mathematics

Article

A Cubic Class of Iterative Procedures for Finding the Generalized Inverses

Munish Kansal [1], Manpreet Kaur [2], Litika Rani [1] and Lorentz Jäntschi [3,*]

[1] School of Mathematics, Thapar Institute of Engineering and Technology, Patiala 147004, India; munish.kansal@thapar.edu (M.K.); litikabansal92@gmail.com (L.R.)
[2] Department of Mathematics, Lovely Professional University, Phagwara 144411, India; mkaur.math@gmail.com
[3] Department of Physics and Chemistry, Technical University of Cluj-Napoca, Muncii Blvd. No. 103-105, Cluj-Napoca 400641, Romania
* Correspondence: lorentz.jantschi@gmail.com or lorentz.jantschi@ubbcluj.ro

Abstract: This article considers the iterative approach for finding the Moore–Penrose inverse of a matrix. A convergence analysis is presented under certain conditions, demonstrating that the scheme attains third-order convergence. Moreover, theoretical discussions suggest that selecting a particular parameter could further improve the convergence order. The proposed scheme defines the special cases of third-order methods for $\beta = 0, 1/2$, and $1/4$. Various large sparse, ill-conditioned, and rectangular matrices obtained from real-life problems were included from the Matrix-Market Library to test the presented scheme. The scheme's performance was measured on randomly generated complex and real matrices, to verify the theoretical results and demonstrate its superiority over the existing methods. Furthermore, a large number of distinct approaches derived using the proposed family were tested numerically, to determine the optimal parametric value, leading to a successful conclusion.

Keywords: generalized inverse; convergence analysis; Moore–Penrose inverse; order of convergence; singular matrices

MSC: 65F20; 65F10; 15A09; 15A10

Citation: Kansal, M.; Kaur, M.; Rani, L.; Jäntschi, L. A Cubic Class of Iterative Procedures for Finding the Generalized Inverses. *Mathematics* **2023**, *11*, 3031. https://doi.org/10.3390/math11133031

Academic Editor: Qing-Wen Wang

Received: 17 June 2023
Revised: 3 July 2023
Accepted: 6 July 2023
Published: 7 July 2023

1. Introduction

This study presents a generalization of the inverse of rectangular, rank-deficient, or non-singular matrices, as a solution to a specific set of equations. A unique solution for simultaneous equations involving a rectangular or rank-deficient coefficient matrix is determined using the theory of generalized inverses and its various results in matrix algebra. The endeavors of the researchers in this study are not limited to pure theoretical investigations of the generalized inverse but also aim to obtain the deepest knowledge of its characteristics and study its potentail for solving practical problems arising in diverse areas of research. For instance, some of its implementations are discovered for digital image processing [1], control theory [2], detection of neurons in cerebral cortex through a magnetic field [3], machine learning [4], graph theory [5], control theory [6], robotics research [7], and chemical balancing [8].

Back in 1920, Moore published the first article [9] on the study of matrix inverses. Unaware of Moore's work, Sir Roger Penrose [10] presented an analogous description of the same concept in a different format, which was later recognized by Richard Rado [11]. Subsequently, Ben-Israel [12] presented a comprehensive analysis of a matrix inverse in accordance with Penrose's work. The definition of pseudo-inverse, as originally presented by Moore and Penrose, is now commonly known as the Moore–Penrose inverse.

The Moore–Penrose inverse has many uses and connections with key concepts, including eigendecomposition [13], eigenproblems [14], and singular value decomposition [15].

Routes to the Moore–Penrose inverse involve matrix operations, which makes the algorithms part of the parallelizable family.

A number of resource- and time-consuming applications for various topics require obtaining the Moore–Penrose inverse (computer vision [16], control systems [17], data compression [18], data mining [19], language processing [20], linear algebra [21], molecular alignment [22], quantum mechanics [23], recommender systems [24], and signal processing [25]), motivating an interest in the parallelization of the procedures for calculating the generalized inverse.

It is important to mention that the conjugate transpose of a matrix argument is defined by the superscript '$*$'.

