

Research Article

A Higher Order Iterative Method for Computing the Drazin Inverse

F. Soleymani¹ and Predrag S. Stanimirović²

¹ Department of Mathematics, Zahedan Branch, Islamic Azad University, Zahedan, Iran

² Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia

Correspondence should be addressed to F. Soleymani; fazlollah.soleymani@gmail.com

Received 6 August 2013; Accepted 27 August 2013

Academic Editors: S. A. Mohiuddine and Q. Xie

Copyright © 2013 F. Soleymani and P. S. Stanimirović. 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.

A method with high convergence rate for finding approximate inverses of nonsingular matrices is suggested and established analytically. An extension of the introduced computational scheme to general square matrices is defined. The extended method could be used for finding the Drazin inverse. The application of the scheme on large sparse test matrices alongside the use in preconditioning of linear system of equations will be presented to clarify the contribution of the paper.

1. Introduction

Computing the matrix inverse of nonsingular matrices of higher sizes is difficult and is a time consuming task. Application of higher order algorithms to solve this problem is very desirable. Generally speaking, in wide variety of topics, one must compute the inverse or particularly the generalized inverses to comprehend and realize significant features of the involved problems [1]. An example could be in phased-array radar whereas the target tracking is a recursive prediction correction process, when Kalman filtering is extensively consumed; see [2, 3]. Target equations are modeled explicitly such that the position and velocity and potentially higher derivatives of each measurement are approximated by the track filter as a state vector. The approximated error with the state vector is modeled by taking into account a covariance matrix, which is then used in subsequent computations. To be more precise, this matrix gets updated in each iteration of the track filter. Finding the inverse in the next iteration could make use of the inverse in the present iteration. In this circumstance, fast and efficient iterative algorithms are required.

There are some techniques to tackle this problem, which are basically divided into two main parts: the direct solvers such as Gaussian Elimination with Partial Pivoting (GEPP), which requires a massive load of computations and memory

for large scale problems, and the iterative methods of the class of Schulz-type iterations, in which an approximation of the matrix inverse (by using a threshold) can be found per step up to the desired accuracy.

Almost all of the direct methods for matrix inversion require high accuracy in the computations to attain proper solutions as they are not tolerant to errors in the computed matrices. In contrast, iterative method compensates for individual and accumulation of round-off errors as it is a process of successive refinement.

In this paper, we focus on presenting and demonstrating a new iterative method to find approximate inverse matrices as fast as possible with a close attention in reducing the computational time. Toward this goal, a theoretical discussion will also be given to show the behavior of the proposed scheme. An interesting point in the contribution is that it could easily be applied to complex matrices as well as for finding the Drazin inverse.

To clarify the procedure, we now remind of some of the well-known methods in what follows. Perhaps, the most common technique to compute the inverse of a nonsingular complex matrix A , is the Schulz method given in [4] as follows:

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

TABLE 1: Some of the general ways to choose V_0 .

Forms	Form 1	Form 2	Form 3	Form 4	Form 5
Initial formulation	$A/\ A\ _1^2$	$A/\ A\ _\infty^2$	$A/\ A\ _F^2$	$A^T/(N\ A\ _1\ A\ _\infty)$	$A^*/\ A\ _2^2$

where I is the identity matrix with the same dimension that of the matrix A . The scheme (1) has become popular in the 1980s due to the emerging of parallel machines.

Such iterative methods are sensitive for the initial guess/value (V_0) to start the process and converge to A^{-1} . In practice, the Schulz-type methods are efficient (specially for structured matrices) but a difficulty lies in the initial approximation of the A^{-1} . This need was fulfilled by providing some appropriate initial approximations in the literature. For example, Rajagopalan in [5] gave some initial approximations for the inverse matrix by considering different norms as:

$$V_0 = \frac{A}{\|A\|_\infty^2}. \tag{2}$$

In fact, by choosing (2), we attain

$$\frac{\|V_0\|_\infty}{\|A^{-1}\|_\infty} = \frac{\|A\|_\infty}{\|A\|_\infty^2\|A^{-1}\|_\infty}. \tag{3}$$

Based on the fact that $\kappa(A) = \|A\|\|A^{-1}\| \geq \|I\| \geq 1$, we could have

$$\frac{\|V_0\|_\infty}{\|A^{-1}\|_\infty} = \frac{1}{\kappa_\infty(A)} \leq 1, \tag{4}$$

which suggests that by choosing (2), the iterative scheme (1) will be almost always convergent for regular matrices.

Some other ways to choose the initial approximation V_0 have been listed in Table 1, where A^T and A^* are the transpose and the conjugate transpose of the complex matrix A , respectively, and N stands for the size of the square matrix.

A vast discussion on choosing the initial approximation V_0 is given in [6]. For instance, Pan and his coworkers discussed the possible ways to reduce the computational load of Schulz-type iteration methods for structured matrices such as Toeplitz or Hankel matrices; see, for example, [7]. To illustrate further, for a symmetric positive definite (SPD) matrix A , one can choose V_0 as follows:

$$V_0 = \frac{I}{\|A\|_F}. \tag{5}$$

Another interesting choice is based on [8] for diagonally dominant matrices, which is fruitful when dealing with large scale sparse systems arising from the discretization of differential equations as follows:

$$V_0 = \text{diag}\left(\frac{1}{a_{11}}, \frac{1}{a_{22}}, \dots, \frac{1}{a_{NN}}\right), \tag{6}$$

with a_{ii} as the diagonal elements of A .

Let us now review some of the high order iterative methods for matrix inversion. The perception of the need for higher order methods is the fact that (1) is too slow at the beginning of the process before arriving at the convergence phase for general matrices, and this would increase the computational load of the whole algorithm used for matrix inversion [9].

