

Research Article

An Improved Computationally Efficient Method for Finding the Drazin Inverse

Haifa Bin Jebreen ¹ and Yurilev Chalco-Cano²

¹Mathematics Department, College of Science, King Saud University, Riyadh, Saudi Arabia

²Departamento de Matematica, Universidad de Tarapaca, Casilla 7D, Arica, Chile

Correspondence should be addressed to Haifa Bin Jebreen; hjebreen@ksu.edu.sa

Received 2 July 2018; Revised 20 September 2018; Accepted 2 October 2018; Published 17 October 2018

Guest Editor: Tarek F. Ibrahim

Copyright © 2018 Haifa Bin Jebreen and Yurilev Chalco-Cano. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Drazin inverse is one of the most significant inverses in the matrix theory, where its computation is an intensive and useful task. The objective of this work is to propose a computationally effective iterative scheme for finding the Drazin inverse. The convergence is investigated analytically by applying a suitable initial matrix. The theoretical discussions are upheld by several experiments showing the stability and convergence of the proposed method.

1. Introductory Notes

Methods for calculating the generalized inverses are a topic of current investigation in computational mathematics (see, e.g., [1, 2]). Enormous amount of work in the topic of generalized inverses has been done during the past 63 years. E.H. Moore (1920) was a pioneer in the area of a generalized algebraic matrix inverse. But intense activities started since 1955 [3, 4]. Note that R. Penrose (1955, 1956), C.R. Rao and S.K. Mitra (1971), and G.H. Golub and W. Kahan (1965) are just a couple of names who have contributed significantly in the area of generalized matrix inverse and its applications leaving very little scope for others to contribute in this area.

The terms “the minimum-norm least squares inverse,” “the Moore-Penrose inverse,” and “the pseudo-inverse” of a matrix are much more used than the term “Drazin inverse.” This unique inverse is globally used for solving linear systems and linear optimization problems.

The fundamental partitioning method of Greville for calculating generalized inverses was discussed in [3]. This method demands many operations and clearly includes more round-off errors. A lot of computational schemes for computing the Drazin inverse lack numerical stability or slow convergence and accordingly it is requisite to study and propose new iterations for this aim.

It is known that cumulative rounding errors must be canceled completely, which is only doable via “symbolic computations.” For such a case, variables are handled in the exact form, yielding no loss of precision in the process of computation. However, once the dimension of the input matrix is big, the calculation of its Drazin inverse via symbolic methods would be so time-consuming. This encouraged researchers to recommend computational stable and fast iterative methods for such a purpose; see, e.g., [5–7].

The author in [8] investigated that, in case of having the following square singular linear system

$$Ax = b, \quad (1)$$

the general solution can be represented as follows:

$$x = A^D b + (I - AA^D)z, \quad (2)$$

using the Drazin inverse, where $z \in \mathcal{R}(A^{k-1}) + \mathcal{N}(A)$.

Let $\mathbb{C}^{N \times N}$ and $\mathbb{C}_r^{N \times N}$ indicate the collection of $N \times N$ complex matrices and ones having rank r , respectively. Furthermore, with A^* , $\mathcal{R}(A)$, $\text{rank}(A)$, and $\mathcal{N}(A)$, we indicate the conjugate transpose, the range, the rank, and the null space of the matrix $A \in \mathbb{C}^{N \times N}$, respectively.

In the study of associative rings and semigroups, Drazin in the fundamental work [9] showed a different type of generalized inverse that does not possess the reflexivity feature

while commuting with the entry/element. The significance of this type of inverse and its calculation was then completely provided by Wilkinson in [10].

Before continuing, it is required to recall several useful definitions in what follows.

Definition 1. The smallest nonnegative integer k , so that

$$\text{rank}(A^{k+1}) = \text{rank}(A^k), \quad (3)$$

is named as the index of A and it is shown by $\text{ind}(A)$.

Definition 2. Assume that A is an $n \times n$ complex matrix; then the Drazin inverse A^D of the matrix A is a unique matrix V that satisfies the following conditions

$$\begin{aligned} (1^k) \quad & A^k V A = A^k, \\ (2) \quad & V A V = V, \\ (5) \quad & A V = V A, \end{aligned} \quad (4)$$

wherein $k = \text{ind}(A)$ is the matrix index of A .

Recal that once $\text{ind}(A) = 1$, then V is named as the group inverse of A . In addition, if A is nonsingular, thence it is straightforward to observe that $\text{ind}(A) = 0$, and $A^D = A^{-1}$.

Note that the idempotent matrix AA^D is the projector on $\mathcal{R}(A^k)$ alongside $\mathcal{N}(A^k)$, whereas $\mathcal{R}(A^k)$ denotes the range of A^k and $\mathcal{N}(A^k)$ is the null space of A^k . Also, if A is nilpotent, then $A^D = 0$; see [11–14] for more.

