mathematics

Article

Constructing a Matrix Mid-Point Iterative Method for Matrix Square Roots and Applications

Javad Golzarpoor [1], **Dilan Ahmed** [2,3] and **Stanford Shateyi** [4,*]

[1] Department of Science, School of Mathematical Sciences, University of Zabol, Zabol 98613-35856, Iran; javad_golzarpoor@uoz.ac.ir
[2] Department of Mathematics, College of Education, University of Sulaimani, Kurdistan Region, Sulaimani 46001, Iraq; dilan.ahmed@univsul.edu.iq
[3] Department of Petroleum Engineering, Komar University of Science and Technology, Kurdistan Region, Sulaymaniyah 46013, Iraq
[4] Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, P. Bag X5050, Thohoyandou 0950, South Africa
* Correspondence: stanford.shateyi@univen.ac.za

Abstract: In this paper, an improvement to the mid-point method is contributed for finding the square root of a matrix as well as its inverse. To this aim, an iteration scheme to find this matrix function is constructed, and its error and stability estimates are provided to show the theoretical rate of convergence. Our higher-order method can compete with the existing iterative methods of a similar nature. This is illustrated in numerical simulations of various sizes.

Keywords: iterative method; matrix square root; higher order; convergence analysis; matrix functions

MSC: 65F60; 41A25

check for updates

Citation: Golzarpoor, J.; Ahmed, D.; Shateyi, S. Constructing a Matrix Mid-Point Iterative Method for Matrix Square Roots and Applications. *Mathematics* **2022**, *10*, 2200. https://doi.org/10.3390/math10132200

Academic Editors: Juan Zhang and Jianzhou Liu

Received: 21 May 2022
Accepted: 19 June 2022
Published: 24 June 2022

1. Introductory Notes

A matrix function is a type of function that obtains matrices (mostly square ones) and maps them into matrices of the same dimensions. There are several definitions in the literature ([1], Chapter 1) for defining the functions of matrices, among which the Jordan canonical form, Cauchy integral, as well as a definition based on the Hermite matrix polynomials are of special interest.

Apart from such standard procedures to compute matrix functions, iterative methods are significant as long as it is required to compute them when the entries are changing over time or when a sharp initial approximation is available in some applications (for more information, see [2,3]).

In this work, the application of iteration methods for finding matrix square roots is considered and discussed in detail. One of the motivations for this is the following fact. Assume that one has a matrix differential equation as follows:

$$\begin{cases} v''(t) + Mv(t) = 0, \\ v(0) = v_0, \quad v'(0) = v_0'(0), \end{cases} \tag{1}$$

where $v(t)$ is a real value function and the exact solution can be obtained as follows (by considering $\sqrt{M}$ as any square roots of M):

$$v(t) = \cos(\sqrt{M}t)v_0(0) + M^{-1/2}\sin(\sqrt{M}t)v_0'(0). \tag{2}$$

As can be observed from Equation (2), both the principal square root and its inverse are required to find the numerical solution. Note that $M \in \mathbb{C}^{n\times n}$ has a square root if no two terms are the same odd integer in the ascent sequence of integers $t_1, t_2, \cdots$, given in [4]:

$$t_i = \dim(\text{null}(M^i)) - \dim(\text{null}(M^{i-1})). \tag{3}$$

The rich variety of iteration schemes for calculating the square root of a matrix (see, for example, [5] and the references cited) with their widely differing computational stability features are a challenging topic of investigation in their own right. As an illustration, an application of the matrix square root is in the solution to the algebraic Riccati matrix equations, as discussed recently in [6].

This paper proposes a new higher-order scheme for calculating the square root and its associated inverse of a suitable matrix. It is illustrated and proven that under some conditions, the solver converges globally to the square root. The method converges with the fourth order of convergence to the matrix square root.

It is necessary to recall some important facts about the matrix sign function, which has a clear relationship with the matrix square root. A significant fact is that although $\text{sign}(M)$ is the unit matrix square root (I), it is unequal to $-I$ or I unless the spectrum of M is totally in the open right or left half-plane(s), respectively [7]. Therefore, $\text{sign}(M)$ is a non-primary square root of I. This function is defined via the Cauchy integral as follows:

$$\text{sign}(M) = S = \frac{2}{\pi}\int_0^\infty (t^2 I + M^2)^{-1} dt. \tag{4}$$

It should also be noted that a variety of mathematical problems which are not related to nonlinear equations at first sight can be rewritten to find the solutions to nonlinear equations in special spaces (e.g., in operator form). As an illustration, solving nonlinear stochastic differential equations [8,9] with semi-analytic methods can be accomplished with Chaplygin-type solvers, which are in fact part of Newton's method in some operator formats (for more information, see [10]).