Li et al. in [10] investigated an iteration of the form

$$V_{n+1} = V_n(3I - AV_n(3I - AV_n)), \quad n = 0, 1, 2, \dots \tag{7}$$

of third-order and also proposed another iterative method for finding A^{-1} of the same order as it is given in

$$V_{n+1} = V_n \left[I + \frac{1}{2} (I - AV_n) (I + (2I - AV_n)^2) \right], \tag{8}$$

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

Note that a general procedure for constructing such methods was given in ([11], Chapter 5). Krishnamurthy and Sen provided the following fourth-order method:

$$V_{n+1} = V_n (I + Y_n (I + Y_n (I + Y_n))), \quad n = 0, 1, 2, \dots, \tag{9}$$

in which $Y_n = I - AV_n$. As another example, a ninth-order method could be presented as

$$V_{n+1} = V_n (I + Y_n (I + Y_n (I + Y_n (I + Y_n (I + Y_n (I + Y_n (I + Y_n (I + Y_n))))))), \quad n = 0, 1, 2, \dots \tag{10}$$

In the sequel, we present a new iteration for matrix inversion. Actually, the following section covers our main contribution as a new ninth-order efficient inverse-finding iterative method. We also prove the main theorem therein.

Next in Section 3, we discuss the complexity of the iterative methods to theoretically find the most efficient method. In Section 4, we analytically discuss the application of the new algorithm in the computation of the Drazin inverse,

which is of interest in numerical analysis. Section 5 applies the suggested iteration in finding robust approximate inverses for large sparse matrices with real or complex entries in details, while the application of the new scheme in preconditioning of the practical problems will be also given. A clear reduction in the computational time to attain the desired accuracy will be observed therein. Finally, in Section 6 our concluding remarks will be furnished.

2. A New Method for Matrix Inversion

A new scheme must be designed by applying an efficient nonlinear (scalar) equation solver to the matrix equation $F(V) = V^{-1} - A$. Then, one may obtain an iterative process using proper factorization. In this way, by applying the iterative scheme

$$\begin{aligned}
 Y_n &= V_n - \frac{F(V_n)}{F'(V_n)}, \\
 Z_n &= V_n - \frac{F(V_n)}{2} \left(\frac{1}{F'(V_n)} + \frac{1}{F'(Y_n)} \right), \\
 V_{n+1} &= Z_n - \frac{F(Z_n)}{2F'(Z_n)} \left(2 + \frac{F''(Z_n)F(Z_n)}{F'(Z_n)^2} \right), \\
 &n = 0, 1, 2, \dots
 \end{aligned}
 \tag{11}$$

on the matrix equation $V^{-1} - A = 0$, we are able to find the fixed-point iteration

$$\begin{aligned}
 V_{n+1} &= -\frac{1}{8}V_n \left(-7I + 9AV_n - 5(AV_n)^2 + (AV_n)^3 \right) \\
 &\times \left(12I - 42AV_n + 103(AV_n)^2 \right)
 \end{aligned}$$

$$\begin{aligned}
 E_{n+1} &= I - AV_{n+1} \\
 &= I - A \left(-\frac{1}{8}V_n (-7I + AV_n (9I + AV_n (-5I + AV_n))) \right. \\
 &\quad \times (12I + AV_n (-7I + AV_n (9I + AV_n (-5I + AV_n)))) \\
 &\quad \left. \times (6I + AV_n (-7I + AV_n (9I + AV_n (-5I + AV_n)))) \right) \\
 &= I - A \left(-\frac{1}{8}V_n (-7I + 9AV_n - 5(AV_n)^2 + (AV_n)^3) \right. \\
 &\quad \times (12I - 42(AV_n)^1 + 103(AV_n)^2 - 156(AV_n)^3 \\
 &\quad \left. + 157(AV_n)^4 - 104(AV_n)^5 + 43(AV_n)^6 - 10(AV_n)^7 + (AV_n)^8 \right) \\
 &= \frac{1}{8}(-2I + AV_n)^3(-I + AV_n)^9 \\
 &= \frac{1}{8}(I + I - AV_n)^3(I - AV_n)^9
 \end{aligned}$$

$$\begin{aligned}
 &- 156(AV_n)^3 + 157(AV_n)^4 - 104(AV_n)^5 \\
 &+ 43(AV_n)^6 - 10(AV_n)^7 + (AV_n)^8).
 \end{aligned}
 \tag{12}$$

Now, by simplification, we suggest the following matrix iteration:

$$\begin{aligned}
 \chi_n &= -7I + \psi_n (9I + \psi_n (-5I + \psi_n)), \\
 \vartheta_n &= \psi_n \chi_n, \\
 V_{n+1} &= -\frac{1}{8}V_n \chi_n (12I + \vartheta_n (6I + \vartheta_n)), \\
 &n = 0, 1, 2, \dots,
 \end{aligned}
 \tag{13}$$

wherein $\psi_n = AV_n$, I is the identity matrix, and the sequence of iterates $\{V_n\}_{n=0}^{\infty}$ converges to A^{-1} under some condition.

In numerical mathematics, it is very useful and essential to know the behavior of an approximate method. Therefore, we are about to prove its order of convergence in Theorem 1.

Theorem 1. Let $A = [a_{ij}] \in \mathbb{C}^{N \times N}$ be a nonsingular complex matrix. If the initial approximation V_0 satisfies

$$\|I - AV_0\| < 1,
 \tag{14}$$

then the iterative method (13) converges with ninth order to A^{-1} .