Definition 1. *Let M be a complex matrix of order $m \times n$. Then, the Moore–Penrose inverse of M is the matrix $M^\dagger$ satisfying the following Penrose equations:*

$$MM^\dagger M = M, \quad M^\dagger MM^\dagger = M^\dagger, \quad (MM^\dagger)^* = MM^\dagger, \quad (M^\dagger M)^* = M^\dagger M.$$

Over the last few decades, studies on various types of technique for evaluating the Moore–Penrose inverse of a matrix have appeared. For instance, Shinozaki et al. [26] presented a survey and classification of the direct algorithms for computing the Moore–Penrose inverse. Besides this, QR factorization [27], LDL^* decomposition [28], and Gauss-Jordan elimination [29] are well-known direct methods. These approaches generally yield highly accurate results, but they can be computationally expensive and time-consuming, particularly when dealing with large matrices. Therefore, iterative approaches to calculating $M^\dagger$ have been considered as an alternative. One well-known iterative approach is the Schulz method [30,31]:

$$\begin{cases} X_{i+1} = X_i - 2X_iMX_i, \quad i = 0, 1, 2, \dots, \\ X_0 \text{ is the initial approximate to } M^\dagger. \end{cases} \tag{1}$$

The selection of the initial estimate plays a critical role in ensuring the convergence of the iterative algorithms to $M^\dagger$. In order to choose an appropriate initial approximate of the form $X_0 = \alpha M^*$, such that it satisfies the condition $\|I - MX_0\| < 1$, several articles [31–33] provided different ways to calculate the value of α. Here, I denotes the identity matrix of an appropriate size. On the other hand, Li et al. [34] introduced a kth-order family, given as follows:

$$X_{i+1} = X_i\left(kI - \frac{k(k-1)}{2}MX_i + \cdots + (-1)^{k-1}(MX_i)^{k-1}\right), \ k = 2, 3, \dots, \tag{2}$$

for a non-singular matrix M. In addition, two different forms of iterative family, along with their theoretical analysis for computing the outer inverse, were discussed in [35,36] and some particular cases weren analyzed in [37–39]. The study of the Moore–Penrose inverse of a matrix has also been explored for tensors [40–43] and in the field of neural networks [44,45]. Additionally, the representations and characteristics of the Moore–Penrose inverse were presented under certain environments in [46,47]. Evidently, the theory of a generalized inverse is not restricted to theory but has also been implemented in various realistic problems [18,48,49].

In addition to the studies mentioned above, some higher-order algorithms have also been proposed in the literature [33,50]. These higher-order iterative methods are known to yield more accurate solutions, with fewer iterations. However, proposing an iterative scheme of the same order that provides more efficient and effective results is a challenging task.

For instance, in the context of third-order methods, well-known solvers such as the Chebyshev matrix method [51], Homeier's matrix method [51], and the mid-point matrix method [51] have been utilized for finding various types of generalized inverses. However, the development of novel third-order methods that can effectively compete with these established techniques, in terms of accuracy and computational efficiency, presents a significant challenge. Motivated by recent studies in matrix inversion and addressing this challenging aspect, this study aims to establish a new and efficient iterative third-order method.In particular, we contribute to the field by proposing an effective cubic parametric iterative family for finding the Moore–Penrose inverse. By conducting thorough theoretical analyses and rigorous numerical evaluations, we attempt to identify the optimal value for the associated parameter, thus further enhancing the efficacy and efficiency of the proposed method. Furthermore, it is noteworthy that certain existing methods can be derived as special cases of our proposed method for specific choices of the free parameter.

With this motivation, we propose an iterative family and extend it to find the Moore–Penrose inverse in Section 2. A theoretical analysis is contributed in Section 3, demonstrating the convergence order three under certain conditions and restricted parametric values. Section 4 defines the various existing and new schemes, on the basis of different parametric values. The testing of these schemes is included in Section 5, by using realistic and academic problems for evaluation. Moreover, the proposed scheme is tested for 100 distinct values of β, signifying fruitful results for the choice of β. At the final section, i.e., in Section 6, the concluding remarks based on numerical testing and theoretical results are presented.

