + X_j^2)][I + 6X_j^2 + X_j^4]^{-1}, \qquad \text{reciprocal of Padé [2,1]}. \tag{21}$$

These iterations are particularly attractive because they combine high convergence order with global convergence properties, making them competitive benchmarks for newly proposed high-order methods.

The numerical termination of iterative schemes is typically governed by residual-based criteria. For MSR iterations, a natural residual is

$$R_j = X_j^2 - B, \tag{22}$$

while for matrix sign iterations, one may consider

$$S_j = X_j^2 - I. \tag{23}$$

Iterations are commonly terminated once $\|R_j\|/\|B\|$ or $\|S_j\|$ falls below a prescribed tolerance. These criteria are closely related to backward error analysis and provide a practical measure of the quality of the computed approximation.

From a computational standpoint, classical iterations differ in per-step cost. The Newton sign iteration (12) requires one matrix inversion or linear solve per step, while the Newton square root iteration (14) involves one inversion and one multiplication by B. The Denman–Beavers scheme (15) requires two inversions per iteration but yields both $B^{1/2}$ and $B^{-1/2}$ simultaneously. Cyclic reduction (16) relies on structured updates and can be advantageous for large-scale or structured matrices; further details can be read in [27,28].

3. An Accelerated Iteration

To further enhance both the accuracy and convergence characteristics of the classical Newton iteration (14), an extra correction stage is given by incorporating a divided-difference approximation, as discussed in [29,30]. This modification yields an improved iterative strategy designed to solve the nonlinear equation in (11) more efficiently. The resulting fixed-point method takes the form

$$\begin{cases} q_j = x_j - \dfrac{125\,g(x_j) - 126\,g(y_j)}{125\,g(x_j) - 251\,g(y_j)}\,\dfrac{g(x_j)}{g'(x_j)}, \\ x_{j+1} = q_j - \dfrac{g(q_j)}{g[q_j, x_j]}, \end{cases} \tag{24}$$

where the auxiliary quantity is defined by $y_j = x_j - \dfrac{g(x_j)}{g'(x_j)}$, and the divided difference is given by $g[a_1, a_2] = (a_1 - a_2)^{-1}(g(a_1) - g(a_2))$.

The construction of scheme (24) serves a dual purpose. On one hand, it increases the asymptotic rate of convergence beyond that of the classical Newton iteration. On the other hand, it is deliberately structured to maintain strong global convergence features, thereby extending its effectiveness to a broader class of nonlinear equations. This combination of accelerated local behavior and enhanced robustness makes the method particularly suitable for challenging nonlinear systems.

The constants in (24) are not ad hoc. One may consider the parameterized family

$$q_j = x_j - \frac{a\,g(x_j) - (a+1)\,g(y_j)}{a\,g(x_j) - (2a+1)\,g(y_j)}\,\frac{g(x_j)}{g'(x_j)}, \qquad x_{j+1} = q_j - \frac{g(q_j)}{g[q_j, x_j]},$$

with $y_j = x_j - \frac{g(x_j)}{g'(x_j)}$. A Taylor expansion around a simple root x^* shows that the method is fourth-order and that the leading error constant is proportional to $1/a$ (in particular, the coefficient in (26) becomes $-(1/a)\,v_2^3$). We set $a = 125$ as a practical compromise: it yields a small leading error constant while keeping all coefficients as moderate integers, which in turn leads to the compact polynomial coefficients in the induced matrix sign iteration (33)–(35). However, the sharp choice of a in this paper leads to larger attraction basins and thus to faster clustering of the eigenvalues in contrast to the competitors of the same order belonging to the Padé family.

Theorem 1. *Let x^* be a simple root of an adequately differentiable function $g : D \subseteq \mathbb{C} \to \mathbb{C}$, that is $g(x^*) = 0$ and $g'(x^*) \neq 0$. Define the iteration map F by $x_{j+1} = F(x_j)$, where F is induced by (24), namely ($y(x) = x - \frac{g(x)}{g'(x)}$):*

$$q(x) = x - \frac{125\,g(x) - 126\,g(y(x))}{125\,g(x) - 251\,g(y(x))}\,\frac{g(x)}{g'(x)}, \qquad F(x) = q(x) - \frac{g(q(x))}{g[q(x), x]}.$$

Assume F is well-defined and four times continuously differentiable on the closed disk $\overline{D}_r = \{x : |x - x^*| \leq r\}$ for some $r > 0$, and set

$$M = \max_{x \in \overline{D}_r} |F^{(4)}(x)|.$$

If the starting value satisfies

$$0 < |x_0 - x^*| < \delta, \qquad \delta = \min\left\{ r,\ \left(\frac{12}{M}\right)^{1/3} \right\}, \tag{25}$$

then the sequence $\{x_j\}$ *generated by (24) is well-defined and converges to* x^*. *Moreover, the following asymptotic error relation holds:*

$$e_{j+1} = \left(-\frac{1}{125}v_2^3\right)e_j^4 + \mathcal{O}\left(e_j^5\right), \tag{26}$$

where $e_j = x_j - x^*$ *and* $v_k = \dfrac{g^{(k)}(x^*)}{k!\,g'(x^*)}$.