Matrix iterations are sensitive upon the choice of the first approximation (V_0) to initiate the scheme and observe the convergence to A^D . Practically speaking, the iterative methods (like the Schulz-type iterations) are effective (particularly for sparse matrices having sparse inverses) but an issue lies in the initial value. However, this requirement was lifted by furnishing several suitable initial values in the literature. A vast discussion on choosing the initial choice V_0 is given in the work [15].

In this work, we concentrate on proposing and investigating a novel matrix iteration for calculating the Drazin inverse quickly and efficiently with a clear concentration on decreasing the elapsed CPU times in contrast to several well-known competitors in the literature. To do this, an analytical discussion will be furnished to manifest the behavior of the presented method. It is proven that the novel approach has a high convergence order using fewer number of matrix-matrix multiplications; i.e., it has a better computational efficiency index.

After a short introduction in Section 1 and a brief review on the most common iterative schemes for calculating the Drazin inverse in Section 2, we propose a novel iterative method for computing the Drazin inverse. Section 3 unfolds our contributed method as a novel high order efficient method. Section 4 furnishes a discussion regarding its convergence rate. Next in Section 5, we study the complexity of the iterative methods to analytically select the best iterative expression. In Section 6, we numerically investigate the application and usefulness of the novel scheme in the calculation

of the Drazin inverse. A clear decrease of the elapsed CPU time will be seen therein. Ultimately in Section 7, summary and comments are brought forward.

2. The Literature

One of the fundamental techniques to calculate the inverse of a nonsingular complex matrix A is the Schulz method given in [16] as follows:

$$V_{n+1} = V_n (2I - AV_n), \quad n = 0, 1, 2, \dots, \quad (5)$$

wherein I is the unit matrix of the same dimension as A .

In 2004, Li and Wei in [17] proved that the matrix method of Schulz (5) could be applied for calculating the Drazin inverse of square matrices having real or complex eigenvalues. They proposed the following initial matrix

$$V_0 = V_0 = \alpha A^l, \quad l \geq \text{ind}(A) = k, \quad (6)$$

wherein the parameter α should be selected such that the following criterion holds

$$\|I - AV_0\| < 1. \quad (7)$$

Using the initial matrix of the form (6) along with (5) results in a quadratically convergent iterative scheme for calculating the well-known Drazin inverse.

Let us now briefly review some of the higher order iteration schemes for such a purpose. The notion of the need for efficient schemes is the fact that (5) is slow at its initial stage of iterates, and this would increase the computational burdensome of the scheme applied for matrix inversion.

Li et al. in [18] studied and discussed an iterative method in the following formulation

$$V_{n+1} = V_n (3I - AV_n (3I - AV_n)), \quad n = 0, 1, 2, \dots, \quad (8)$$

with cubic convergence rate and also presented another scheme for calculating A^{-1} (and similarly for A^D using a suitable initial value) of the same order as it is provided in the coming formula:

$$V_{n+1} = V_n \left[I + \frac{1}{2} (I - AV_n) (I + (2I - AV_n)^2) \right], \quad (9)$$

$$n = 0, 1, 2, \dots$$

Recall that a general way for deriving similar iterations was furnished in [19, Chapter 5]. In fact, Krishnamurthy and Sen proposed the following quartically-convergent scheme:

$$V_{n+1} = V_n (I + R_n (I + R_n (I + R_n))), \quad (10)$$

$$n = 0, 1, 2, \dots,$$

in which $R_n = I - AV_n$. As another instance, a ninth-order matrix iteration can be written as follows:

$$V_{n+1} = V_n (I + R_n (I + R_n))))))), \quad (11)$$

$$n = 0, 1, 2, \dots$$

Generally speaking, applying Schröder's general iteration (often named as Schröder-Traub's sequence [20]) to the nonlinear matrix equation

$$AV = I, \quad (12)$$

one can obtain the following scheme [21]:

$$\begin{aligned} V_{n+1} &= V_n (I + R_n + R_n^2 + \cdots + R_n^{m-1}) \\ &= V_n (I + R_n (I + R_n (\cdots + R_n) \cdots)), \end{aligned} \quad (13)$$

$n = 0, 1, 2, \dots,$

of order m , needing m Horner's matrix products, whereas $R_n = I - AV_n$.