2. Iterative Scheme

In this section, we will introduce an algorithm that can be used to evaluate the Moore–Penrose inverse of a given matrix. The approach depends on an iterative process that generates a sequence of matrices, each providing a better approximation of the Moore–Penrose inverse. We begin this iterative process using the following iteration formula:

$$X_{i+1} = X_i\Big(aI + bMX_i + c(MX_i)^2 + d(MX_i)^3\Big), \quad i \geq 0, \tag{3}$$

where M is a given matrix, X_0 is the initial approximation of M^{-1}, and a, b, c, d are real parameters satisfying the condition $a + b + c + d = 1$. In order to minimize the free parameters, we establish the relationship between each parameter and a new free variable, $\beta \in \mathbb{R}$. We consider the values of $a = 3 + \beta$, $b = -3 - 3\beta$, $c = 1 + 3\beta$, $d = -\beta$ such that $a + b + c + d = 1$, and propose the following iterative scheme:

$$X_{i+1} = X_i\Big((3+\beta)I + (-3-3\beta)MX_i + (1+3\beta)(MX_i)^2 - \beta(MX_i)^3\Big), \quad i \geq 0, \tag{4}$$

where M is a given matrix, X_0 is the initial estimate of M^{-1}, and β is any real parameter.

The lemma presented below will be helpful in the subsequent theoretical analysis.

Lemma 1. *For a given initial matrix $X_0 = \alpha M^*$, where α is an appropriate real number, the sequence $\{X_i\}$ generated by scheme (4) satisfies for all $i \geq 0$,*

$$\mathbf{MP_1}: \ M^\dagger M X_i = X_i,$$
$$\mathbf{MP_2}: \ X_i M M^\dagger = X_i,$$
$$\mathbf{MP_3}: \ (X_i M)^* = X_i M,$$
$$\mathbf{MP_4}: \ (M X_i)^* = M X_i.$$

Proof. With the help of mathematical induction, we will prove these results. Clearly, $\mathbf{MP_1}$ and $\mathbf{MP_2}$ are valid for $i = 0$, that is $M^\dagger M X_0 = (M^\dagger M)(\alpha M^*) = \alpha(M^\dagger M)M^* = \alpha(M^\dagger M)^* M^* = \alpha(MM^\dagger M)^* = \alpha M^* = X_0$ and $X_0 M M^\dagger = \alpha M^* M M^\dagger = \alpha M^*(MM^\dagger)^* = \alpha(MM^\dagger M)^* = \alpha M^* = X_0$.

Assume that $\mathbf{MP_1}$ is true for some i, we will prove the result for $i+1$ using (4) as

$$\begin{aligned} M^\dagger M X_{i+1} &= M^\dagger M X_i\Big((3+\beta)I - (3+3\beta)MX_i + (1+3\beta)(MX_i)^2 - \beta(MX_i)^3\Big) \\ &= X_i\Big((3+\beta)I - (3+3\beta)MX_i + (1+3\beta)(MX_i)^2 - \beta(MX_i)^3\Big) \\ &= X_{i+1}. \end{aligned}$$

In a similar manner, we demonstrate the validity of the second statement $\mathbf{MP_2}$ for $i+1$, assuming it is true for i, as follows:

$$\begin{aligned} X_{i+1}MM^{\dagger} &= X_i\Big((3+\beta)I-(3+3\beta)MX_i+(1+3\beta)(MX_i)^2-\beta(MX_i)^3\Big)MM^{\dagger} \\ &= (3+\beta)X_iMM^{\dagger}-(3+3\beta)X_iMX_iMM^{\dagger}+(1+3\beta)X_i(MX_i)^2MM^{\dagger} \\ &\quad -\beta X_i(MX_i)^3MM^{\dagger}. \end{aligned}$$

Using the results $X_i(MX_i)^n=(X_iM)^nX_i$ for $n\geq 0$ and $X_iMM^{\dagger}=X_i$, we have

$$\begin{aligned} X_{i+1}MM^{\dagger} &= (3+\beta)X_iMM^{\dagger}-(3+3\beta)X_iMX_iMM^{\dagger}+(1+3\beta)(X_iM)^2X_iMM^{\dagger} \\ &\quad -\beta(X_iM)^3X_iMM^{\dagger} \\ &= (3+\beta)X_i-(3+3\beta)X_iMX_i+(1+3\beta)(X_iM)^2X_i-\beta(X_iM)^3X_i \\ &= (3+\beta)X_i-(3+3\beta)X_iMX_i+(1+3\beta)X_i(MX_i)^2-\beta X_i(MX_i)^3 \\ &= X_{i+1}. \end{aligned}$$