Proof. We note that the bound (25) follows from Taylor's theorem applied to the fixed-point map F and guarantees a contractive neighborhood of x^*; the expansion below then yields the stated asymptotic error equation. Since g is holomorphic in a neighborhood of x^* and $g'(x^*) \neq 0$, it admits a Taylor expansion about x^*; substituting these expansions into (24) yields the stated asymptotic error relation. We write

$$g(x_j) = g'(x^*)\left(e_j + v_2e_j^2 + v_3e_j^3 + v_4e_j^4 + O(e_j^5)\right). \tag{27}$$

We also have

$$g'(x_j) = g'(x^*)\left(1 + 2v_2e_j + 3v_3e_j^2 + 4v_4e_j^3 + O(e_j^4)\right). \tag{28}$$

Using (27) and (28), we obtain

$$y_j - x^* = v_2e_j^2 - (2v_2^2 - 2v_3)e_j^3 + \left(4v_2^3 - 7v_3v_2 + 3v_4\right)e_j^4 + O(e_j^5). \tag{29}$$

Inserting (29) into (24) and expanding the resulting fraction, we obtain the error equation

$$\begin{aligned}\frac{125g(x_j) - 126g(y_j)}{125g(x_j) - 251g(y_j)} = 1 + v_2e_j + \left(2v_3 - \frac{124v_2^2}{125}\right)e_j^2 \\ + \left(3v_4 - \frac{3v_2\left(83v_2^2 + 10,250v_3\right)}{15,625}\right)e_j^3 + O(e_j^4).\end{aligned} \tag{30}$$

Utilizing the error term in (30), we have

$$q_j - x^* = -\frac{v_2^2}{125}e_j^3 + \frac{3\left(5333v_2^3 - 5375v_2v_3\right)}{15,625}e_j^4 + O(e_j^5). \tag{31}$$

Now by (30), we have

$$g[q_j, x_j] = g'(x^*) + g'(x^*)v_2e_j + g'(x^*)v_3e_j^2 + \left(g'(x^*)v_4 - \frac{g'(x^*)v_2^3}{125}\right)e_j^3 + O\left(e_j^4\right). \tag{32}$$

This finally yields (26). The proof is complete. □

By applying the iterative construction in (24) to the matrix Equation (11) and extending the formulation in a natural manner to the matrix setting, one obtains a higher-order iterative procedure for computing the matrix sign function. The resulting algorithm takes the form

$$X_{j+1} = X_j\left(314I + 624X_j^2 + 62X_j^4\right)\left(63I + 626X_j^2 + 311X_j^4\right)^{-1}, \tag{33}$$

where the iteration is initiated from a nonsingular matrix,

$$X_0 = B. \tag{34}$$

An equivalent representation can also be obtained by rearranging the factors, yielding the alternative formulation

$$X_{j+1} = \left(63I + 626X_j^2 + 311X_j^4\right)\left[X_j\left(314I + 624X_j^2 + 62X_j^4\right)\right]^{-1}. \tag{35}$$

A comparison between the procedures defined in (33)–(35) and those introduced in (20) and (21) reveals that all of these methods involve the same computational workload per iteration, namely four matrix–matrix products and a single matrix inversion. Consequently, the newly proposed schemes offer a competitive alternative to the corresponding Padé-type methods, achieving comparable computational cost while retaining high-order convergence and global behavior.

To complement the convergence-order analysis, we summarize the dominant per-iteration costs in terms of matrix–matrix multiplications (MM) and linear solves (SOL). Here, one SOL refers to computing $\tilde{M}^{-1}\tilde{N}$ by factorizing $\tilde{M}$ once (e.g., LU) and solving with n right-hand sides (we do not form $\tilde{M}^{-1}$ explicitly). For dense unstructured matrices, a single MM costs about $2n^3$ flops, while one SOL costs about $\frac{8}{3}n^3$ flops (LU factorization plus triangular solves with n right-hand sides). Lower-order operations (additions and scalar multiples) are negligible in this comparison. Results can be seen in Table 1. Although higher-order methods may incur a larger per-step arithmetic count, they typically require substantially fewer iterations.

Table 1. Dominant per-iteration cost (dense $n \times n$ case).

Method	Output	#MM	#SOL	Approx. Cost/n^3
Newton MSR (14)	$B^{1/2}$	0	1	$\frac{8}{3}$
Denman–Beavers (15)	$B^{1/2}$, $B^{-1/2}$	0	2	$\frac{16}{3}$
Cyclic reduction (16)	$B^{1/2}$	1	1	$2+\frac{8}{3}$
Padé [1,2] sign (20) and (21)	$\mathrm{sign}(\cdot)$	3	1	$6+\frac{8}{3}$
Proposed sign (33)–(35)	$\mathrm{sign}(\cdot)$	3	1	$6+\frac{8}{3}$

4. On Convergence and the MSR

A fundamental distinction among iterative algorithms lies in their convergence behavior, particularly in whether convergence is guaranteed globally or only within a local neighborhood of the solution. Several Padé-type iterations and sign-function-based schemes exhibit global convergence (for finding the matrix sign function and subsequently for the MSR), meaning that convergence is achieved for any admissible initial matrix whose spectrum does not intersect the imaginary axis. In contrast, a number of Newton-type approaches possess only local convergence unless supplemented by appropriate stabilization or scaling strategies [31].