Here, iteration schemes are the major concentration. In fact, such iterative matrix methods considering an appropriate initial matrix are fixed point-type methods or Newton-type schemes that produce a convergent sequence of matrices. The relation between the iteration schemes for the sign matrix and the root-finding methods is unclear at first sight, but many sign finders are derived via the application of root solvers to the following nonlinear equation:

$$g(x) = x^2 - 1 = 0, \tag{5}$$

where $g : D \subseteq \mathbb{C} \to \mathbb{C}$ is a scalar function.

The remaining structure of this article is as follows. Section 2 furnishes some of the existing iterative methods of various orders to compute the matrix sign and the square root. The proposed method is required mainly in the context of symmetric positive definite matrices. Section 3 proposes a mid-point-type method for finding the square root and its inverse. The relation to the matrix sign as well as its global convergence are illustrated. Then, the convergence of the proposed scheme is given. The convergence rate along with its asymptotical stability are brought forward in detail. The applicability of the iterative methods as well as its reciprocal for finding the square roots of matrices are illustrated and tested in Section 4. To establish the efficiency of the new scheme, we employ numerical experiments of arbitrary dimensions. The numerical results uphold the analytical discussion of this article. Finally, in Section 5, the conclusions are drawn.

2. Existing Iterative Methods

Applying the classic Newton scheme to Equation (5) yields the following scheme for the matrix sign [11]:

$$H_{k+1} = \frac{1}{2}\left(H_k + H_k^{-1}\right). \tag{6}$$

Here, the operator H is used to show the iterates in the iterative method. In a similar manner, the Newton scheme for the square root of a matrix can be written as follows:

$$H_{k+1} = \frac{1}{2}[H_k + MH_k^{-1}], \tag{7}$$

with an appropriate initial matrix H_0. The scheme in Equation (7) suffers from instability as long as the initial matrix is not chosen well or the input matrix has some special structures.

To have stability and higher rates, several methods have been proposed in the literature. A second-order method based on the Newton method (7) is given as follows [12]:

$$\begin{cases} Z_0 = I, \;\; H_0 = M, \;\; k = 0, 1, \cdots, \\ H_{k+1} = \frac{1}{2}[H_k + Z_k^{-1}], \\ Z_{k+1} = \frac{1}{2}[Z_k + H_k^{-1}]. \end{cases} \tag{8}$$

The scheme in Equation (8) is stable and produces $\{H_k\}$ that converges to $M^{1/2}$. Another well-known iteration is due to the cyclic reduction algorithm, which is expressed as follows [13]:

$$\begin{cases} Z_0 = 2(I + M), \;\; H_0 = I - M, \;\; k = 0, 1, \cdots, \\ H_{k+1} = -H_k Z_k^{-1} H_k, \\ Z_{k+1} = Z_k - 2H_k Z_k^{-1} H_k. \end{cases} \tag{9}$$

The sequence $\{Z_k\}$ tends toward $4M^{1/2}$ in the convergence phase and calculates only $M^{1/2}$.

3. Improved Mid-Point Method for the Matrix Square Root

3.1. An Improved Mid-Point Method

It is well known that the mid-point solver has the cubical convergence rate to solve nonlinear equations, which is an improvement over the well-known Newton's method. Adding one extra sub-step via the divided difference approximation can yield the following improved mid-point type method for solving Equation (5):

$$\begin{cases} y_k = t_k - \frac{1}{2}\frac{g(t_k)}{g'(t_k)}, \\ h_k = t_k - \frac{g(t_k)}{g'(y_k)}, \\ t_{k+1} = h_k - \frac{g(h_k)}{g[h_k, y_k]}, \qquad t_0 \geq 0, \end{cases} \tag{10}$$

where $g[a, b] := (a - b)^{-1}(g(a) - g(b))$.

Noting that the iterative solver in Equation (10) does not satisfy the Kung–Traub conjecture regarding the construction of iterative schemes without memory to solve equations [14], but it has an important feature. As a matter of fact, if we pursue the optimality of the iterative solvers for nonlinear equations, then we lose the global convergence behavior in solving Equation (5), which clearly limits the matrix application of such solvers. Hence, Equation (10) was designed to be not only an improvement over the mid-point method and possess a higher rate of convergence but also to reach global convergence behavior:

Theorem 1. *Assume that $x^* \in D$ is a simple root of the smooth enough nonlinear function $g : D \subseteq \mathbb{C} \to \mathbb{C}$. If the starting value t_0 is close enough to the root, then the iterative method (Equation (10)) has the following error equation:*

$$e_{k+1} = \frac{d_2^3}{2} - \frac{d_2 d_3}{8} e_k^4 + \mathcal{O}(e_k^5), \tag{11}$$

where $e_k = t_k - x^*$ and $d_j = \frac{g^{(j)}(x^*)}{j! g'(x^*)}$.