To demonstrate that $\mathbf{MP_3}$ is true, we first prove the case when $i=0$ as $(X_0M)^*=(\alpha M^*M)^*=\alpha MM^*=M(\alpha M^*)=MX_0$. Furthermore, let us assume that this holds for some i. Now, consider the case for $i+1$, as follows:

$$\begin{aligned} (X_{i+1}M)^* &= \Big(X_i\big((3+\beta)I-(3+3\beta)MX_i+(1+3\beta)(MX_i)^2-\beta(MX_i)^3\big)M\Big)^* \\ &= (3+\beta)(X_iM)^*-(3+3\beta)\Big((X_iM)^2\Big)^*+(1+3\beta)\Big((X_iM)^3\Big)^*-\beta\Big((X_iM)^4\Big)^* \\ &= (3+\beta)(X_iM)-(3+3\beta)(X_iM)^2+(1+3\beta)(X_iM)^3-\beta(X_iM)^4 \\ &= X_i\Big((3+\beta)I-(3+3\beta)X_iM+(1+3\beta)(X_iM)^2-\beta(X_iM)^3\Big)M \\ &= X_{i+1}M. \end{aligned}$$

On similar lines, $\mathbf{MP_4}$ can be proven. This completes the lemma proof. □

3. Convergence Behavior

This section deals with establishing a convergence analysis of the recursive form (4) with the starting value $X_0=\alpha M^*$. The following theorem indicates that for a limited free parameter β and under certain conditions, the sequence produced by the scheme in (4) converges to $M^{\dagger}$.

Theorem 1. *Let $M\in\mathbb{C}_r^{m\times n}$, the initial estimate $X_0=\alpha M^*$ for arbitrary real number α and $X=M^{\dagger}$ such that the residual $F_0=(X_0-X)M$ satisfies $\|F_0\|<1$. Then, the sequence of approximations generated by (4) for $\beta\in[0,1]$ converges to $M^{\dagger}$. Moreover, it has third-order convergence and fourth-order convergence for $\beta\in[0,1)$ and $\beta=1$, respectively.*

Proof. To prove the first part of the theorem, it suffices to demonstrate that $\|X_{i+1}-X\|$ approaches 0 as n approaches infinity. This can be accomplished by employing the properties of the Moore–Penrose inverse M and using Lemma 1, resulting in

$$\|X_{i+1}-X\|=\|X_{i+1}MX-XMX\|\leq\|X_{i+1}M-XM\|\|X\|. \tag{5}$$

By using (4), one can derive

$$\begin{aligned} X_{i+1}M - XM =& \Big((3+\beta)X_i - (3+3\beta)X_iMX_i + (1+3\beta)X_i(MX_i)^2 \\ & - \beta X_i(MX_i)^3 \Big) M - XM \\ =&(1-\beta)\Big((X_iM)^3 - 3(X_iM)^2 + 3X_iM - XM \Big) \\ & - \beta\Big((X_iM)^4 - 4(X_iM)^3 + 6(X_iM)^2 - 4(X_iM) + XM \Big) \\ =&(1-\beta)(X_iM - XM)^3 - \beta(X_iM - XM)^4. \end{aligned}$$

Therefore, the sequence of residual matrices $F_i = X_iM - XM$ satisfies the recurrence relation:

$$F_{i+1} = (1-\beta){F_i}^3 - \beta {F_i}^4. \quad (6)$$

We will apply mathematical induction to prove the convergence of sequence $\{X_i\}$. Specifically, we will demonstrate that $\|F_i\| \to 0$ as $i \to \infty$, which establishes the desired convergence result. It is clear that $\|F_0\| < 1$, and thus this holds for $i = 0$. Now, assume that for some i, $\|F_i\| < 1$. To prove the inductive step $i+1$, we consider the norm of Equation (6) for $\beta \in [0, 1)$, which yields the result:

$$\|F_{i+1}\| \leq (1-\beta)\|F_i\|^3 + \beta\|F_i\|^4 < (1-\beta)\|F_i\|^3 + \beta\|F_i\|^3 < \|F_i\|^3 < \|F_0\|^{3^{i+1}}. \quad (7)$$