To investigate these properties, the notion of attraction basins using the produced fractals is utilized. This concept offers both qualitative and quantitative insight into the global dynamics of an iterative process [32,33]. In practice, a square region of the complex plane, defined as $[-10, 10] \times [-10, 10]$, is discretized into a dense grid of initial points. Each grid point is used as a starting value for the iteration, and its asymptotic behavior is monitored. Points are colored associated to the root to which the solver converges, while those that fail to converge within a prescribed tolerance are marked distinctly in order to see the produced fractals clearly. Convergence is declared when the condition

$$|g(x_j)| \leq 10^{-5}$$

is satisfied. The resulting basin plots, displayed in Figures 1 and 2, encode convergence speed through color intensity. No special fractal behavior can be seen for the quadratic polynomial test to yield in local convergence. The scheme defined by (33) displays broader and lighter-colored attraction regions, indicating faster convergence across a larger portion of the complex plane. This advantage is especially noteworthy given that all compared methods require an equivalent computational cost per iteration in terms of matrix multiplications and inversions. We note that the convergence of the matrix sign computation corresponding to $x^2 - 1 = 0$ is global, but faster for (33).

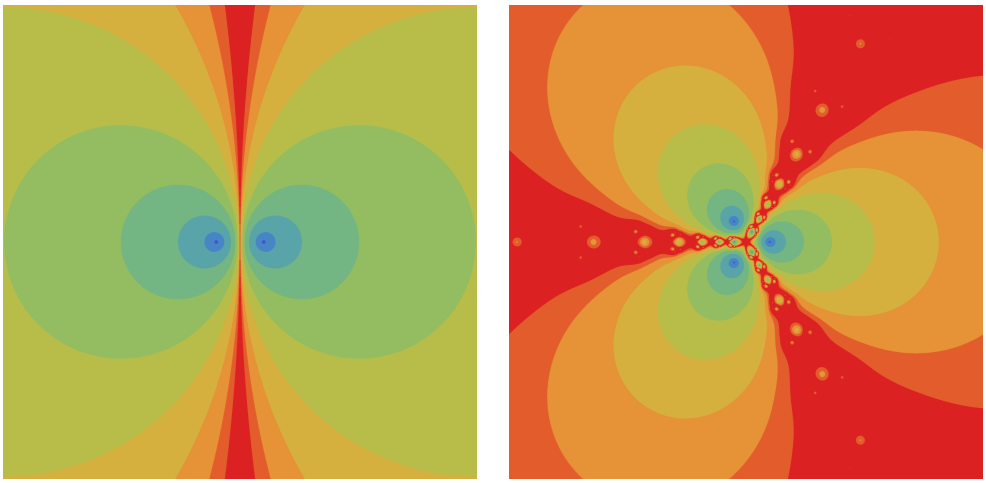

Figure 1. Attraction basins and fractals when computing the sign function for (12) in (**left**) for $x^2 - 1 = 0$ and in (**right**) for $x^3 - 1 = 0$.

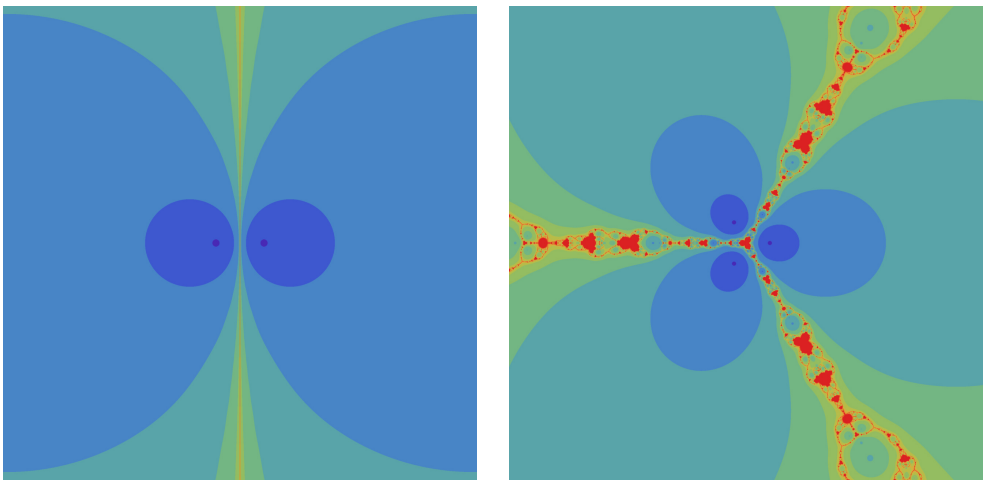

Figure 2. Attraction basins and fractals when computing the sign function for (33) in (**left**) for $x^2 - 1 = 0$ and in (**right**) for $x^3 - 1 = 0$.

For $g(x) = x^2 - 1$, the plots model the spectral dynamics relevant to matrix sign iterations. For additional illustration of the underlying nonlinear root-finder (24), we also include a second test, $g(x) = x^3 - 1$; in that case, the plots should be interpreted purely as basins for scalar root-finding rather than as a sign function computation.

In order to extend the applicability of the iterative procedure defined in (33) to the computation of MSRs, we make use of a fundamental identity previously introduced in (9) and (10). This relationship provides a crucial link between the evaluation of the matrix sign function and the computation of MSRs, allowing the latter problem to be reformulated through an augmented block-matrix representation. In particular, the original

matrix is embedded into a larger matrix $\mathcal{H} \in \mathbb{C}^{2n\times 2n}$, whose sign function encodes the desired square-root information.