3. Proposing a New Method

Now, we propose a high order scheme which is computationally efficient in terms of the number of matrix-matrix multiplications. Let us first consider a tenth-order method using (13) as follows:

$$V_{n+1} = V_n (I + R_n + R_n^2 + \cdots + R_n^9). \quad (14)$$

To improve the performance of this scheme, we factorize (14) as much as possible so as to decrease the number of matrix-matrix products. First we can obtain

$$V_{n+1} = V_n (I + R_n (I + R_n + R_n^2) (I + R_n^3 + R_n^6)), \quad (15)$$

which includes seven matrix-matrix multiplications.

By further factorizations and simplifications, we can propose the following iterative method:

$$V_{n+1} = V_n (I + R_n) [(I + \alpha R_n^2 + R_n^4) (I + \beta R_n^2 + R_n^4)], \quad (16)$$

with only 6 matrix-matrix products per cycle where

$$\begin{aligned} \alpha &= \frac{1}{2} (1 - \sqrt{5}), \\ \beta &= \frac{1}{2} (1 + \sqrt{5}). \end{aligned} \quad (17)$$

Obtaining α and β so as to reduce the matrix matrix products is new and not easy. In fact, we should solve a system of equations in symbolic environment so as to do such a task.

The Schulz-type iterations such as (16) are numerically stable; i.e., they have the self-correcting characteristic and are exactly based upon matrix multiplication per cycle. Multiplication is efficiently parallelizable for special matrices. The method (16) could be mixed with sparse-saving techniques so as to decrease the burdensome of matrix-matrix products per cycle.

We can now apply (16) with tenth convergence rate for calculating the Drazin inverse when the first value is selected as follows:

$$V_0 = \frac{2}{\text{Tr}(A^{k+1})} A^k, \quad (18)$$

or

$$V_0 = \frac{1}{2 \|A\|_2^{k+1}} A^k, \quad (19)$$

wherein $\text{Tr}(\cdot)$ stands for the trace of an arbitrary square matrix with index k .

Before providing the main theorems regarding the convergence analysis and the rate of convergence of the proposed method, we recall the following lemmas.

Proposition 3 (see [22]). *Let $M \in \mathbb{C}^{n \times n}$ and $\epsilon > 0$ be given. There is at least one matrix norm $\|\cdot\|$ so that*

$$\rho(M) \leq \|M\| \leq \rho(M) + \epsilon, \quad (20)$$

where $\rho(M)$ indicates the set of all eigenvalues of M (in the maximum of absolute value sense).

Proposition 4 (see [22]). *If $P_{L,M}$ indicates the projector on a space L along a space M , then*

- (i) $P_{L,M}Q = Q$ iff $\mathcal{R}(Q) \subseteq L$;
- (ii) $QP_{L,M} = Q$ iff $\mathcal{N}(Q) \supseteq M$.

4. Convergence Analysis

Theorem 5. *Assume that $A \in \mathbb{C}^{N \times N}$ is a singular square matrix. In addition, let the initial value V_0 be selected via (6) or (18). Thence, the matrices $\{V_n\}_{n=0}^{\infty}$ generated via (16) satisfy the following error estimate for calculating the Drazin inverse:*

$$\frac{\|A^D - V_n\|}{\|A^D\|} \leq \|I - AV_0\|^{10^n}. \quad (21)$$

Also the convergence speed is ten.

Proof. Let us write

$$\begin{aligned} R_{n+1} &= I - AV_{n+1} = I - A (V_n (I + R_n) \\ &\quad \cdot [(I + \alpha R_n^2 + R_n^4) (I + \beta R_n^2 + R_n^4)]) = I \\ &\quad - A (V_n (I + R_n (I + R_n + R_n^2) (I + R_n^3 + R_n^6))) \\ &= I \\ &\quad - A (V_n (I + R_n (I + R_n + R_n^2) (I + R_n^3 + R_n^6))) \\ &= (I - AV_n)^{10} = R_n^{10}, \end{aligned} \quad (22)$$

wherein α and β are defined in (17). Applying an arbitrary matrix norm on relation (22), it is possible to write down

$$\|R_{n+1}\| \leq \|R_n\|^{10}. \quad (23)$$

Since V_0 is chosen as in (6) or (18), it follows that

$$\mathcal{R}(V_0) \subseteq \mathcal{R}(A^k). \quad (24)$$

This could now state that

$$\mathcal{R}(V_n) \subseteq \mathcal{R}(V_{n-1}). \quad (25)$$

Thus, we can conclude that

$$\mathcal{R}(V_n) \subseteq \mathcal{R}(A^k), \quad n \geq 0. \quad (26)$$

Similarly if the novel scheme for the Drazin inverse is defined by left multiplying of V_n , we can state that

$$\mathcal{N}(V_n) \supseteq \mathcal{N}(A^k), \quad n \geq 0. \quad (27)$$

Now, an application of definition of the Drazin inverse yields

$$AA^D = A^D A = P_{\mathcal{R}(A^k), \mathcal{N}(A^k)}. \quad (28)$$

Proposition 4 along with (26), (27), and (28) could lead to

$$V_n AA^D = V_n = A^D AV_n, \quad n \geq 0. \quad (29)$$

To complete the proof, we proceed in what follows. The error matrix $\delta_n = A^D - V_n$ satisfies

$$\begin{aligned} \delta_n &= A^D - V_n = A^D - A^D AV_n = A^D (I - AV_n) \\ &= A^D R_n. \end{aligned} \quad (30)$$