Proof. The proof is based on Taylor expansions and is straightforward. Hence, it has been omitted. □

3.2. Construction for the Sign Function

Now, by using Equation (10) to solve the problem in Equation (5) and a naive extension to the matrix environment, one may construct a higher-order matrix iterative scheme to calculate the sign as follows:

$$H_{k+1} = H_k\left(7I + 22H_k^2 + 3H_k^4\right)\left[I + 18H_k^2 + 13H_k^4\right]^{-1}, \tag{12}$$

with the starting value given for a non-singular square complex matrix as follows:

$$H_0 = M. \tag{13}$$

It is noteworthy that, in a similar manner, the reciprocal form can be used and written as follows:

$$H_{k+1} = \left(I + 18H_k^2 + 13H_k^4\right)\left[H_k\left(7I + 22H_k^2 + 3H_k^4\right)\right]^{-1}. \tag{14}$$

3.3. Competitors from the Padé Family

An optimal family of solvers to compute the sign function was given in [15] based on the Padé approximants to $g(\zeta) = 1/\sqrt{(1-\zeta)}$ by considering $\zeta = 1 - z^2$ as follows:

$$\text{sign}(z) = s = \frac{z}{(z^2)^{1/2}} = \frac{z}{(1-\zeta)^{1/2}}. \tag{15}$$

If we consider $m + n \geq 1$ and write the $[m, n]$ approximant of Padé for $g(\xi)$ as $\frac{P_{m,n}(\xi)}{Q_{m,n}(\xi)}$, then it is possible to have

$$z_{k+1} = \frac{z_k P_{m,n}(1 - z_k^2)}{Q_{m,n}(1 - z_k^2)} := \varphi_{2m+1,2n}, \tag{16}$$

which tends toward ±1 while having a convergence rate of $1 + m + n$. Here, P and Q are the numerator and denominator of the Padé approximant, respectively.

In general, the family in Equation (16) (or its reciprocal), generated by the $[m/m]$ and $[(m-1)/m]$ Padé approximants, are convergent globally, and their order depends on k.

Some fourth-order methods from this family possessing global convergence can be written as follows:

$$H_{k+1} = [I + 6H_k^2 + H_k^4][4H_k(I + H_k^2)]^{-1}, \qquad \text{Padé [1,2]} \tag{17}$$

$$H_{k+1} = [4H_k(I + H_k^2)][I + 6H_k^2 + H_k^4]^{-1}. \qquad \text{Reciprocal of Padé [1,2]} \tag{18}$$

It is remarked that the convergence rate is not the only important point in an iterative scheme; the cost per iteration is also important for determining the total cost of the method. By comparing Equations (12) and (14) to Equations (17) and (18), it is clear that all the methods need four matrix-matrix multiplications and only one matrix inverse per cycle, which means that Equations (12)–(14) are competitors to Padé variants of the same order with global convergence.

3.4. Global Convergence

Investigating the global convergence for the proposed scheme is necessary, since no iterative method for calculating the matrix sign or square root is useful if it lacks the global convergence (see [16] for more information).

Here, an efficient way to ensure such a global convergence is to employ the theory of attraction basins to check the convergence type. To this aim, a square in the complex plane is considered where $[-2,2]\times[-2,2]\in\mathbb{C}$. Then, this area is divided by a mesh at which each point of the mesh is colored based on the root it converges. Note that it is colored black in the case of divergence. The stopping criterion for the iterative methods in this subsection is $|g(x_k)|\le 10^{-2}$. The results are given in Figures 1–3 and are shaded based on the number of iterations. Some competitors from the Padé family are also plotted to show the local convergence in contrast to the global convergence.

Here, although the iterative methods in Equations (7), (14), and (17) are globally convergent, the basins for Equation (14) contains lighter areas, which shows it can arrive in the convergence phase faster than its Padé competitor of the fourth order of convergence when the cost (i.e., the number of matrix-by-matrix multiplications) and inverse per cycle are the same. This superiority of the proposed scheme ensures its usefulness in solving practical problems.

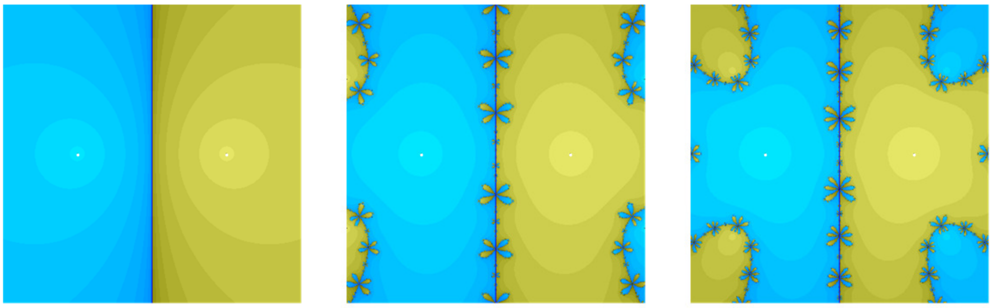

Figure 1. Attraction basins for Equation (7) on the left, Padé 0–3 in middle and Padé 0–4 to the right.