Proof. We use notations that $E_0 = I - AV_0$ and subsequently $E_n = I - AV_n$. Then,

$$\begin{aligned}
 &= \frac{1}{8}(I + E_n)^3 E_n^9 \\
 &= \frac{1}{8} (E_n^9 + 3E_n^{10} + 3E_n^{11} + E_n^{12}).
 \end{aligned}
 \tag{15}$$

Hence, by taking arbitrary matrix norm on both sides of (15), we attain

$$\|E_{n+1}\| \leq \frac{1}{8} (\|E_n\|^9 + 3\|E_n\|^{10} + 3\|E_n\|^{11} + \|E_n\|^{12}). \tag{16}$$

In addition, since $\|E_0\| < 1$, by relation (16), we obtain that

$$\begin{aligned}
 \|E_1\| &\leq \frac{1}{8} (\|E_0\|^9 + 3\|E_0\|^{10} + 3\|E_0\|^{11} + \|E_0\|^{12}) \\
 &\leq \|E_0\|^9 < 1.
 \end{aligned}
 \tag{17}$$

Now if we consider $\|E_n\| < 1$, therefore

$$\|E_{n+1}\| \leq \frac{1}{8} (\|E_n\|^9 + 3\|E_n\|^{10} + 3\|E_n\|^{11} + \|E_n\|^{12}) \leq \|E_n\|^9. \tag{18}$$

Using mathematical induction, we obtain

$$\|E_{n+1}\| \leq \|E_n\|^9, \quad n \geq 0. \tag{19}$$

Furthermore, we get that

$$\|E_{n+1}\| \leq \|E_n\|^9 \leq \dots \leq \|E_0\|^{9^{n+1}}. \tag{20}$$

That is, $I - AV_n \rightarrow 0$, when $n \rightarrow \infty$ and

$$V_n \rightarrow A^{-1}, \quad \text{as } n \rightarrow \infty. \tag{21}$$

Thus, the new method (13) converges to the inverse of the matrix A in the case $\rho(AV_0) < 1$, where ρ is the spectral radius. Now, we prove that the order of convergence for the sequence $\{V_n\}_{n=0}^{n=\infty}$ is at least nine. Let ε_n denotes the error matrix $\varepsilon_n = A^{-1} - V_n$; afterwards

$$A\varepsilon_n = I - AV_n = E_n. \tag{22}$$

The identity (22) in conjunction with (15) implies that

$$A\varepsilon_{n+1} = \frac{1}{8} ((A\varepsilon_n)^9 + 3(A\varepsilon_n)^{10} + 3(A\varepsilon_n)^{11} + (A\varepsilon_n)^{12}). \tag{23}$$

Therefore, using invertibility of A , it follows immediately that

$$\begin{aligned}
 \varepsilon_{n+1} &= \frac{1}{8} (\varepsilon_n(A\varepsilon_n)^8 + 3\varepsilon_n(A\varepsilon_n)^9 \\
 &\quad + 3\varepsilon_n(A\varepsilon_n)^{10} + \varepsilon_n(A\varepsilon_n)^{11}).
 \end{aligned}
 \tag{24}$$

By taking any subordinate norm of (24), we obtain

$$\begin{aligned}
 \|\varepsilon_{n+1}\| &\leq \left(\frac{1}{8} (\|A\|^8 + 3\|A\|^9 \|\varepsilon_n\| + 3\|A\|^{10} \|\varepsilon_n\|^2 \right. \\
 &\quad \left. + \|A\|^{11} \|\varepsilon_n\|^3) \right) \|\varepsilon_n\|^9.
 \end{aligned}
 \tag{25}$$

Consequently, it is proved that the iterative formula (13) converges to A^{-1} , and the order of this method is at least nine. \square

The Schulz-type iterations are strongly numerically stable, that is, they have the self-correcting characteristic and are essentially based upon matrix multiplication per an iterative step. Multiplication is effectively parallelizable for structured matrices represented in compressed form.

The iterative scheme (13) could efficiently be combined with sparse techniques in order to reduce the computational load of matrix-by-matrix multiplications per step. We should also point out that even if the matrix A is singular, the Schulz-type methods, including the scheme (13), converge to the Moore-Penrose inverse using a proper initial matrix. A full discussion on this feature of this type of iterative methods has been given in [12].

Note that $\{V_n\}_{n=0}^{n=\infty}$ produced from (13), under a certain condition (when $AV_0 = V_0A$), may be applied not only to the left preconditioned linear system $V_nAx = V_nb$ but also to the right preconditioned linear system $AV_ny = b$, where $y = V_nx$. In fact, an important application of the new method (13) is in preconditioning of the linear system of equations. Practically, experimental results in Section 5 will show that the preconditioner obtained from (13) may lead to nicely clustered eigenvalue distributions of the preconditioned matrices and, hence, results in fast convergence of the preconditioned Krylov subspace iteration methods, such as GMRES and BiCGSTAB for solving some classes of large sparse system of linear equations.

3. Complexity of the Methods

Let us consider the computational complexity of the existing iterative processes (1), (7), (8), (9), (10), and (13), since they are all convergent to A^{-1} in the same condition. From a theoretical analysis, and by assuming a uniform cost for the arithmetic operations, typical of the floating point computations, we consider the *inverse-finder informational efficiency index*. It uses two parameters ρ and κ which stand for the rate of convergence and the number of matrix-by-matrix multiplications in floating point arithmetics, respectively. Then the comparative index could be expressed by

$$\text{IEI} = \frac{\rho}{\kappa}. \tag{26}$$

Hence, a favorable method in theoretical point of view must reach an order ρ with fewer matrix multiplications κ , (i.e., $\kappa \leq \rho$).

In Table 2, we furnish a comparison on the order along the number of matrix multiplications, rate of convergence, and the index (26) for different methods. The results show that the new established method in Section 2 is better than the others. In fact, by comparing these results, one can see that the iterative process (13) reduces the computational

TABLE 2: Comparison of the computational complexity for different methods.

Methods	(1)	(7)	(8)	(9)	(10)	(13)
Rate of convergence	2	3	3	4	9	9
Number of matrix multiplications	2	3	4	4	9	7
IIIEI	2/2 = 1	3/3 = 1	3/4 = 0.75	4/4 = 1	9/9 = 1	9/7 ≈ 1.285

complexity by using less number of basic operations and leads to the better equilibrium between the high speed and the operational cost.

4. Application in Finding the Drazin Inverse

In 1958, Drazin in [13] introduced a different kind of generalized inverse in associative rings and semigroups that does not have the reflexivity property but commutes with the element. The importance of this kind of inverse and its computation was later expressed away fully by Wilkinson in [14]. This was the motivation of many authors to develop direct or iterative methods for this important problem; see, for example, [1].