A further structural advantage of the proposed framework is its invariance with respect to commutativity, which is crucial for preserving algebraic consistency during the iteration. Let B be an invertible matrix whose spectrum does not intersect the closed negative real axis, so that the principal MSR $B^{1/2}$ and inverse square root $B^{-1/2}$ are well defined. Assume that the initial iterate X_0 commutes with B, that is,

$$BX_0 = X_0B. \tag{36}$$

Suppose that the update map associated with (33) can be written in the rational form

$$X_{j+1} = \Phi(X_j; B), \tag{37}$$

where $\Phi(\cdot; B)$ is constructed from X_j using only matrix addition, multiplication, inversion of intermediate iterates, and multiplication by B (and possibly scalar coefficients). Then, commutativity is preserved by induction: assuming $BX_j = X_jB$, every algebraic term appearing in $\Phi(X_j; B)$ commutes with B, and consequently

$$BX_{j+1} = B\Phi(X_j; B) = \Phi(X_j; B)B = X_{j+1}B. \tag{38}$$

Hence, Equation (36) implies $BX_j = X_jB$ for all $j \geq 0$. This invariance is significant for two reasons. First, commutativity ensures that the iterates evolve within the commutant of B, aligning the iteration with the primary functional calculus in which $B^{1/2}$ is defined; in particular, when X_0 is chosen as a primary function of B, the entire sequence remains within that functional framework. Second, preservation of commutativity prevents the iteration from introducing spurious noncommuting components, which can otherwise amplify rounding errors and complicate the theoretical interpretation of the iterates. Consequently, the commutativity invariance (38) contributes to both the numerical stability and the conceptual consistency of the proposed method when it is employed for MSR computations through the block-sign reformulation.

Corollary 1. *Let $B \in \mathbb{C}^{n\times n}$ satisfy $\sigma(B) \cap \mathbb{R}_{\leq 0} = \varnothing$ and define $\mathcal{H}$ by (9). Apply the sign iteration (33) to $\mathcal{H}$ with $X_0 = \mathcal{H}$. Then $X_j \to \operatorname{sign}(\mathcal{H})$ with fourth-order convergence, and the off-diagonal blocks of X_j converge to $B^{1/2}$ and $B^{-1/2}$, respectively, as in*

$$\operatorname{sign}(\mathcal{H}) = \begin{bmatrix} 0 & B^{1/2} \\ B^{-1/2} & 0 \end{bmatrix}. \tag{39}$$

In particular, the proposed framework computes the principal square root and its inverse simultaneously.

The convergence claims are made for admissible starting matrices, i.e., matrices whose spectra avoid the discontinuity set of the sign function. For the sign iteration (33)–(35), admissibility means

$$\sigma(X_0) \cap i\mathbb{R} = \varnothing, \tag{40}$$

in which case, $\operatorname{sign}(X_0)$ is well-defined and the iteration converges to it. In our computations of $\operatorname{sign}(B)$, we take $X_0 = B$, and the assumption $\sigma(B) \cap i\mathbb{R} = \varnothing$ guarantees (40).

For the principal MSR, we use the block reformulation $\mathcal{H} = \left[\begin{smallmatrix} 0 & B \\ I & 0 \end{smallmatrix}\right]$. If $\sigma(B) \cap \mathbb{R}_{\leq 0} = \varnothing$, then $\sigma(\mathcal{H}) = \{\pm\lambda^{1/2} : \lambda \in \sigma(B)\}$ avoids $i\mathbb{R}$, so $\operatorname{sign}(\mathcal{H})$ is well-defined and encodes $B^{1/2}$ and $B^{-1/2}$. Accordingly, for MSR computations, we initialize with $X_0 = \mathcal{H}$, equivalently with the structured blocks $Y_0 = B$ and $Z_0 = I$ in the off-diagonal form.

In addition, although the iteration is globally convergent on the admissible set, a simple scaling can improve numerical robustness and accelerate convergence. For sign computation, one may replace X_0 by αB with $\alpha > 0$ (e.g., $\alpha = 1/\|B\|_\infty$), since $\operatorname{sign}(\alpha B) = \operatorname{sign}(B)$. For MSR computation, one may scale B by a power of two to bring its norm closer to one: choose an integer k and set $\widetilde{B} = 2^{-2k}B$, compute $\widetilde{B}^{1/2}$, and then recover $B^{1/2} = 2^k\widetilde{B}^{1/2}$; this scaling preserves the principal branch and improves the conditioning of intermediate linear solves.

Quantitative Basin Metrics

To complement the visual basin plots in Figures 1 and 2, we report simple quantitative measures computed on the same uniform grid in the square $[-10, 10] \times [-10, 10]$. Let $\mathcal{G}$ denote the set of all grid points (initial values) and let $\mathcal{G}_{\text{conv}}$ be the subset for which the stopping condition $|g(x_j)| \leq 10^{-5}$ is reached within a prescribed maximum number of iterations $J_{\max}$. For each convergent initial value $x_0 \in \mathcal{G}_{\text{conv}}$, let $N(x_0)$ be the iteration count at termination. We then define:

$$p_{\text{conv}} = \frac{|\mathcal{G}_{\text{conv}}|}{|\mathcal{G}|}, \qquad \overline{N} = \frac{1}{|\mathcal{G}_{\text{conv}}|} \sum_{x_0 \in \mathcal{G}_{\text{conv}}} N(x_0), \qquad N_{\max} = \max_{x_0 \in \mathcal{G}_{\text{conv}}} N(x_0).$$

Here, p_{conv} approximates the relative area of the convergence region in the selected window, while $\overline{N}$ and $N_{\max}$ quantify typical and worst-case speeds of convergence on that region.