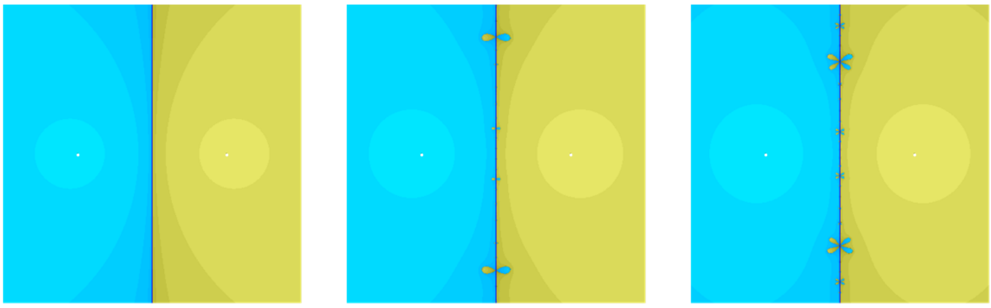

Figure 2. Attraction basins for Padé 1–2 on the left, Padé 1–3 in the middle and Padé 1–4 to the right.

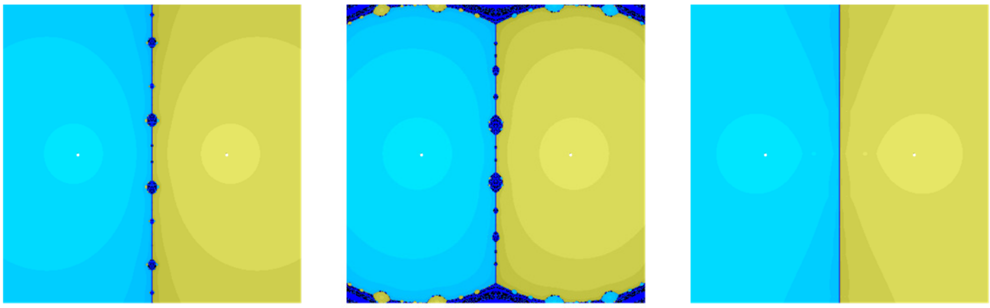

Figure 3. Attraction basins for Padé 2–1 on the left, Padé 3–1 in the middle and (14) to the right.

3.5. Application to the Matrix Square Root

In order to use Equation (12) for our main target, an identity is used as follows [1]:

$$\mathrm{sign}(\mathbf{M}) = \mathrm{sign}\left(\begin{bmatrix} 0 & M \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & M^{1/2} \\ M^{-1/2} & 0 \end{bmatrix}, \tag{19}$$

which is actually the main connection between the matrix sign function and the matrix square root. Here, **M** is a $2n \times 2n$ matrix.

It is also mentioned that for an invertible matrix M without any eigenvalues on $\mathbb{R}^-$, and using the proposed iteration(s), if $MH_0 = H_0M$, then we have the fact that all the subsequent iterates commune with the original matrix M.

3.6. Error Estimate

Theorem 2. *If M is an invertible matrix without any eigenvalues on $\mathbb{R}^-$, and H_0 is sufficiently close to $M^{1/2}$ that it commutes with M, then the scheme in Equation (14) (or equivalently Equation (12)) converges to $M^{1/2}$, and this convergence is of the fourth order.*

Proof. Note that all diagonalizable or non-diagonalizable matrices have a Jordan canonical decomposition form $M = TJT^{-1}$, where the matrix J includes the Jordan blocks. Using this fact and standard Jordan decomposition, one may find the following relation, which is valid for the eigenvalues of the iterates for the steps from k to $k+1$:

$$\begin{aligned}\lambda^i_{k+1} = &\left(1 + 18{\lambda^i_k}^2 + 13{\lambda^i_k}^4\right)\\ &\times \left[\lambda^i_k\left(7 + 22{\lambda^i_k}^2 + 3{\lambda^i_k}^4\right)\right]^{-1}, \quad 1 \leq i \leq 2n,\end{aligned} \tag{20}$$

wherein $s_i = \pm 1$. The expression in Equation (20) reveals that, in general, the eigenvalues are convergent to $s_i = \pm 1$. This means that

$$\lim_{k\to\infty}\left|\frac{\lambda^i_{k+1} - s_i}{\lambda^i_{k+1} + s_i}\right| = 0. \tag{21}$$

After the convergence by Equation (21), and in order to find the convergence rate, we assume that $D_k = H_k(7I + Y_k)(I + 3Y_k)$. Thus, we can write

$$\begin{aligned} H_{k+1} - S &= (I + 18H_k^2 + 13H_k^4)D_k^{-1} - S\\ &= [I + 18H_k^2 + 13H_k^4 - SD_k]D_k^{-1}\\ &= [I + 18H_k^2 + 13H_k^4 - 7H_kS - 22H_k^3S - 3H_k^5S]D_k^{-1}\\ &= [(H_k - S)^4 - 3H_k^5S^5 + 12H_k^4S^4 - 18H_k^3S + 12H_k^2S^2 - H_kS^3]D_k^{-1}\\ &= [(H_k - S)^4 - 3H_kS\big(H^4 - 4H_k^3S + 6H_k^2S^2 - 4H_kS^3 + I\big)]D_k^{-1}\\ &= [(H_k - S)^4 - 3H_kS(H_k - S)^4]D_k^{-1}\\ &= (H_k - S)^4[I - 3H_kS]D_k^{-1}. \end{aligned} \tag{22}$$