However, for $\beta = 1$, we obtain

$$\|F_{i+1}\| \leq \|F_i\|^4 < \|F_0\|^{4^{i+1}}. \quad (8)$$

Therefore, as $i \to \infty$, Equations (7) and (8) imply that $\|F_i\| \to 0$, which completes the convergence proof, i.e., $X_i \to X$ as $i \to \infty$.

Now, to determine the convergence order of the proposed technique, we define the error estimate at step i as $E_i = X_i - X$, where X is the exact solution. Then, using (4), we arrive at the following expression for the error matrix at $i+1$ step:

$$\begin{aligned} E_{i+1} =&(3+\beta)E_i - (2+\beta)E_iMX - (2+\beta)XME_i + (1+\beta)XME_iMX + (1+\beta) \\ & E_iME_iMX + (1+\beta)XME_iME_i - (2+\beta)E_iME_i - \beta XME_iME_iMX + (1 \\ & + \beta)E_iME_iME_i - \beta E_iME_iME_iMX - \beta XME_iME_iME_i - \beta E_iME_iME_iME_i. \end{aligned}$$

Thus, one can determine the following errors:

$$\begin{aligned} Error_1 &= (3+\beta)E_i - (2+\beta)E_iMX - (2+\beta)XME_i + (1+\beta)XME_iMX, \\ Error_2 &= (1+\beta)E_iME_iMX + (1+\beta)XME_iME_i - (2+\beta)E_iME_i - \beta XME_iME_iMX, \\ Error_3 &= (1+\beta)E_iME_iME_i - \beta E_iME_iME_iMX - \beta XME_iME_iME_i, \\ Error_4 &= -\beta E_iME_iME_iME_i. \end{aligned}$$

Using $E_i = X_i - X$ and Lemma 1, one can achieve

$$\begin{aligned} Error_1 &= 0, \\ Error_2 &= 0, \\ Error_3 &= (1-\beta)E_iME_iME_i, \\ Error_4 &= -\beta E_iME_iME_iME_i. \end{aligned}$$

This completes the proof and demonstrates that scheme (4) converges with order three for $0 \leq \beta < 1$ and four for $\beta = 1$. □

Theorem 2. *Iterative scheme (4) with the initial estimate $X_0 = \alpha M^*$ results in the following relation*

$$\lim_{i\to+\infty} \frac{\Omega_{i+1}}{\Omega_i^3} = 1-\beta. \tag{9}$$

Proof. From the recurrence relation (6), one can derive the following inequalities using the norm

$$\begin{aligned} F_{i+1} &= (1-\beta)F_i^3 - \beta F_i^4 \\ \|F_{i+1}\| &\geq \|(1-\beta)F_i^3\| - \beta\|F_i^4\| \\ &\geq (1-\beta)\|F_i\|^3 - \beta\|F_i\|^4. \end{aligned} \tag{10}$$

Let $\Omega_i = \|F_i\|$, the inequality (10) yields

$$\begin{aligned} \Omega_{i+1} &\geq (1-\beta)\Omega_i^3 - \beta\Omega_i^4 \\ \frac{\Omega_{i+1}}{\Omega_i^3} &\geq (1-\beta) - \beta\Omega_i. \end{aligned} \tag{11}$$

On the other hand, we can obtain

$$\begin{aligned} \Omega_{i+1} = \|F_{i+1}\| &\leq \|(1-\beta)F_i^3\| + \beta\|F_i^4\| \\ &\leq (1-\beta)\|F_i\|^3 + \beta\|F_i\|^4 \\ &\leq (1-\beta)\Omega_i^3 + \beta\Omega_i^4 \\ \frac{\Omega_{i+1}}{\Omega_i^3} &\leq (1-\beta) + \beta\Omega_i. \end{aligned} \tag{12}$$