In the experiments producing Figures 1 and 2, we used a 2000×2000 grid and set $J_{\max} = 50$. The results in Table 2 show that the proposed method has the same (or essentially the same) convergence region for $g(x) = x^2 - 1$ but achieves a substantially smaller mean iteration count, which quantitatively supports the lighter color intensity observed in the basin plots.

Table 2. Quantitative basin metrics on $[-10, 10] \times [-10, 10]$ (grid 2000×2000, $J_{\max} = 50$, tolerance 10^{-5}).

Method	p_{conv}	$\overline{N}$	$N_{\max}$
Newton sign iteration (12)	1.00	9.6	18
Proposed iteration (24)/(33)	1.00	4.1	8

We stress that these scalar basins serve as a proxy for the spectral dynamics of the corresponding matrix sign iterations: faster scalar convergence translates into faster clustering of eigenvalues to ± 1, which is consistent with the improved iteration counts observed in the matrix experiments of Section 6.

5. Convergence Analysis

Theorem 2. *Let $B \in \mathbb{C}^{n\times n}$ be a nonsingular matrix, that is, $\sigma(B) \cap i\mathbb{R} = \varnothing$. Assume further that the initial iterate X_0 commutes with B, i.e., $BX_0 = X_0B$. Then, the iterative scheme defined in (35), which is algebraically equivalent to (33), converges to the matrix sign function with fourth-order convergence.*

Proof. Since $\sigma(B) \cap i\mathbb{R} = \varnothing$, the matrix sign function $\operatorname{sign}(B)$ is well-defined and has eigenvalues in $\{+1, -1\}$. Furthermore, by the Jordan canonical decomposition, there exists a nonsingular matrix T such that

$$B = TJT^{-1},$$

where J is a block-diagonal matrix whose blocks are Jordan blocks associated with the eigenvalues of B. Because the iteration is constructed from rational functions of B and its iterates, the analysis may be carried out at the level of each Jordan block independently. In particular, the behavior of the matrix iteration is governed by the scalar iteration induced on the eigenvalues of B.

Let β_j denote an eigenvalue of the iterate X_j corresponding to a fixed eigenvalue of B. The iterative scheme (35) then induces a scalar recurrence of the form

$$\beta_{j+1} = \frac{63 + 626\beta_j^2 + 311\beta_j^4}{\beta_j\left(314 + 624\beta_j^2 + 62\beta_j^4\right)}, \tag{41}$$

which governs the local dynamics of the iteration. The fixed points of this mapping are $\beta = \pm 1$, corresponding to the eigenvalues of the matrix sign function. Define the Cayley-type scalar error transform

$$t_j = \frac{\beta_j - s}{\beta_j + s}, \qquad s = \operatorname{sign}(\Re(\beta_0)) \in \{+1, -1\}.$$

Then, $|t_0| < 1$ is equivalent to $\Re(s\,\beta_0) > 0$, which holds whenever $\Re(\beta_0) \neq 0$. For the scalar map induced by (35), one obtains the closed recursion

$$t_{j+1} = t_j^4\,\frac{125\,t_j - 1}{t_j - 125}.$$

Since the Möbius factor $\frac{125t-1}{t-125} = \frac{t-\frac{1}{125}}{1-\frac{1}{125}t}$ maps the unit disk into itself, it follows that $|t_0| < 1$ implies $|t_{j+1}| \leq |t_j|^4$ and hence $t_j \to 0$. Consequently, $\beta_j \to s \in \{+1, -1\}$ for every eigenvalue with $\Re(\beta_0) \neq 0$, i.e., for every admissible initial eigenvalue. To determine the order of convergence, define

$$D_j = 314I + 624X_j^2 + 62X_j^4,$$

so that the iteration may be rewritten in the compact form

$$X_{j+1} - \Omega = \left(63I + 626X_j^2 + 311X_j^4 - \Omega D_j\right)D_j^{-1}, \tag{42}$$

where $\Omega = \operatorname{sign}(B)$. By algebraic manipulation and repeated use of the identity $\Omega^2 = I$, this expression can be factorized as

$$X_{j+1} - \Omega = (X_j - \Omega)^4\,[-63I + 62X_j\Omega]\,D_j^{-1}.$$

Taking norms on both sides yields

$$\|X_{j+1} - \Omega\| \leq \|D_j^{-1}\|\,\|-63I + 62X_j\Omega\|\,\|X_j - \Omega\|^4, \tag{43}$$

which clearly establishes fourth-order convergence of the iteration, provided that X_j remains sufficiently close to Ω and that D_j remains nonsingular. It is worth emphasizing that although the eigenvalues of the matrix sign function are confined to $\{\pm 1\}$, the operator norm of $\operatorname{sign}(B)$ need not be small, especially for highly nonnormal matrices. This completes the proof. □

6. Computational Results

This section is devoted to a computational investigation of the performance and convergence behavior of the iterative methods discussed in the preceding sections for computing MSRs and their inverses. The computational tests were performed in the Mathematica 14.0 environment [34], ensuring consistent numerical precision and a uniform computational framework across all test cases. The objective of these experiments is to assess the practical effectiveness of the presented algorithm relative to several well-established schemes from the literature.