It is now easy to take the appropriate norm from Equation (22) and find that

$$\|H_{k+1} - S\| \leq \left(\|D_k^{-1}\|\|I - 3H_kS\|\right)\|H_k - S\|^4, \tag{23}$$

which shows a convergence rate of four for Equation (14). Note that the norm of $\text{sign}(A)$ can be large arbitrarily, though its eigenvalues are ± 1. This completes the proof. □

3.7. Asymptotical Stability

The stability discussion for Equation (14) is best addressed in the following theorem. Theorem 3 follows from a general result [17,18] on the stability of "pure matrix iterations". The following two facts will also be used: $S^2 = I$ and $S^{-1} = S$:

Theorem 3. *By using Equation (14) and similar assumptions on M or subsequently on* **M** *(to have no pure imaginary eigenvalue), then $\{H_k\}_{k=0}^{\infty}$ with $H_0 =$* **M** *is stable asymptotically.*

Proof. Let γ_k be a perturbation in the numerical solution of the scheme in the k-th iterate and further assume that

$$\widetilde{H}_k = H_k + \gamma_k. \tag{24}$$

We now consider a first-order error analysis which means implicitly that $(\gamma_k)^i \approx 0$, $i \geq 2$. This is rational as long as γ_k is sufficiently small. It is now possible to write down

$$\widetilde{H}_{k+1} = [I + 18\widetilde{H}_k^2 + 13\widetilde{H}_k^4][\widetilde{H}_k(7I + 22\widetilde{H}_k^2 + 3\widetilde{H}_k^4)]^{-1}. \tag{25}$$

For a sufficiently large value k (i.e., in the convergence phase), it is assumed that $H_k \approx \text{sign}(\mathbf{M}) = S$. After extensive simplifications, we obtain

$$\widetilde{H}_{k+1} \approx \left(S + \frac{1}{2}\gamma_k - \frac{1}{2}S\gamma_k S\right). \tag{26}$$

Using $\gamma_{k+1} = \widetilde{H}_{k+1} - H_{k+1}$, we can write

$$\gamma_{k+1} \approx \frac{1}{2}\gamma_k - \frac{1}{2}S\gamma_k S. \tag{27}$$

This can lead to the point that the iteration $k+1$ is bounded. In other words, we have

$$\|\gamma_{k+1}\| \leq \frac{1}{2}\|\gamma_0 - S\gamma_0 S\|. \tag{28}$$

Therefore, the sequence $\{H_k\}_{k=0}^{k=\infty}$, produced via Equation (14), is stable asymptotically. This finishes the proof. □

3.8. Scaling

Since iteration schemes are mostly slow at the beginning of the iterates due to the choice of an initial matrix that is not sharp enough, it is possible to accelerate the process by employing an approach called scaling. This process has been investigated before ([1], Chapter 5.5) for Newton's method based on computing an extra parameter adaptively in each iterate by replacing H_k with $\mu_k H_k$ as follows:

$$\mu_k = \begin{cases} |\det(H_k)|^{\frac{-1}{n}}, & \text{(determinantal scaling)}, \\ \sqrt{\frac{\rho(H_k^{-1})}{\rho(H_k)}}, & \text{(spectral scaling)}, \\ \sqrt{\frac{\|H_k^{-1}\|}{\|H_k\|}}, & \text{(norm scaling)}. \end{cases} \tag{29}$$

Since the proposed method is of the fourth order, we revise Equation (29) as follows:

$$\mu_k = \left(\frac{\|H_k^{-1}\|}{\|H_k\|}\right)^{1/4}. \tag{30}$$

Hence, an accelerated improved version of the mid-point solver using Equation (30) for finding the matrix square root can be obtained as follows:

$$\begin{cases} H_0 = \mathbf{M}, \\ \bar{H}_k = \mu_k H_k, \qquad k \geq 0, \\ \bar{H}_{k+1} = \bar{H}_k(7I + 22\bar{H}_k^2 + 3\bar{H}_k^4)\left[I + 18\bar{H}_k^2 + 13\bar{H}_k^4\right]^{-1}, \end{cases} \tag{31}$$

Note that, in a similar manner, the reciprocal form can be used and written as follows:

$$\bar{H}_{k+1} = \left(I + 18\bar{H}_k^2 + 13\bar{H}_k^4\right)\left[\bar{H}_k\left(7I + 22\bar{H}_k^2 + 3\bar{H}_k^4\right)\right]^{-1}. \tag{32}$$

4. Benchmark Tests

The aim of this section is to evaluate the efficacy of the discussed iteration schemes for finding the matrix square root and its inverse. All tests were performed on the same computer and under the package of Wolfram Mathematica [19].