Consequently, the two preceding inequalities lead to

$$(1-\beta) - \beta\Omega_i \leq \frac{\Omega_{i+1}}{\Omega_i^3} \leq (1-\beta) + \beta\Omega_i. \tag{13}$$

As the norm of F_i, denoted by Ω_i, approaches zero, the Theorem 1 permits us to deduce by taking a limit for the Equation (13), where $\frac{\Omega_{i+1}}{\Omega_i^3} \to 1-\beta$ as i approaches infinity. Thus, the theorem is proven. □

4. Variants of the New Family (4)

By introducing different values for the parameter β in the proposed iterative matrix method (4), one can define various new methods to solve different problems. For instance, some of the existing methods used for finding different types of generalized inverse for a specified matrix can be derived from the proposed method. Furthermore, one can also develop entirely new methods for different parametric values β and analyze the algorithm's performance. This can lead to the discovery of more efficient and accurate problem-solving methodologies across diverse fields of study. In light of this, we derived the following cases that pertain to this novel technique, as well as established techniques, for computing generalized matrix inverses.

Case 1: For $\beta = 0$, the technique (4) corresponds to the widely known third-order Chebyshev matrix scheme [51], which is defined as follows:

$$\textbf{CM} \qquad X_{i+1} = X_i\Big(3I - 3MX_i + (MX_i)^2\Big), \quad i \geq 0. \tag{14}$$

Case 2: The method developed in [51], which is an extension of Homeier's method [52], is obtained from (4) for $\beta = \frac{1}{2}$, as demonstrated below:

$$\textbf{HM} \qquad X_{i+1} = X_i\left(I + \frac{1}{2}(I - MX_i)(I + (2I - MX_i)^2)\right). \tag{15}$$

Case 3: By selecting $\beta = \frac{1}{4}$ in Equation (4), we obtain the matrix method given by [51], which is derived from the mid-point method [52] and is read as

$$\textbf{MP} \qquad X_{i+1} = X_i\left(I + \frac{1}{4}(I - MX_i)(3I - MX_i)^2\right). \tag{16}$$

Case 4: When $\beta = 1$, the scheme proposed in Equation (4) can be interpreted as a fourth order hyperpower method [53]:

$$\textbf{HP4} \qquad X_{i+1} = X_i\left(4I - 6MX_i + 4(MX_i)^2 - (MX_i)^3\right). \tag{17}$$

Case 5: A new algorithm can be derived by incorporating $\beta = \frac{9}{10}$ into the matrix family (4):

$$\textbf{NM1} \qquad X_{i+1} = X_i\left(3.9I - 5.7MX_i + 3.7(MX_i)^2 - 0.9(MX_i)^3\right). \tag{18}$$

Case 6: Another new matrix method can be derived when $\beta = \frac{4}{5}$ from (4):

$$\textbf{NM2} \qquad X_{i+1} = X_i\left(3.8I - 5.4MX_i + 3.4(MX_i)^2 - 0.8(MX_i)^3\right). \tag{19}$$

In a similar manner, we can establish several new and distinct third-order iterative schemes for matrix inverse by selecting different values of β that fulfill the conditions of Theorem 1.

5. Numerical Testing

In this section, we examined the convergence and efficiency of the proposed method using various test matrices. The scheme was evaluated by applying it to randomly selected real and complex matrices, as well as some practical problems. The performance of the scheme was determined using several criteria, including the computational time (Time) measured in seconds, number of iterations (i), convergence order ρ, and several norms, such as: $e_1 = \|MXM - M\|$, $e_2 = \|XMX - M\|$, $e_3 = \|(MX)^* - MX\|$, $e_4 = \|(XM)^* - XM\|$. For each test matrix, we fixed the initial guess $X_0 = \dfrac{1}{\|M\|_2{}^2} M^*$ and the stopping criteria, $\max\{\|MXM - M\|, \|XMX - M\|, \|(MX)^* - MX\|, \|(XM)^* - XM\|\} < \text{tol}$. We considered two distinct values of 'tol' depending on the size of the testing matrix. Additionally, the approximate value of the computational convergence order was determined using the following formulae:

$$\rho \approx \frac{\ln(\|X_{i+1} - X_i\| / \|X_i - X_{i-1}\|)}{\ln(\|X_i - X_{i-1}\| / \|X_{i-1} - X_{i-2}\|)}, \; i = 2, 3, \ldots. \tag{20}$$

The numerical values used for comparison purposes were calculated utilizing `Mathematica` software [54], version 11. In the resulting tables, the expression of the form $a(\pm b)$ denotes $a \times 10^{\pm b}$.