Using (23), we obtain

$$\|\delta_n\| = \|A^D\| \|R_n\| \leq \|A^D\| \|R_0\|^{10^n}, \quad (31)$$

which is a confirmation of (21). As a direct result of (31) and Proposition 4, we can obtain

$$\begin{aligned} A\delta_{n+1} &= AA^D - AV_{n+1} = AA^D - I + I - AV_{n+1} \\ &= AA^D - I + R_{n+1}. \end{aligned} \quad (32)$$

Considering (22) and applying several simplifications, we obtain that

$$A\delta_{n+1} = AA^D - I + R_n^{10}. \quad (33)$$

Applying the idempotent property $(I - AA^D)^t = (I - AA^D)$, $t \geq 1$ being a positive integer from now on, and the following fact of (29):

$$\begin{aligned} (I - AA^D)A\delta_n &= (I - AA^D)A(A^D - V_n) \\ &= V_n - AA^D V_n = 0, \end{aligned} \quad (34)$$

we obtain for each $t \geq 1$ that (here we use (34) in simplifications)

$$\begin{aligned} (R_n)^t + AA^D - I &= (I - AV_n)^t + AA^D - I \\ &= (I - AA^D + AA^D - AV_n)^t \\ &\quad + AA^D - I \\ &= ((I - AA^D) + A\delta_n)^t + AA^D - I \\ &= I - AA^D + (A\delta_n)^t + AA^D - I \\ &= (A\delta_n)^t. \end{aligned} \quad (35)$$

From (35) and (33), we have

$$A\delta_{n+1} = (A\delta_n)^{10}. \quad (36)$$

Finally, we get that

$$\|A\delta_{n+1}\| \leq \|A\delta_n\|^{10}. \quad (37)$$

It is now straightforward to calculate the error inequality of the proposed iteration (16) considering (37) and the second criterion of (4), when computing the Drazin inverse, as comes next

$$\begin{aligned} \|\delta_{n+1}\| &= \|V_{n+1} - A^D\| = \|A^D AV_{n+1} - A^D AA^D\| \\ &= \|A^D (AV_{n+1} - AA^D)\| \leq \|A^D\| \|A\delta_{n+1}\| \\ &\leq \|A^D\| \|\delta_n\|^{10}. \end{aligned} \quad (38)$$

The inequalities in (38) lead to the fact that $V_n \rightarrow A^D$ as $n \rightarrow +\infty$ with the tenth convergence speed. The proof is ended now. \square

Remark 6. An important issue is to find initial approximations V_0 . In accordance with Proposition 3, V_0 must read as the following relation to guarantee the convergence in the Drazin inverse case:

$$\max_{1 \leq i \leq t} |1 - \lambda_i(AV_0)| < 1, \quad (39)$$

where

$$\text{rank}(AV_0) = h, \quad (40)$$

and $\lambda_i(AV_0)$, $i = 1, 2, \dots, h$, are eigenvalues of AV_0 .

5. Efficiency Comparison

The computational complexity of the matrix inverse is $O(mn\text{-squared})$, where the order of the matrix is $m \times n$ (a rectangular matrix). Broadly speaking, if the complexity $O(n\text{-cubed})$ is brought down to $O(n^k)$, where $k < 3$, say 2.9 for a general matrix, then it is definitely an achievement. Reducing the complexity by a linear factor is not attractive since what we have currently in 2018 is exa-flops computing speed and desktop and laptop computers are available in 100s of millions worldwide unlike the period during mid-20th centuries.

Thus, we calculate and compare theoretically the computational efficiencies of various schemes (5), (8), (9), (10), (11), and (16), since they all could converge to A^D upon the choice of a suitable initial value (6).

By considering a unit cost for the arithmetic operations, typical of the floating point calculations, we can take into account the *inverse-finder informational efficiency index*. This index applies two values ρ and κ , which are for the convergence speed and the number of matrix-matrix products, respectively. Hence, this index is given by [20]

$$IIEI = \frac{\rho}{\kappa}. \quad (41)$$

Apart from this index, another useful one which is mainly in agreement with the CPU elapsed time of the iterative methods in numerical linear algebra is the *computational efficiency index* defined by [23]

$$CEI = \rho^{1/\kappa}. \tag{42}$$

Therefore, a fruitful scheme in analytical standpoint should achieve the speed ρ with fewer matrix products κ in contrast to the competitors (viz, $\kappa \leq \rho$).