Definition 2. The smallest nonnegative integer k , such that $\text{rank}(A^{k+1}) = \text{rank}(A^k)$, is called the index of A and denoted by $\text{ind}(A)$.

Definition 3. Let A be an $N \times N$ complex matrix; the Drazin inverse of A , denoted by A^D , is the unique matrix V satisfying

$$(1^k) A^k V A = A^k, \quad (2) V A V = V, \quad (5) A V = V A, \tag{27}$$

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

Note that if $\text{ind}(A) = 1$, the matrix V is called the group inverse of A . Also, if A is nonsingular, then it is easily seen that $\text{ind}(A) = 0$ and $A^D = A^{-1}$. Note that the idempotent matrix $A A^D$ is the projector on $\mathcal{R}(A^k)$ along $\mathcal{N}(A^k)$, where $\mathcal{R}(A^k)$ denotes the range of A^k and $\mathcal{N}(A^k)$ is the null space of A^k .

Wei in [15] proved that the general solution of the square singular linear system $Ax = b$ can be obtained using the Drazin inverse as $x = A^D b + (I - A A^D)z$, where $z \in \mathcal{R}(A^{k-1}) + \mathcal{N}(A)$.

In 2004, Li and Wei in [16] proved that the matrix method of Schulz (1) can be used for finding the Drazin inverse of square matrices both possessing real or complex spectra. They proposed the following initial matrix:

$$V_0 = W_0 = \alpha A^l, \quad l \geq \text{ind}(A) = k, \tag{28}$$

where the parameter α must be chosen so that the condition $\|I - A V_0\| < 1$ is satisfied. Using the initial matrix of the form (28) yields to a matrix method for finding the famous Drazin inverse with quadratical convergence.

As a consequence, we could present the iterative method of the form (13) with ninth order of convergence for finding the Drazin inverse, where the initial approximation is chosen as

$$V_0 = W_0 = \frac{2}{\text{Tr}(A^{k+1})} A^k, \tag{29}$$

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

In what follows, we use the following auxiliary results.

Proposition 4 (see [17]). *Let $M \in \mathbb{C}^{N \times N}$ and $\epsilon > 0$ be given. There is at least one matrix norm $\|\cdot\|$ such that*

$$\rho(M) \leq \|M\| \leq \rho(M) + \epsilon, \tag{30}$$

where $\rho(M)$ denotes the spectral radius of M .

Proposition 5 (see [18]). *If $P_{L,M}$ denotes the projector on a space L along a space M , then*

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

Theorem 6. *Let $A \in \mathbb{C}^{N \times N}$ be singular square matrix. Also, suppose that the initial approximation W_0 is chosen by means of (28). Then the sequence $\{W_n\}_{n=0}^{\infty}$ defined by the iterative method (13) satisfies the following error estimate when finding the Drazin inverse:*

$$\frac{\|A^D - W_n\|}{\|A^D\|} \leq \|I - A W_0\|^{9^n}. \tag{31}$$

Proof. Consider the notation $F_0 = I - A W_0$ and subsequently the residual matrix as $F_n = I - A W_n$. Then similarly as in (15), we get

$$\begin{aligned} F_{n+1} &= I - A W_{n+1} \\ &= I - A \left(-\frac{1}{8} W_n (-7I + A W_n (9I + A W_n (-5I + A W_n))) \right) \\ &\quad \times (12I + A W_n (-7I + A W_n (9I + A W_n (-5I + A W_n))) \end{aligned}$$

$$\begin{aligned}
 & \times (6I + AW_n(-7I + AW_n(9I + AW_n(-5I + AW_n)))) \\
 & = \frac{1}{8}(I + I - AW_n)^3(I - AW_n)^9 \\
 & = \frac{1}{8}(F_n^9 + 3F_n^{10} + 3F_n^{11} + F_n^{12}).
 \end{aligned} \tag{32}$$

By taking an arbitrary matrix norm on both sides of (32), we attain

$$\|F_{n+1}\| \leq \frac{1}{8} (\|F_n\|^9 + 3\|F_n\|^{10} + 3\|F_n\|^{11} + \|F_n\|^{12}). \tag{33}$$

In addition, since $\|F_0\| < 1$, by relation (33), we obtain that $\|F_1\| \leq (1/8)(\|F_0\|^9 + 3\|F_0\|^{10} + 3\|F_0\|^{11} + \|F_0\|^{12}) \leq \|F_0\|^9$. Similarly, $\|F_{n+1}\| \leq (1/8)(\|F_n\|^9 + 3\|F_n\|^{10} + 3\|F_n\|^{11} + \|F_n\|^{12}) \leq \|F_n\|^9$. Using mathematical induction, we obtain $\|F_{n+1}\| \leq \|F_n\|^9$ for each $n \geq 0$. This implies that

$$\|F_n\|^9 \leq \|F_0\|^{9^n}, \quad n \geq 0. \tag{34}$$

Since W_0 is chosen as in (28), $\mathcal{R}(W_0) \subseteq \mathcal{R}(A^k)$ immediately follows. Further, using this fact in conjunction with (12), which implies that $\mathcal{R}(W_n) \subseteq \mathcal{R}(W_{n-1})$, we conclude that

$$\mathcal{R}(W_n) \subseteq \mathcal{R}(A^k), \quad n \geq 0. \tag{35}$$

Similarly, if we rewrite (13) in the form

$$\begin{aligned}
 W_{n+1} & = -\frac{1}{8} (-7I + 9W_nA - 5(W_nA)^2 + (W_nA)^3) \\
 & \times (12I - 42AW_n + 103(W_nA)^2 \\
 & - 156(W_nA)^3 + 157(W_nA)^4 \\
 & - 104(W_nA)^5 + 43(W_nA)^6 \\
 & - 10(W_nA)^7 + (W_nA)^8) W_n.
 \end{aligned} \tag{36}$$