Here, the following iteration methods (Equations (7)–(9), (17), (18), (14) and (32), denoted by NM, DB, CR, Pade12, Pade12-R, PM and APM, respectively) are compared in terms of the stability and efficiency for finding matrix square roots. These solvers are compared based on the following relative error in l_∞:

$$E_{k+1} = \frac{\|H_{k+1} - H_k\|_\infty}{\|H_{k+1}\|_\infty} \leq \epsilon. \tag{33}$$

The following example is not only one test, since it includes different sizes of the input matrices, and accordingly, four different cases are involved:

Example 1. *To test the effectiveness of the variety of solvers here, we consider the following symmetric positive definite (SPD) matrix of a size n and compute the square root and its inverse simultaneously:*

$$M = \begin{pmatrix} 12 & -5 & -1 & 0 & 0 \\ -5 & \ddots & \ddots & \ddots & 0 \\ -1 & \ddots & \ddots & \ddots & -1 \\ 0 & \ddots & \ddots & \ddots & -5 \\ 0 & 0 & -1 & -5 & 12 \end{pmatrix}_{n \times n}.$$

The results in this test are compared to Pade12 and Pade12-R of the same order of convergence, coming from the Padé family of methods. Computational pieces of evidence are brought forward in Figures 4 and 5 for Example 1 with $\epsilon = 10^{-6}$. For the tested examples, the effectiveness of the obtained numerical results compared with some existing methods can be seen, and it was observed that the PM and APM beat all their other competitors by reaching the stopping termination as fast as possible. In fact, the results reveal that PM of the fourth order of convergence was faster than Pade12 and Pade12-R of the same order of convergence, while all had almost the same computational complexity. PM mostly requires one fewer iterate, in contrast to Pade12 and Pade12-R, which state a lower number of matrix matrix products and one less computation of the stopping termination in Equation (33).

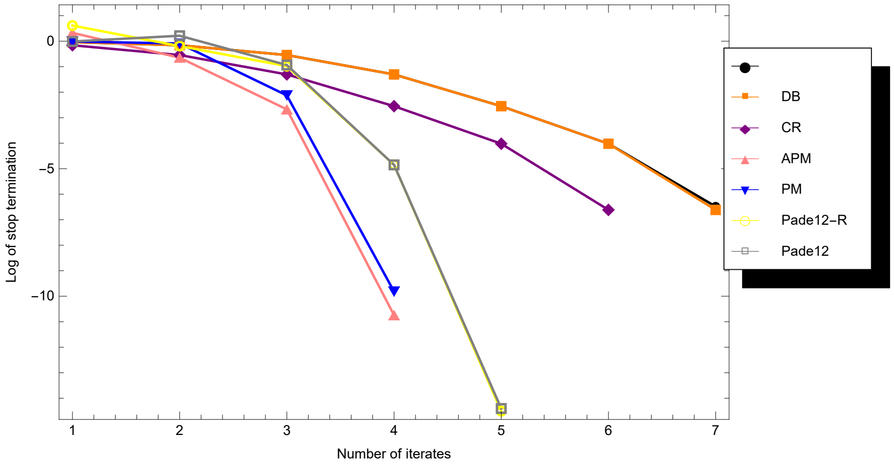

Figure 4. *Cont.*

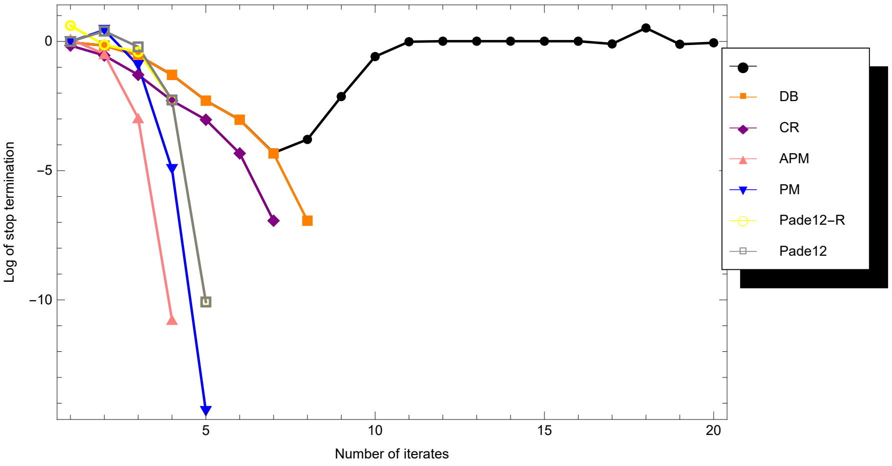

Figure 4. Numerical comparisons for finding the matrix square root in Example 1 when $n = 100$ at the top and $n = 200$ at the bottom.