The numerical comparison involves the following iterative methods: the classical Newton iteration (NS) (14), the Denman–Beavers (DS) scheme (15), the cyclic reduction (MS) method (16), the Padé-based (PS) methods (20) and the proposed approach (PM) (33). All algorithms are implemented using the same arithmetic precision and identical stopping criteria to ensure a fair and meaningful comparison.

To quantify convergence, we employ the relative error measure

$$E_{j+1} = \frac{\|Y_{j+1} - Y_j\|_\infty}{\|Y_{j+1}\|_\infty}, \tag{44}$$

where $\|\cdot\|_\infty$ denotes the matrix infinity norm. This criterion is commonly used in the numerical analysis of matrix iterations, as it provides a reliable indicator of stabilization of successive iterates. Iterations are terminated once the prescribed tolerance ε is satisfied, thereby ensuring comparable termination conditions across all algorithms. The numerical experiments reported below are not restricted to a single problem instance, but rather comprise a representative set of test matrices of increasing dimension.

Example 1. *We consider a structured symmetric positive definite matrix of dimension n, defined by*

$$B = \left(b_{ij}\right)_{n\times n}, \qquad b_{ij} = \begin{cases} 10, & i = j, \\ -2.8, & |i-j| = 1, \\ -1.5, & |i-j| = 2, \\ -0.5, & |i-j| = 3, \\ 0, & \textit{otherwise}. \end{cases}$$

This banded symmetric positive definite test matrix is constructed for the present study as a scalable benchmark: its Toeplitz-like structure allows varying n, while the strong diagonal dominance ensures positive definiteness and makes it suitable for assessing iteration speed and numerical stability across problem sizes.

For each method under consideration, the iteration is initialized using the same admissible starting matrix, and convergence is monitored using the criterion (44). The numerical results, summarized in the subsequent tables, demonstrate the relative efficiency of the proposed approach in terms of iteration count and convergence rate, while confirming its robustness across varying problem dimensions.

The numerical results obtained from this test case are compared in Figures 3 and 4, where the stopping tolerance is fixed. Noting that in Figure 3, Equation (14) coincides with Equation (15). An examination of the computational outcomes reveals that the proposed solver (33) delivers better performance relative to the other methods. In all tested scenarios, these schemes attain convergence in fewer iterations and require less overall computational effort to satisfy the prescribed accuracy criterion.

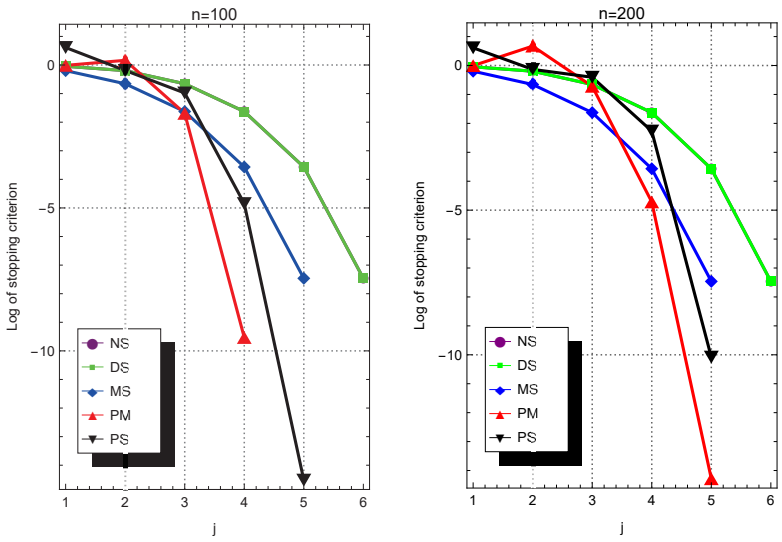

Figure 3. Results of comparisons for various sizes when the tolerance is $\varepsilon = 10^{-6}$.

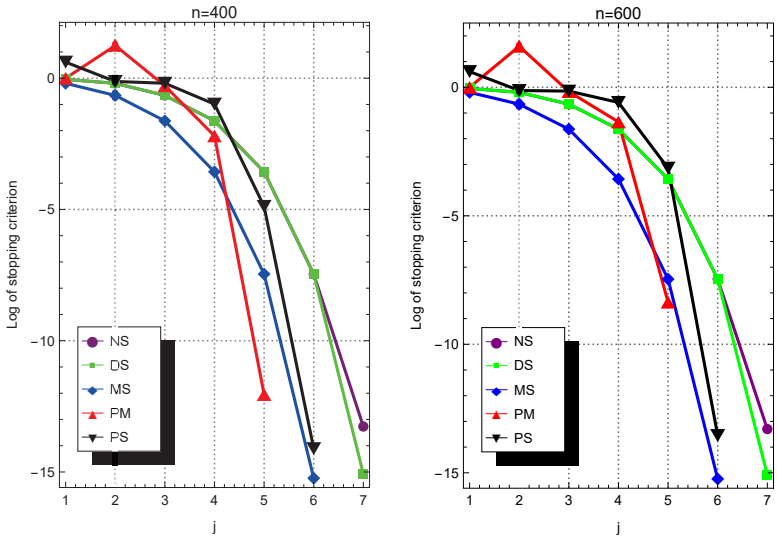

Figure 4. Results of comparisons for various sizes when the tolerance is $\varepsilon = 10^{-8}$.