it is not difficult to verify that

$$\mathcal{N}(W_n) \supseteq \mathcal{N}(A^k), \quad n \geq 0. \tag{37}$$

Now, an application of the well-known results from [12]

$$AA^D = A^D A = P_{\mathcal{R}(A^k), \mathcal{N}(A^k)} \tag{38}$$

in conjunction with Proposition 5 and (35), (37), immediately follows

$$W_n A A^D = W_n = A^D A W_n, \quad n \geq 0. \tag{39}$$

Therefore, the error matrix $\delta_n = A^D - W_n$ satisfies

$$\delta_n = A^D - W_n = A^D - A^D A W_n = A^D (I - A W_n) = A^D F_n. \tag{40}$$

From the last identity and (34) we have

$$\|\delta_n\| = \|A^D\| \|F_n\| \leq \|A^D\| \|F_0\|^{9^n}, \tag{41}$$

which is a confirmation of (31). □

The following result is a consequence of Theorem 6.

Corollary 7. *If the conditions of Theorem 6 are satisfied and the initial iteration W_0 is chosen such that*

$$\|F_0\| = \|I - A W_0\| < 1, \tag{42}$$

the iterative method (13) converges to A^D .

Therefore, our goal is to find initial approximations W_0 satisfying (42). In accordance with Proposition 4, W_0 must satisfy the following inequality to ensure the convergence in the Drazin inverse case:

$$\max_{1 \leq i \leq t} |1 - \lambda_i(AW_0)| < 1, \tag{43}$$

where $\text{rank}(AW_0) = t$ and $\lambda_i(AW_0)$, $i = 1, \dots, t$ are eigenvalues of AW_0 .

Theorem 8. *Let $A \in \mathbb{C}^{N \times N}$ be a singular square matrix with $\text{ind}(A) = k$, and the sequences F_n, δ_n are defined as in Theorem 6. The sequence $\{W_n\}_{n=0}^{n=\infty}$ defined by the iterative method (13) converges with ninth order to A^D if the initial approximation W_0 is in accordance with (42).*

Proof. Now, by considering $\delta_n = A^D - W_n$ as the error matrix for finding the Drazin inverse, we have

$$\begin{aligned}
 A\delta_{n+1} & = AA^D - AW_{n+1} = AA^D - I + I - AW_{n+1} \\
 & = AA^D - I + F_{n+1}.
 \end{aligned} \tag{44}$$

Taking into account (32) and using elementary algebraic transformations, we further derive

$$\begin{aligned}
 A\delta_{n+1} & = AA^D - I + \frac{1}{8} (F_n^9 + 3F_n^{10} + 3F_n^{11} + F_n^{12}) \\
 & = \frac{1}{8} ((F_n^9 + 3F_n^{10} + 3F_n^{11} + F_n^{12}) + 8(AA^D - I)) \\
 & = \frac{1}{8} ((F_n^9 + AA^D - I) + 3(F_n^{10} + AA^D - I) \\
 & \quad + 3(F_n^{11} + AA^D - I) + (F_n^{12} + AA^D - I)).
 \end{aligned} \tag{45}$$

```
n = 5000; number = 10; SeedRandom [123]
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}];
```

ALGORITHM 1

Now, using the idempotent property $(I - AA^D)^t = (I - AA^D)$, $t \geq 1$, and the following consequence of (39)

$$\begin{aligned} (I - AA^D) A\delta_n &= (I - AA^D) A (A^D - W_n) \\ &= W_n - AA^D W_n = 0, \end{aligned} \tag{46}$$

we obtain for each $t \geq 1$ the following:

$$\begin{aligned} (F_n)^t + AA^D - I &= (I - AW_n)^t + AA^D - I \\ &= (I - AA^D + AA^D - AW_n)^t + AA^D - I \\ &= ((I - AA^D) + A\delta_n)^t + AA^D - I \\ &= I - AA^D + (A\delta_n)^t + AA^D - I \\ &= (A\delta_n)^t. \end{aligned} \tag{47}$$

From (47) and (45),

$$A\delta_{n+1} = \frac{1}{8} ((A\delta_n)^9 + 3(A\delta_n)^{10} + 3(A\delta_n)^{11} + (A\delta_n)^{12}) \tag{48}$$

Therefore,

$$\begin{aligned} \|A\delta_{n+1}\| &\leq \frac{1}{8} (\|A\delta_n\|^9 + 3\|A\delta_n\|^{10} + 3\|A\delta_n\|^{11} + \|A\delta_n\|^{12}) \\ &\leq \|A\delta_n\|^9. \end{aligned} \tag{49}$$

Finally, applying (39), it is now easy to find the error inequality of the new scheme (13) using (49) and the second condition of (27), when finding the Drazin inverse, as follows:

$$\begin{aligned} \|\delta_{n+1}\| &= \|W_{n+1} - A^D\| = \|A^D AW_{n+1} - A^D AA^D\| \\ &= \|A^D (AW_{n+1} - AA^D)\| \\ &\leq \|A^D\| \|A\delta_{n+1}\| \\ &\leq \|A^D\| \|A\|^9 \|\delta_n\|^9. \end{aligned} \tag{50}$$

Therefore, since (42) is satisfied, from (31) follows $\delta_n \rightarrow 0$. Furthermore, the inequalities in (50) immediately lead to the conclusion that $W_n \rightarrow A^D$ as $n \rightarrow +\infty$ with the ninth order of convergence. \square

5. Numerical Aspects

Using the programming package Mathematica 8 [19] in this section, we apply our iterative method on some practical numerical tests and compare it with the existing methods in order to manifest the applicability and the consistency of numerical results with the theoretical aspects illustrated in Sections 2–4.

For numerical comparisons in this section, we have used the methods (1), (7), (8), (9), (10), and (13) denoted by “Schulz”, “Li et al. I”, “Li et al. II”, “KMS4”, “KMS9”, and the “Proposed method”, respectively. We have carried out the numerical tests with machine precision on a computer with Pentium 4. In fact, the computer characteristics are Microsoft Windows XP Intel(R), Pentium(R) 4 CPU, 3.20 GHz with 4 GB of RAM. In all computations, the running time in seconds using `AbsoluteTiming[]` was attained.