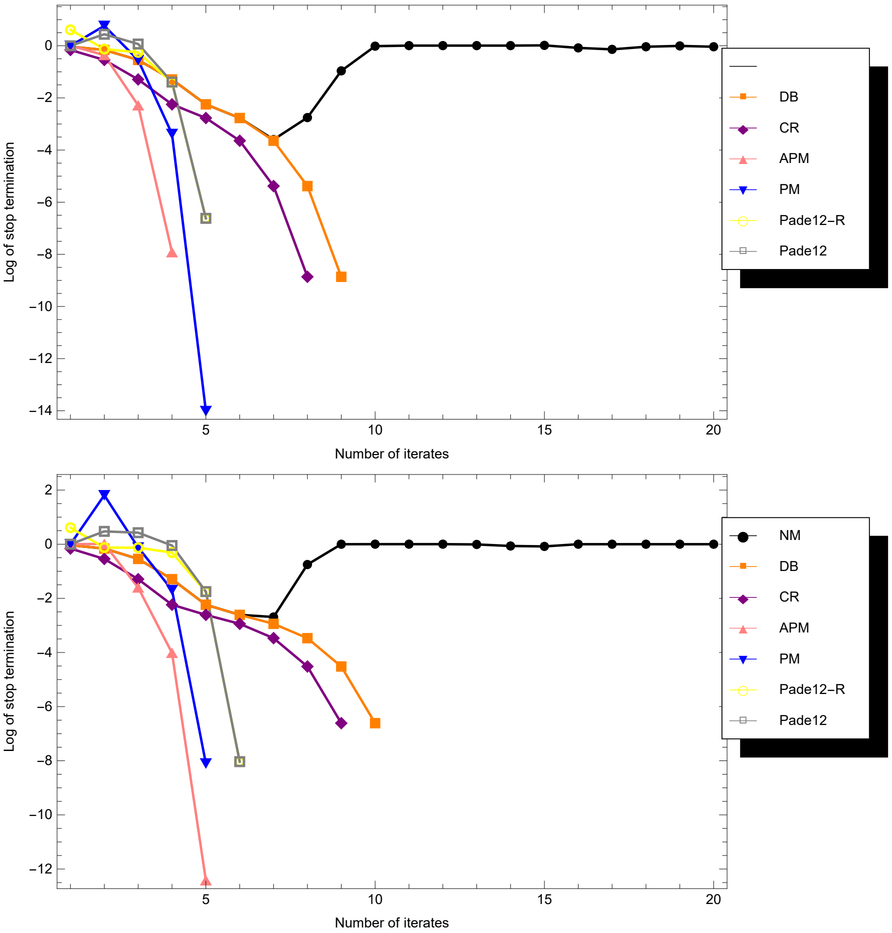

Figure 5. Numerical comparisons for finding the matrix square root in Example 1 when $n = 300$ at the top and $n = 1000$ at the bottom.

To ease the understanding of the proposed algorithm for computing the matrix square root and its inverse simultaneously, here, we provide the written Mathematica code for PM in solving Example 1 when $n = 1000$:

```
ClearAll["Global‘*"];
n = 1000;
M = SparseArray[{{i_, i_} -> 12, {i_, j_} /; Abs[i - j] == 1 -> -5.,
    {i_, j_} /; Abs[i - j] == 2 -> -1.}, {n, n}, 0.];
PositiveDefiniteMatrixQ[M]
Id = SparseArray[{{i_, i_} -> 1.}, {n, n}];
tolerance = 10^-6; max = 20;

M = SparseArray@ArrayFlatten[{{0, M}, {Id, 0}}];
Id = SparseArray[{{i_, i_} -> 1.}, {2 n, 2 n}];
Y[0] = M; k = 0; R5[0] = 0.1;

While[
   k < max && R5[k] >= tolerance,
   Y2 = SparseArray[Y[k].Y[k]];
   Y4 = SparseArray[Y2.Y2];
   l1 = SparseArray[(Y[k].(7 Id + 22 Y2 + 3 Y4))];
   l2 = SparseArray@ArrayFlatten[
      {{0, Inverse@l1[[n + 1 ;; 2 n, 1 ;; n]]},
       {Inverse@l1[[1 ;; n, n + 1 ;; 2 n]], 0}}];
   Y[k + 1] = SparseArray[(Id + 18 Y2 + 13 Y4).l2];
   R5[k + 1] = Norm[Y[k + 1] - Y[k], Infinity]/
    Norm[Y[k + 1], Infinity];
   k++]; // AbsoluteTiming
k
Table[R5[m], {m, 1, k}]
R5[k]
Y[k][[1 ;; n, n + 1 ;; 2 n]];
Y[k][[n + 1 ;; 2 n, 1 ;; n]];
```

5. Conclusions

The calculation of the square root of a matrix and its inverse are of importance not just from the theoretical point of view but also from the numerical and application viewpoints. The computation is required mainly in the context of symmetric positive definite matrices. A common application is in the solution of matrix differential equations of the second order (Equation (1)) and another well-known one in an application at which the inverse matrix square root turns up for the calculation of tight windows of Gabor frames [20].