Stability and Conditioning Diagnostics

To substantiate the remarks on sensitivity and stability, we include an additional test, in which an eigenvalue approaches the branch cut of the principal square root. For the proposed method, we monitor both a backward residual and a forward error (against an exact reference available for this constructed family). Specifically, for $Y \approx B^{1/2}$, we report

$$\eta = \frac{\|Y^2 - B\|_\infty}{\|B\|_\infty}, \qquad \mathrm{FE} = \frac{\|Y - Y_\star\|_\infty}{\|Y_\star\|_\infty}, \tag{45}$$

where $Y_\star$ denotes the exact principal square root for the considered example. We also report the consistency defect $\|I - YZ\|_\infty$ when the method simultaneously computes an approximation $Z \approx B^{-1/2}$.

Example 2. *(Near-branch-cut and nonnormal diagonalizable family.) Let $n = 10$ and $\delta > 0$. Define*

$$\Lambda(\delta) = \mathrm{diag}\big(-1 + i\delta,\ 2,\ 3, \ldots,\ n\big), \qquad V = I + \gamma U,\ \ \gamma = 5,$$

where $U \in \mathbb{R}^{n\times n}$ is the strictly upper shift matrix with $U_{k,k+1} = 1$ and all other entries zero. Set

$$B(\delta) = V\,\Lambda(\delta)\,V^{-1}.$$

Since $\sigma(B(\delta)) = \sigma(\Lambda(\delta))$ and $\sigma(B(\delta)) \cap \mathbb{R}_{\le 0} = \varnothing$ for $\delta > 0$, the principal square root exists. Moreover, because $B(\delta)$ is diagonalizable, the exact principal square root is

$$Y_\star(\delta) = V\,\Lambda(\delta)^{1/2}\,V^{-1}, \qquad \Lambda(\delta)^{1/2} = \mathrm{diag}\big(\sqrt{-1+i\delta},\ \sqrt{2},\ldots,\sqrt{n}\big),$$

where $\sqrt{\cdot}$ denotes the principal scalar branch. We apply the proposed sign-based framework for MSR computation (producing $(Y, Z) \approx (B^{1/2}, B^{-1/2})$) with the tolerance $\varepsilon = 10^{-8}$ in double precision and report η, $\|I - YZ\|_\infty$, and FE.

Table 3 shows that as $\delta \downarrow 0$ (approaching the branch cut), the proposed method remains backward stable (very small η) and maintains high accuracy with respect to the exact principal square root, while the iteration count increases moderately. The forward error is observed to saturate around the level expected in double precision, which is consistent with the intrinsic sensitivity of the square-root map near the branch cut and the effect of rounding errors in repeated matrix operations.

Table 3. Stability diagnostics for the proposed method on $B(\delta)$ in Example 2 ($n = 10$, $\gamma = 5$, $\varepsilon = 10^{-8}$).

δ	Iter.	$\eta = \|Y^2 - B\|_\infty / \|B\|_\infty$	$\|I - YZ\|_\infty$	FE $= \|Y - Y_\star\|_\infty / \|Y_\star\|_\infty$
10^{-2}	6	4.45×10^{-15}	4.63×10^{-10}	2.42×10^{-15}
10^{-4}	9	9.44×10^{-15}	4.10×10^{-10}	4.98×10^{-15}
10^{-6}	12	3.76×10^{-15}	6.25×10^{-10}	2.24×10^{-15}
10^{-8}	15	4.59×10^{-15}	4.97×10^{-10}	2.73×10^{-15}

7. Ending Notes

In this manuscript, we have investigated the problem of calculating MSRs through the lens of matrix function theory and iterative numerical analysis. By exploring the intrinsic relationship between the MSR and the matrix sign function, we developed a unified framework that allows the construction of efficient high-order iterative schemes based on rational approximations and block-matrix representations. A detailed convergence analysis was presented, demonstrating that the proposed iteration attains fourth-order convergence. In addition, the formulation naturally accommodates the simultaneous calculation of the square root and its inverse, offering a computational advantage in applications where both quantities are required. Numerical experiments confirmed the theoretical findings and showed that the proposed method outperforms several classical schemes, including existing standard iterations, in terms of convergence speed and robustness.

Future research directions include extending the proposed framework to other matrix functions, such as fractional powers and logarithms, and exploring structure-preserving variants for large-scale and sparse problems. The integration of adaptive strategies and parallel implementations also represents a promising avenue for further investigation.

Author Contributions: Conceptualization, S.W. and T.L. (Tao Liu); methodology, S.W., Z.Y., S.S. and T.L. (Taher Lotfi); software, Z.Y. and W.K.; validation, X.H. and Y.L.; formal analysis, S.S. and T.L. (Taher Lotfi); investigation, S.W. and S.S.; resources, X.H.; data curation, Y.L. and W.K.; writing—original draft preparation, S.W. and Z.Y.; writing—review and editing, S.S., T.L. (Taher Lotfi) and T.L. (Tao Liu); visualization, T.L. (Taher Lotfi); supervision, T.L. (Tao Liu); project administration, T.L. (Tao Liu); and funding acquisition, T.L. (Tao Liu). 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.

Acknowledgments: The authors would like to thank all three referees for their constructive suggestions relating to this work.