In sparse-matrix algebra, the iteration methods such as (1), (7), (8), (9), (10) and (13) 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. This is done herein along with a threshold by applying `Chop[]` on each approximate inverse V_i .

Now, we apply the above inverse finders for finding the approximate inverses of the following large sparse matrices.

Test Problem 1. In this test, 10 large sparse random complex matrices of the dimension 5000 are considered as shown in Algorithm 1.

Note that $I = \sqrt{-1}$. 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. Note that in this test the initial choice has been constructed using different available ways as discussed in Table 1.

The results of comparisons for this test have been presented in Figures 1, 2, 3, and 4. In Figure 1, we show the number of iterations required by different methods to attain the desired accuracy when $V_0 = A/\|A\|_1^2$. As it is obvious that higher order methods require lower number of iterations to converge, we then put our focus on the computational time needed to satisfy the desired tolerance using three different initial approximations $V_0 = A/\|A\|_1^2$, $V_0 = A/\|A\|_\infty^2$, and $V_0 = A/\|A\|_F^2$. As could be observed for all the three forms of the initial guesses and in all the 10 test problems (given in Figures 2–4), our iterative method (13) beats the other existing schemes, which immediately follows the theoretical results of Table 2.

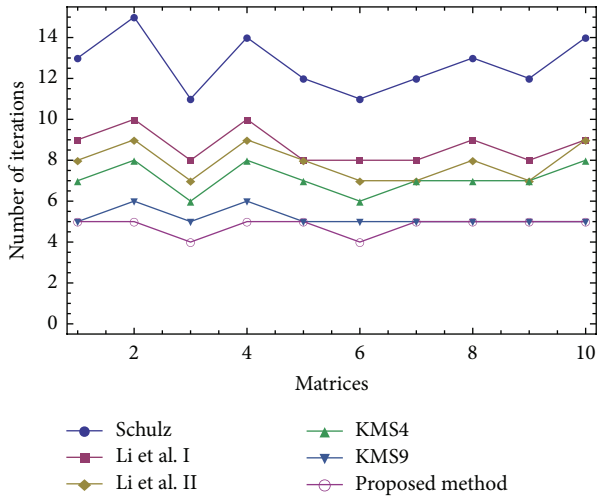


FIGURE 1: Comparison of the number of evaluations for solving the Test Problem 1 using $V_0 = A/\|A\|_1^2$.

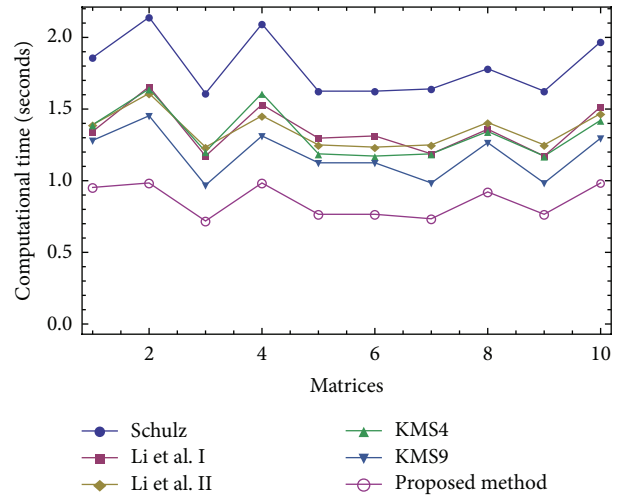


FIGURE 3: Comparison of the elapsed time for solving the Test Problem 1 using $V_0 = A/\|A\|_\infty^2$.

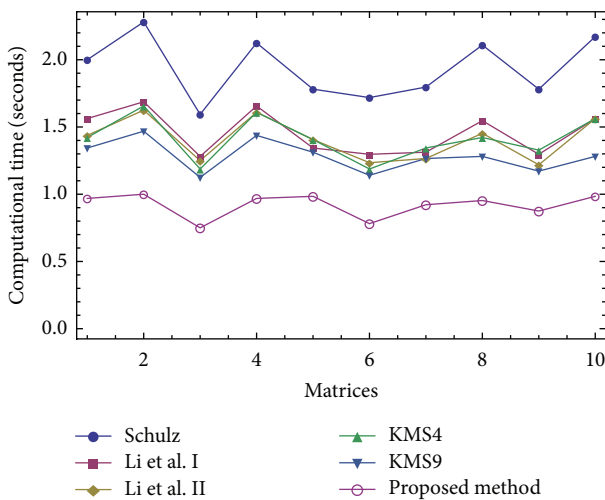


FIGURE 2: Comparison of the elapsed time for solving the Test Problem 1 using $V_0 = A/\|A\|_1^2$.

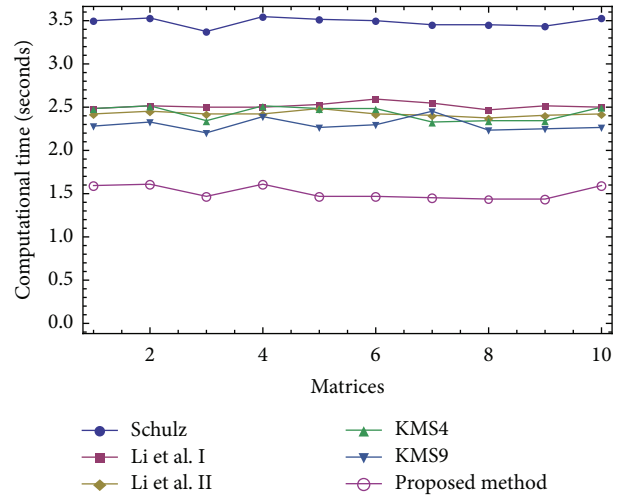


FIGURE 4: Comparison of the elapsed time for solving the Test Problem 1 using $V_0 = A/\|A\|_F^2$.