In this paper, a fourth-order variant of the mid-point method for finding the square root of a matrix as well as its inverse, based on a stable iteration for the matrix sign, was discussed in detail. The convergence rate as well as the scaling of the scheme were given. Finally, numerical simulations were performed to support the theoretical discussions. One of the forthcoming works is to employ the efficient iterations in Equations (12) and (14) in the context of fractional Sturm–Liouville problems [21] in order to devise iterative methods that tackle those problems more efficiently.

Author Contributions: Conceptualization, J.G.; Formal analysis, J.G. and S.S.; Funding acquisition, S.S.; Investigation, J.G. and D.A.; Methodology, J.G., D.A. and S.S.; Supervision, S.S.; Validation, D.A.; Writing—original draft, D.A.; Writing—review & editing, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No special data were used in this study.

Acknowledgments: The authors are grateful to the three anonymous referees for several suggestions on an earlier version of this paper.

Conflicts of Interest: The writers mention that they have no known personal relationships or competing financial interests which could have appeared to influence the work reported in this article.

References

1. Higham, N.J. *Functions of Matrices: Theory and Computation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2008.
2. Cordero, A.; Neta, B.; Torregrosa, J.R. Memorizing Schröder's method as an efficient strategy for estimating roots of unknown multiplicity. *Mathematics* **2021**, *9*, 2570. [CrossRef]
3. Deadman, E.; Higham, N.J. Testing matrix function algorithms using identities. *ACM Trans. Math. Softw.* **2016**, *42*, 1–16. [CrossRef]
4. Cross, G.W.; Lancaster, P. Square roots of complex matrices. *Lin. Multilinear Alg.* **1974**, *1*, 289–293. [CrossRef]
5. Gomilko, O.; Greco, F.; Ziętak, K. A Padé family of iterations for the matrix sign function and related problems. *Numer. Lin. Alg. Appl.* **2012**, *19*, 585–605. [CrossRef]
6. Shirilord, A.; Dehghan, M. Closed-form solution of non-symmetric algebraic Riccati matrix equation. *Appl. Math. Lett.* **2022**, *131*, 108040. [CrossRef]
7. 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]
8. Soheili, A.R.; Soleymani, F. Iterative methods for nonlinear systems associated with finite difference approach in stochastic differential equations. *Numer. Algor.* **2016**, *71*, 89–102. [CrossRef]
9. Soleymani, F.; Soheili, A.R. A revisit of stochastic theta method with some improvements. *Filomat* **2017**, *31*, 585–5966. [CrossRef]
10. Soheili, A.R.; Toutounian, F.; Soleymani, F. A fast convergent numerical method for matrix sign function with application in SDEs. *J. Comput. Appl. Math.* **2015**, *282*, 167–178. [CrossRef]
11. Higham, N.J. Newton's method for the matrix square root. *Math. Comput.* **1986**, *46*, 537–549.
12. Denman, E.D.; Beavers, A.N. The matrix sign function and computations in systems. *Appl. Math. Comput.* **1976**, *2*, 63–94. [CrossRef]
13. 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.
14. Ghorbanzadeh, M.; Mahdiani, K.; Soleymani, F.; Lotfi, T. A class of Kung-Traub-type iterative algorithms for matrix inversion. *Int. J. Appl. Comput. Math.* **2016**, *2*, 641–648. [CrossRef]
15. Kenney, C.S.; Laub, A.J. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.* **1991**, *12*, 273–291. [CrossRef]
16. Zainali, N.; Lotfi, T. A globally convergent variant of mid-point method for finding the matrix sign. *Comp. Appl. Math.* **2018**, *37*, 5795–5806. [CrossRef]
17. Iannazzo, B. Numerical Solution of Certain Nonlinear Matrix Equations. Ph.D. Thesis, Universita degli Studi di Pisa, Pisa, Italy, 2007.
18. Iannazzo, B. On the Newton method for the matrix pth root. *SIAM J. Matrix Anal. Appl.* **2006**, *28*, 503–523. [CrossRef]
19. Wagon, S. *Mathematica in Action*, 3rd ed.; Springer: New York, NY, USA, 2010.
20. Janssen, A.J.E.M.; Strohmer, T. Characterization and computation of canonical tight windows for Gabor frames. *J. Fourier Anal. Appl.* **2002**, *8*, 1–28. [CrossRef]
21. Maralani, E.M.; Saei, F.D.; Akbarfam, A.A.J.; Ghanbari, K. Computation of eigenvalues of fractional Sturm-Liouville problems. *Iran. J. Numer. Anal. Optim.* **2021**, *11*, 117–133.