In Table 1, we provide a comparison of the number of matrix products and the rate of convergence, accompanied by (41) and (42) for various schemes. The results indicate that the proposed scheme in Section 3 is more efficient than its competitors.

As a matter of fact, one may observe that the proposed iteration (16) decreases the calculation burdensome via applying fewer operations and yields a better balance between the higher order and the computational load. This fact will be numerically investigated later in Section 6.

6. Experiments

This section studies issues associated with the computational accuracy and times of finding the Drazin inverse. The new iteration (16) is free from matrix power in its implementation and this allows one to apply it for finding generalized inverses easily. For computational comparisons, we have used the methods (5), (8), (9), (10), (11), and (16), denoted by “SM2,” “CM3,” “LM3,” “KMS4,” “HM9,” and “PM10,” respectively.

A couple of remarks are in order:

- (i) The simulations are done in Mathematica 11.0 [24].
- (ii) The time needed to obtain the approximate inverses is reported in seconds.
- (iii) Whenever the elapsed times are reported, the compared methods are programmed in the same environment.

Experiment 1. The aim of this experiment is to calculate the Drazin inverse while the input matrix is [17]

$$A = \begin{bmatrix} 2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 2 \end{bmatrix}, \tag{43}$$

with $k = \text{ind}(A) = 3$. The exact Drazin inverse is given by

$$A^D = \begin{bmatrix} 0.25 & -0.25 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1.25 & 1.25 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -1.66406 & -0.992187 & 0.25 & -0.25 & 0. & 0. & 0. & 0. & -0.0625 & -0.0625 & 0. & 0.15625 \\ -1.19531 & -0.679687 & -0.25 & 0.25 & 0. & 0. & 0. & 0. & -0.0625 & 0.1875 & 0.6875 & 1.34375 \\ -2.76367 & -1.04492 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.48438 & 2.57813 & 3.32031 & 6.64063 \\ -2.76367 & -1.04492 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.48438 & 2.57813 & 4.57031 & 8.51563 \\ 14.1094 & 6.30078 & 6.625 & 3.375 & 5. & -3. & -5. & -5. & -4.1875 & -8.5 & -10.5078 & -22.4609 \\ -19.3242 & -8.50781 & -9.75 & -5.25 & -7.5 & 4.5 & 7.5 & 7.5 & 6.375 & 12.5625 & 15.9766 & 33.7891 \\ -0.625 & -0.3125 & 0. & 0. & 0. & 0. & 0. & 0. & 0.25 & -0.25 & -0.875 & -1.625 \\ -1.25 & -0.9375 & 0. & 0. & 0. & 0. & 0. & 0. & -0.25 & 0.25 & -0.875 & -1.625 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1.25 & 1.25 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -0.25 & 0.25 \end{bmatrix}. \tag{44}$$

The stopping termination is $\|V_{k+1} - V_k\|_\infty \leq \epsilon$ with $\epsilon = 10^{-8}$. Checking the conditions of Definition 2 for our proposed iterative method PM10 yields

$$\|A^{k+1}A^D - A^k\|_\infty = 3.69638 \times 10^{-12},$$

$$\|A^D A A^D - A^D\|_\infty = 8.43992 \times 10^{-10},$$

$$\|A A^D - A^D A\|_\infty = 3.75205 \times 10^{-10},$$

(45)

which supports the theoretical discussions.

The practical application of the new scheme (16) lies in several problems; see, for example, [25, 26]. For instance, in the process of resolving second-kind integral equations via Wavelet-like technique, the whole discretized problem will be yielded to calculate the inverse of a large sparse matrix [27].

Additionally, we can apply/use (16), as an approach to provide good preconditioners for speeding-up modern Krylov methods, such as GMRES or BiCGSTAB, for solving large scale sparse linear systems; see, e.g., [28]. For such a task, we need to define a new initial matrix defined in Table 2 [29].

In modern numerical linear algebra, schemes like (16) should be coded in sparse form using some well-known commands such as `SparseArray[]` to reduce the computational burden and preserve the sparsity feature of the approximate inverse per computing step.

It is necessary to test the behavior of the new method in a fair environment with a clear comparison taking into account various competitors. On the other hand, since the generation of random square matrices having Drazin inverses is difficult, in what follows, we compare various competitors in terms of the elapsed computational time so as to attain regular approximate inverses for large sparse matrices.