The attained results have reverified the robustness of the proposed iterative method (13) by a *clear reduction* in the number of iterations and *the elapsed time*.

The practical application of the new scheme (13) falls within many problems as discussed in Section 1. For instance, in solving second-kind integral equations by Wavelet-like approach, the problem will be reduced to find that the inverse of a large sparse matrix possesses a sparse inverse as well [20]. In the rest of this section, we apply our new iterative method as a robust technique to produce accurate preconditioners for accelerating modern iterative solvers such as GMRES or BiCGSTAB for solving large scale sparse linear systems; see, for example, [21].

Test Problem 2. Consider solving the following Boundary Value Problem (BVP) using discretization. In such problems, and in order to capture all the shocks and the behavior of the solution, a very fine grid of points is required in \mathbb{F} approach:

$$\begin{aligned}
 y'' &= 3y - 2y', \quad t \in [a, b], \\
 y(a) &= e^3, \\
 y(b) &= e^{-3},
 \end{aligned}
 \tag{51}$$

where the domain of the solution is $a = 0$ and $b = 2$. To solve (51) using 3-point \mathbb{F} discretization, we assume that in the grid points t_i , the exact solution is denoted by $y(t_i)$ and the approximate solution is defined by w_i , for any $i = 0, 1, \dots, n, n + 1$.


```

h = (2. - 0)/(n + 1); Subscript[t, 0] = 0; Subscript[w, 0] =
E^3; Subscript[w, n + 1] = Exp[-3];
Table[N[(Subscript[w, i - 1] - 2 Subscript[w, i] + Subscript[w, i + 1])/
h^2 - 3 Subscript[w, i] +
2 (Subscript[w, i + 1] - Subscript[w, i - 1])/(2 h) == 0], {i, 1, n}];
    
```

ALGORITHM 2

TABLE 3: Comparison of the computational time in solving the linear system resulting of discretization of (51) when $n = 1500$.

Methods	GMRES	PGMRES-(1)-VI	PGMRES-(7)-IV	PGMRES-(8)-III	PGMRES-(9)-III	PGMRES-(13)-II
Total time	Fail	3.50	1.78	1.53	2.62	1.31

Thus, (51) can be written in discretized form in MATHEMATICA language (see Algorithm 2), wherein n is the number of grid points which result in an $n \times n$ sparse linear system of equations. In this test, we consider the tolerance as 10^{-8} for the final solution of the linear systems. The results of solving this system “without preconditioning” and “with left preconditioning” are given in Tables 3 and 4. Note that in Tables 3 and 4, for example, PBiCGSTAB-(8)-II stands for the left preconditioned linear system ($VAx = Vb$) applying the preconditioner (approximative inverse) obtained by the scheme (8) after 2 iterations, while it is solved by the iterative solver BiCGSTAB.

The numerical results clearly support the efficiency of the method (13). A clear reduction in the *elapsed time* is observable. Even for the case of GMRES solver which had failed (not convergent after 1500 iterations to the considered tolerance), a simple preconditioner obtained by the new method (13) significantly improved the problem. Note that in this test, we have constructed V_0 for the compared methods using (6).

Test Problem 3. The aim of this example is to apply the discussions of Section 4, for finding the Drazin inverse of the following square matrix (taken from [16]):

$$A = \begin{bmatrix} 2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & -2 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 2 & 2 \end{bmatrix}, \tag{52}$$

with $k = \text{ind}(A) = 3$. To simplify the process, we write a general code in the programming package Mathematica for the iterative process (13), to find the (pseudo-)inverse or the Drazin inverse of arbitrary matrices (see Algorithm 3).

The two-argument function `DrazinInverse` [`A`., `tolerance`.] takes the arbitrary matrix A and the tolerance from the user to obtain its Drazin inverse by knowing the index k . In this case, by choosing tolerance $= 10^{-8}$, we obtain

$$A^D = \begin{bmatrix} 0.25 & -0.25 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1.25 & 1.25 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -1.66406 & -0.992187 & 0.25 & -0.25 & 0. & 0. & 0. & 0. & -0.0625 & -0.0625 & 0. & 0.15625 & 0. \\ -1.19531 & -0.679687 & -0.25 & 0.25 & 0. & 0. & 0. & 0. & -0.0625 & 0.1875 & 0.6875 & 1.34375 & 0. \\ -2.76367 & -1.04492 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.48438 & 2.57813 & 3.32031 & 6.64063 & 0. \\ -2.76367 & -1.04492 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.48438 & 2.57813 & 4.57031 & 8.51563 & 0. \\ 14.1094 & 6.30078 & 6.625 & 3.375 & 5. & -3. & -5. & -5. & -4.1875 & -8.5 & -10.5078 & -22.4609 & 0. \\ -19.3242 & -8.50781 & -9.75 & -5.25 & -7.5 & 4.5 & 7.5 & 7.5 & 6.375 & 12.5625 & 15.9766 & 33.7891 & 0. \\ -0.625 & -0.3125 & 0. & 0. & 0. & 0. & 0. & 0. & 0.25 & -0.25 & -0.875 & -1.625 & 0. \\ -1.25 & -0.9375 & 0. & 0. & 0. & 0. & 0. & 0. & -0.25 & 0.25 & -0.875 & -1.625 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1.25 & 1.25 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -0.25 & 0.25 & 0. \end{bmatrix}. \tag{53}$$

TABLE 4: Comparison of the computational time in solving the linear system resulting of discretization of (51) when $n = 2000$.