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. Kenney, C.S.; Laub, A.J. The matrix sign function. *IEEE Trans. Autom. Control* **1995**, *40*, 1330–1348. [CrossRef]
3. Higham, N.J.; Liu, X. A multiprecision derivative-free Schur–Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.* **2021**, *42*, 1401–1422. [CrossRef]
4. Smith, M. A Schur algorithm for computing matrix square roots. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 971–989. [CrossRef]
5. Parlett, B.N. *The Symmetric Eigenvalue Problem*; SIAM: Philadelphia, PA, USA, 1998.
6. Iannazzo, B. On the Newton method for the matrix *p*th root. *SIAM J. Matrix Anal. Appl.* **2006**, *28*, 503–523. [CrossRef]
7. Cross, G.W.; Lancaster, P. Square roots of complex matrices. *Lin. Multilinear Alg.* **1974**, *1*, 289–293. [CrossRef]
8. Go, S.; Pan, V.Y.; Soto, P. Root-Squaring for Root-Finding. In *Computer Algebra in Scientific Computing. CASC 2023*; Boulier, F., England, M., Kotsireas, I., Sadykov, T.M., Vorozhtsov, E.V., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 14139.
9. Hauenstein, J.D.; Pan, V.Y.; Szanto, A. Global Newton iteration over Archimedianand non-Archimedianfields. In *Computer Algebra in Scientific Computing. CASC 2014*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014.
10. Cardoso, J.R. Evaluating the Fréchet derivative of the matrix *p*th root. *Electron. Trans. Numer. Anal.* **2011**, *38*, 202–217.
11. Soleymani, F.; Shateyi, S.; Haghani, F.K. A numerical method for computing the principal square root of a matrix. *Abst. Appl. Anal.* **2014**, *2014*, 525087. [CrossRef]
12. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: New York, NY, USA, 1964.
13. Abdullah, S.; Ganie, A.; Alharthi, N. A new parameterized iterative scheme for nonlinear equations: Convergence and basin of attraction analysis. *J. Appl. Math. Comput.* **2025**, *71*, 7929–7958. [CrossRef]
14. Yilmaz, N. Introducing three new smoothing functions: Analysis on smoothing-Newton algorithms. *J. Math. Model.* **2024**, *12*, 463–479.
15. Soleymani, F.; Haghani, F.K.; Shateyi, S. Several numerical methods for computing unitary polar factor of a matrix. *Adv. Differ. Equ.* **2016**, *2016*, 4. [CrossRef]
16. Abdullah, S.; Choubey, N.; Dara, S. Optimal fourth- and eighth-order iterative methods for solving nonlinear equations with basins of attraction. *J. Appl. Math. Comput.* **2024**, *70*, 3477–3507. [CrossRef]
17. Feng, Y.; Othman, A.Z. An accelerated iterative method to find the sign of a nonsingular matrix with quartical convergence. *Iran. J. Sci.* **2023**, *47*, 1359–1366. [CrossRef]
18. 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. [CrossRef]
19. Higham, N.J. Newton's method for the matrix square root. *Math. Comput.* **1986**, *46*, 537–549. [CrossRef]
20. Lu, D.; Guo, C.-H. Explicit *p*-dependent convergence regions of Newton's method for the matrix *p*th root. *Appl. Math. Lett.* **2021**, *122*, 107566. [CrossRef]
21. 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]
22. Denman, E.D.; Beavers, A.N. The matrix sign function and computations in systems. *Appl. Math. Comput.* **1976**, *2*, 63–94. [CrossRef]
23. 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.
24. Meini, B. The matrix square root from a new functional perspective: Theory and algorithms. *SIAM J. Matrix Anal. Appl.* **2004**, *26*, 362–376. [CrossRef]

25. Kenney, C.S.; Laub, A.J. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.* **1991**, *12*, 273–291. [CrossRef]
26. Rani, L.; Kansal, M. Numerically stable iterative methods for computing matrix sign function. *Math. Meth. Appl. Sci.* **2023**, *46*, 8596–8617. [CrossRef]
27. Björck, A.; Hammarling, S. A Schur method for the square root of a matrix. *Linear Algebra Appl.* **1983**, *52–53*, 127–140. [CrossRef]
28. Iannazzo, B. A family of rational iterations and its application to the computation of the matrix *p*th root. *SIAM J. Matrix Anal. Appl.* **2009**, *30*, 1445–1462. [CrossRef]
29. Arora, H.; Cordero, A.; Torregrosa, J.R. On generalized one-step derivative-free iterative family for evaluating multiple roots. *Iran. J. Numer. Anal. Optim.* **2024**, *14*, 291–314.
30. Soleymani, F. A three-step iterative method for nonlinear systems with sixth order of convergence. *Int. J. Comput. Sci. Math.* **2013**, *4*, 363–373. [CrossRef]
31. Ullah, M.Z.; Alaslani, S.M.; Mallawi, F.O.; Ahmad, F.; Shateyi, S.; Asma, M. A fast and efficient Newton-type iterative scheme to find the sign of a matrix. *AIMS Math.* **2023**, *8*, 19264–19274. [CrossRef]
32. Golzarpoor, J.; Ahmed, D.; Shateyi, S. Constructing a matrix mid-point iterative method for matrix square roots and applications. *Mathematics* **2022**, *10*, 2200. [CrossRef]
33. Adullah, S.; Choubey, N.; Dara, S. A new three-step optimal without memory iterative scheme for solving non-linear equations with basins of attraction. *Comput. Methods Differ. Equ.* **2025**, *13*, 1289–1305.
34. Trott, M. *The Mathematica Guidebook for Numerics*; Springer: New York, NY, USA, 2006.