Experiment 2. The aim of this experiment is to compare the elapsed CPU times of different methods for the following 25 large sparse random complex matrices:

```
SeedRandom[123]
n = 5000; number = 25;
Table[
  A[j] = SparseArray[{Band[{-100, 1100}]} -> RandomReal[20],
    Band[{1, 1}] -> 2.,
    Band[{1000, -50}, {n - 20, n - 25}] -> {2.8, RandomReal[] + I},
    Band[{600, 150},
      {n - 100, n - 400}] -> {
      -RandomReal[3], 3. + 3 I}
    ], {n, n}, 0.], {j, number}];
```

Here $I = \sqrt{-1}$. In this test, the stopping criterion is $\|V_{n+1} - V_n\|_1 \leq 10^{-6}$ and the maximum number of iterations allowed is set to 75. Moreover, the initial value is constructed via Form 3 of Table 2.

The results of comparisons for this test are presented in Figure 1. As is obvious that higher order methods require lower number of iterations to converge, thus we then put our focus on the computational time needed to satisfy the desired tolerance. As could be observed, our scheme (16) beats its competitors and also supports the analytical reports in Table 1.

Experiment 3. Here we rerun Experiment 2 with the stopping termination $\|V_{n+1} - V_n\|_\infty \leq 10^{-4}$ and initial value chosen by Form 2 of Table 2. The results are summarized in Figure 2. Furthermore, to check the usefulness of PM10, we rerun

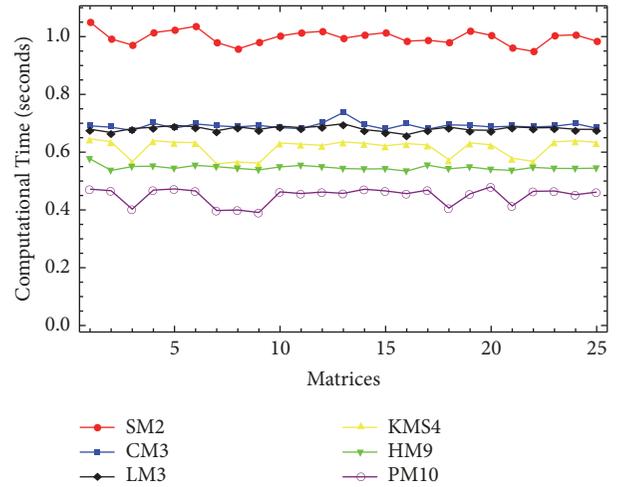


FIGURE 1: Comparison of the elapsed times for various methods in Experiment 2.

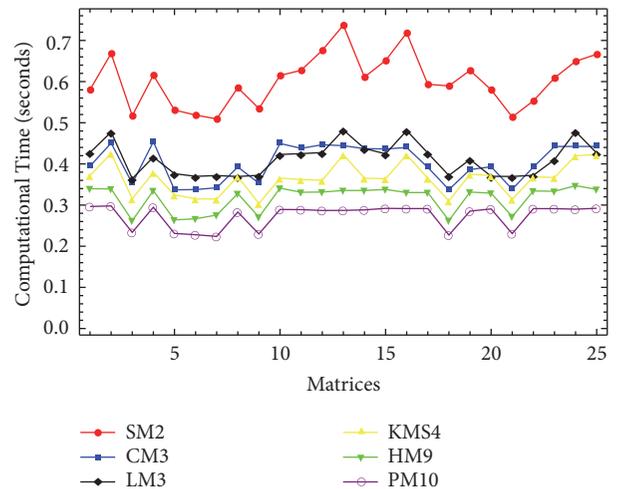


FIGURE 2: Comparison of the elapsed times for various methods in Experiment 3.

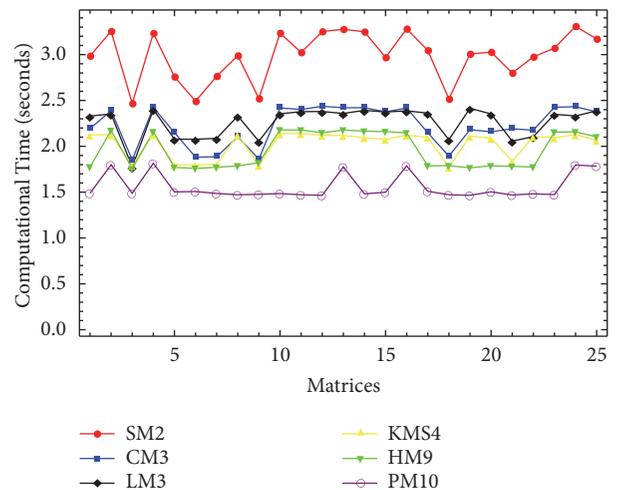


FIGURE 3: Comparison of the elapsed times for various methods in Experiment 3 for larger sizes $A_{10000 \times 10000}$ and a different initial value.