Methods	BiCGSTAB	PBiCGSTAB-(1)-III	PBiCGSTAB-(7)-III	PBiCGSTAB-(8)-II	PBiCGSTAB-(9)-II	PBiCGSTAB-(13)-I
Total time	1.98	0.53	0.60	0.53	0.54	0.46

```

PM[X_] :=
With[{Id = SparseArray[{{i_, i_} -> 1.}, {n, n}]},
  X1 = A.X; X2 = -7 Id + X1.(9 Id + X1.(-5 Id + X1)); X3 = X1.X2;
  (-1/8) X.X2.(12 Id + X3.(6 Id + X3))
];
InitialMatrix[A_] := 1/SingularValueList[A, 1][[1]]^2 ConjugateTranspose[A];
InitialDrazin[A_] := 2/Tr[MatrixPower[A, k + 1]] MatrixPower[A, k];
DrazinInverse[A_, tolerance_] := If[k == 0,
  Module[{X0 = InitialMatrix[A]}, FixedPoint[(PM[#] &), X0,
    SameTest -> (Norm[#1 - #2, Infinity] <= tolerance &)]],
  Module[{X0 = InitialDrazin[A]}, FixedPoint[(PM[#] &), X0,
    SameTest -> (Norm[#1 - #2, Infinity] <= tolerance &)]
];
    
```

ALGORITHM 3

Checking the conditions of Definition 3 yields to

$$\begin{aligned}
 \|A^{k+1}A^D - A^k\|_\infty &= 1.48415 \times 10^{-12}, \\
 \|A^DAA^D - A^D\|_\infty &= 1.20264 \times 10^{-10}, \\
 \|AA^D - A^DA\|_\infty &= 8.93836 \times 10^{-11},
 \end{aligned}
 \tag{54}$$

which supports the theoretical discussions.

6. Conclusion

In many engineering applications, extracting the diagonal or the whole entries of the inverse of a given matrix (basically large and sparse) is an important part of the computation, for example, in electronic structure calculation and especially for models based on effective one-electron Hamiltonians such as the tight-binding models, or in modern statistics, and thus, developing high-order efficient Schulz-type methods is of practical interest.

In this paper, we have studied a high-order iteration method (13) for matrix inversion. Convergence analysis of our iterative algorithm has been studied and a discussion on the choice of the initial value in order to start the process and preserve the convergence order has been given. We also discussed that under what conditions a new method could be applied for finding the Drazin inverse of square matrices having real or complex spectra.

As a result, the total time consuming of the suggested iteration (13) is remarkably low in contrast with the existing methods of this type in the case of constructing approximate inverses and in preconditioning.

Working on the extension of the proposed method (13) for interval matrix inversion [22] can be considered as future works in this field of study.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

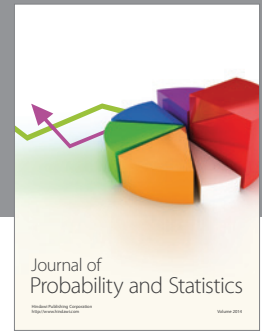
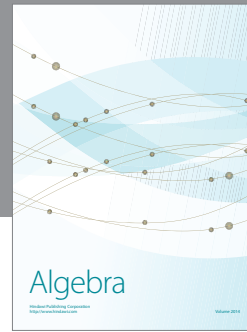
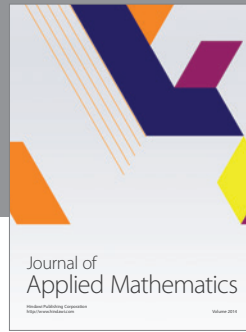
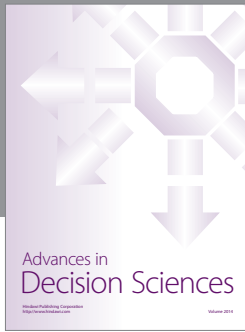
Acknowledgment

The second author gratefully acknowledges support from the Research Project 174013 of the Serbian Ministry of Science.

References

- [1] 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, pp. 7632–7644, 2013.
- [2] R. G. Brow and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1992.
- [3] R. C. Dorf, *The Electrical Engineering Handbook*, CRC Press, Ann Arbor, Mich, USA, 1993.
- [4] G. Schulz, "Iterative Berechnung der Reziproken matrix," *Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, pp. 57–59, 1933.
- [5] J. Rajagopalan, *An iterative algorithm for inversion of matrices [M.S. thesis]*, Concordia University, 1996.
- [6] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, BirkhWauser-Springer, New York, NY, USA, 2001.

- [7] V. Y. Pan, M. Kunin, R. E. Rosholt, and H. Kodai, "Homotopic residual correction processes," *Mathematics of Computation*, vol. 74, no. 253, pp. 345–368, 2006.
- [8] L. Grosz, "Preconditioning by incomplete block elimination," *Numerical Linear Algebra with Applications*, vol. 7, no. 7-8, pp. 527–541, 2000.
- [9] T. Soderstrom and G. W. Stewart, "On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse," *SIAM Journal on Numerical Analysis*, vol. 11, no. 1, pp. 61–74, 1974.
- [10] H.-B. Li, T.-Z. Huang, Y. Zhang, X.-P. Liu, and T.-X. Gu, "Chebyshev-type methods and preconditioning techniques," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 260–270, 2011.
- [11] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms—Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, India, 1986.
- [12] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer, New York, NY, USA, 2nd edition, 2003.
- [13] M. P. Drazin, "Pseudoinverses in associative rings and semi-groups," *American Mathematical Monthly*, vol. 65, pp. 506–514, 1958.
- [14] J. H. Wilkinson, "Note on the practical significance of the Drazin inverse," in *Recent Applications of Generalized Inverses*, S. L. Campbell, Ed., Research Notes in Mathematics, pp. 82–99, Pitman Advanced Publishing Program, Boston, Mass, USA, 1982.
- [15] 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.
- [16] 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.
- [17] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, USA, 1986.
- [18] G. Wang, Y. Wei, and S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, New York, NY, USA, 2004.
- [19] S. Wagon, *Mathematica in Action*, Springer, Berlin, Germany, 3rd edition, 2010.
- [20] 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, pp. 159–184, 1993.
- [21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd edition, 2003.
- [22] J. Rohn and R. Farhadsefat, "Inverse interval matrix: a survey," *Electronic Journal of Linear Algebra*, vol. 22, pp. 704–719, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