TABLE 1: A comparison on the calculational burdensome of various methods applying different indices.

Methods	(5)	(8)	(9)	(10)	(11)	(16)
ρ	2	3	3	4	9	6
κ	2	3	4	4	9	10
III EI	$\frac{2}{2} = 1$	$\frac{3}{3} = 1$	$\frac{3}{4} = 0.75$	$\frac{4}{4} = 1$	$\frac{9}{9} = 1$	$\frac{10}{6} \approx 1.666$
CEI	$2^{1/2} \approx 1.414$	$3^{1/3} \approx 1.442$	$3^{1/4} \approx 1.316$	$4^{1/4} \approx 1.414$	$9^{1/9} \approx 1.276$	$10^{1/6} \approx 1.467$

TABLE 2: Several choices for V_0 (for calculating A^{-1}).

Different ways	Form 1	Form 2	Form 3	Form 4	Form 5
Initial value	$\frac{A}{\ A\ _1^2}$	$\frac{A}{\ A\ _\infty^2}$	$\frac{A}{\ A\ _F^2}$	$\frac{A^T}{N \ A\ _1 \ A\ _\infty}$	$\frac{A^*}{\ A\ _2^2}$

Experiment 2 with larger sizes matrices, i.e., when $n = 10000$, the stopping termination $\|V_{n+1} - V_n\|_\infty \leq 10^{-4}$ and initial value chosen by Form 1 of Table 2. The results of comparisons for this case are given in Figure 3.

The attained results have reverified the robustness of the proposed iterative method (16) by a clear reduction in the elapsed CPU times.

Note that the 10th-order method is better than a fourth-order method in CPU time. However, this is more tangible if a sharp initial approximation is chosen meaning that the 10th-order method arrives at the convergence phase quickly.

We also emphasise that the construction of any higher order method is meaningful only if we observe an improvement in CEI as discussed in Section 5. Accordingly, a member of the family (13) should be chosen such that we could arrive at the unknown parameters (e.g., like (17)) in order to improve CEI. An attempt to find an optimal iteration in this way is still a research topic in the field.

Remark 7. With the standard precision of 15 digits (most widely used globally, e.g., in Matlab or Mathematica), convergence of order 2 or 3 has been found to be computationally (amount of numerical computation) optimal [19].

7. Summary and Remarks

The Drazin inverse is investigated in the matrix theory (particularly in the topic of generalized inverses) and also in the ring theory; see, e.g., [30].

In this work, we have investigated a higher-order matrix scheme (16) for calculating the Drazin inverse. Convergence analysis of our scheme has been discussed and an investigation on the selection of the initial approximation so as to initiate the iterates and keep the convergence speed was furnished. We also discussed under what conditions the novel scheme can be taken into account for calculating the Drazin inverse of square matrices.

Furthermore, the total elapsed timings consuming of the proposed scheme (16) is low in contrast to the competitors of this type in the case of constructing approximate inverses.

Tackling on the generalization of the new scheme (16) for interval matrix inversion or the application of such matrix

methods in option pricing in order to act as a preconditioner to reduce the ill-conditioning of the large sized matrices occurring in the process of pricing (see, e.g., [31]) could be taken into account as future investigations in this field of study.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The funding body had no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research project was supported by a grant from the Research Center of the Female Scientific and Medical Colleges, Deanship of Scientific Research, King Saud University.

References

- [1] M. Ghorbanzadeh, K. Mahdiani, F. Soleymani, and T. Lotfi, "A class of Kung-Traub-type iterative algorithms for matrix inversion," *International Journal of Applied and Computational Mathematics*, vol. 2, no. 4, pp. 641–648, 2016.
- [2] X. Liu, G. Zhu, G. Zhou, and Y. Yu, "An analog of the adjugate matrix for the outer inverse $AT,S(2)$," *Mathematical Problems in Engineering*, Art. ID 591256, 14 pages, 2012.
- [3] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer, New York, NY, USA, 2nd edition, 2003.
- [4] A. Kumar, P. S. Stanimirović, F. Soleymani, M. Krstić, and K. Rajković, "Factorizations of hyperpower family of iterative methods via least squares approach," *Computational & Applied Mathematics*, vol. 37, no. 3, pp. 3226–3240, 2018.

- [5] I. I. Kyrchei, "Cramer's Rule for Generalized Inverse Solutions," in *Advances in Linear Algebra Research*, I. Kyrchei, Ed., pp. 79–132, Nova Science Publishers, New York, NY, USA, 2015.
- [6] V. Y. Pan, F. Soleymani, and L. Zhao, "An efficient computation of generalized inverse of a matrix," *Applied Mathematics and Computation*, vol. 316, pp. 89–101, 2018.
- [7] F. Soleymani, "A Rapid Numerical Algorithm to Compute Matrix Inversion," *International Journal of Mathematics and Mathematical Sciences*, vol. 2012, Article ID 134653, 11 pages, 2012.
- [8] Y. Wei, "Index splitting for the Drazin inverse and the singular linear system," *Applied Mathematics and Computation*, vol. 95, no. 2-3, pp. 115–124, 1998.
- [9] M. P. Drazin, "Pseudo-inverses in associative rings and semi-groups," *The American Mathematical Monthly*, vol. 65, pp. 506–514, 1958.
- [10] J. H. Wilkinson, "Note on the practical significance of the Drazin inverse," in *Recent Applications of Generalized Inverses*, S. L. Campbell, Ed., vol. 66 of *Research Notes in Mathematics*, pp. 82–99, Pitman Advanced Publishing Program, Boston, 1982.
- [11] I. Kyrchei, "Explicit formulas for determinantal representations of the Drazin inverse solutions of some matrix and differential matrix equations," *Applied Mathematics and Computation*, vol. 219, no. 14, pp. 7632–7644, 2013.
- [12] F. Soleymani, P. S. Stanimirović, and F. Khaksar Haghani, "On hyperpower family of iterations for computing outer inverses possessing high efficiencies," *Linear Algebra and its Applications*, vol. 484, pp. 477–495, 2015.
- [13] Y. Xia, T. Chen, and J. Shan, "A novel iterative method for computing generalized inverse," *Neural Computation*, vol. 26, no. 2, pp. 449–465, 2014.
- [14] L. Zhao, "The Expression of the Drazin Inverse with Rank Constraints," *Journal of Applied Mathematics*, vol. 2012, Article ID 390592, 10 pages, 2012.
- [15] V. Y. Pan, *Structured matrices and polynomials*, BirkhWauser Boston, Inc., Boston, MA; Springer-Verlag, New York, 2001.
- [16] G. Schulz, "Iterative Berechnung der reziproken Matrix," *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, no. 1, pp. 57–59, 1933.
- [17] X. Li and Y. Wei, "Iterative methods for the Drazin inverse of a matrix with a complex spectrum," *Applied Mathematics and Computation*, vol. 147, no. 3, pp. 855–862, 2004.
- [18] H.-B. Li, T.-Z. Huang, Y. Zhang, V.-P. Liu, and T.-V. Gu, "Chebyshev-type methods and preconditioning techniques," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 260–270, 2011.
- [19] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms: Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, India, 2007.
- [20] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New York, NY, USA, 1964.
- [21] S. K. Sen and S. S. Prabhu, "Optimal iterative schemes for computing the Moore-Penrose matrix inverse," *International Journal of Systems Science*, vol. 7, no. 8, pp. 847–852, 1976.
- [22] F. Soleymani and P. S. Stanimirović, "A Higher Order Iterative Method for Computing the Drazin Inverse," *The Scientific World Journal*, vol. 2013, Article ID 708647, 11 pages, 2013.
- [23] A. M. Ostrowski, "Sur quelques transformations de la serie de LiouvilleNewman," *Comptes rendus de l'Académie des Sciences, Paris*, vol. 206, pp. 1345–1347, 1938.
- [24] M. Trott, *The Mathematica guidebook for numerics*, Springer, New York, NY, USA, 2006.
- [25] S. L. Campbell, C. D. Meyer Jr., and N. J. Rose, "Applications of the Drazin inverse to linear systems of differential equations with singular constant coefficients," *SIAM Journal on Applied Mathematics*, vol. 31, no. 3, pp. 411–425, 1976.
- [26] N. C. González, J. J. Koliha, and V. Rakocević, "Continuity and general perturbation of the Drazin inverse for closed linear operators," *Abstract and Applied Analysis*, vol. 7, pp. 335–347, 2002.
- [27] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, "Wavelet-like bases for the fast solution of second-kind integral equations," *SIAM Journal on Scientific Computing*, vol. 14, no. 1, pp. 159–184, 1993.
- [28] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd edition, 2003.
- [29] J. Rajagopalan, *An iterative algorithm for inversion of matrices*, M.Sc. Thesis, Concordia University, Canada, 1996.
- [30] B. Cantó, C. Coll, and E. Sánchez, "Identifiability for a Class of Discretized Linear Partial Differential Algebraic Equations," *Mathematical Problems in Engineering*, vol. 2011, Article ID 510519, 12 pages, 2011.
- [31] R. Company, V. N. Egorova, L. Jodar, and F. Soleymani, "A local radial basis function method for high-dimensional American option pricing problems," *Mathematical Modelling and Analysis*, vol. 23, no. 1, pp. 117–138, 2018.