Example 1. *Consider a randomly generated real matrix of size* 100×101 *using built-in commands, as follows:*

```
SeedRandom[123];
M=RandomReal[{-2, 2}, {100, 101}];
```

The results obtained from a randomly generated matrix of size 100×101 *with a tolerance value of* 10^{-50} *are presented in Table 1. Analysis of the results reveals that the newly proposed methods, labeled as* ***NM1*** *and* ***NM2****, outperformed the existing third-order iterative approaches* ***CM****,* ***MP****, and* ***HM*** *in terms of providing highly accurate solutions. Furthermore, the new methods were able to achieve the required level of accuracy in a comparatively shorter amount of time.*

Table 1. Experimental data obtained from iterative methods, for Example 1.

Method	i	e_1	e_2	e_3	e_4	ρ	Time
SM [30]	21	1.6 (−55)	5.6 (−54)	0	0	2.0001	277.203
CM [51]	14	1.1 (−124)	3.7 (−123)	0	0	3.0000	194.312
MP [51]	13	3.4 (−88)	1.1 (−86)	0	0	3.0000	194.325
HM [51]	12	8.7 (−57)	3.0 (−55)	0	0	3.0000	179.031
NM1 (18)	12	1.4 (−147)	4.7 (−146)	0	0	3.0000	174.985
NM2 (19)	12	1.5 (−115)	1.9 (−113)	0	0	3.0000	171.470
HP4 [53]	11	1.6 (−109)	5.3 (−108)	0	0	4.0000	155.702

Example 2. *Consider a randomly generated complex matrix of order* 100×101 *using built-in commands, as follows:*

```
SeedRandom[123];
M=RandomReal[{-2+I, 2+I}, {100, 101}];
```

The findings of the study are summarized in Table 2, using tol = 10^{-50}. *The* ***CM*** *method required more iterations compared to other third-order methods to achieve an enforced exactness of the solution. On the other hand, the* ***NM1*** *method exhibited a superior performance in terms of iterations, accuracy, and time compared to the third-order techniques.*

Table 2. Experimental data obtained from iterative methods, for Example 2.

Method	i	e_1	e_2	e_3	e_4	ρ	Time
SM [30]	25	7.4 (−90)	1.2 (−88)	0	0	2.0000	1644.86
CM [51]	16	6.6 (−115)	1.1 (−113)	0	0	3.0000	1680.99
MP [51]	15	1.2 (−94)	1.9 (−93)	0	0	3.0000	1153.19
HM [51]	14	7.8 (−69)	1.2 (−67)	0	0	3.0000	941.094
NM1 (18)	13	1.4 (−70)	4.7 (−69)	0	0	3.0000	793.515
NM2 (19)	13	2.6 (−53)	4.2 (−52)	0	0	3.0000	817.546
HP4 [53]	13	2.2 (−178)	3.4 (−177)	0	0	4.0000	831.734

Example 3. *Consider the partial differential equation (pde):*

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad (0 < x < 1,\ 0 < t \leq 0.1), \tag{21}$$

satisfying the initial condition $U = \sin \pi x$, *when* $t = 0$ *for* $0 \leq x \leq 1$, *and the boundary condition* $U = 0$, *at* $x = 0$ *and* 1 *for* $t > 0$. *Our objective was to obtain the approximate* U *using finite-difference methods. Specifically, we applied the Crank–Nicolson implicit method on Equation (21) to evaluate* U *at* n *points. This procedure resulted in the following approximated equation:*

$$\frac{U_{i,j+1} - U_{i,j}}{k} = \frac{1}{2}\left\{\frac{U_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}\right\},$$

implies

$$-rU_{i-1,j+1} + (2+2r)U_{i,j+1} - rU_{i+1,j+1} = rU_{i-1,j} + (2-2r)U_{i,j} + rU_{i+1,j},$$

where $r = k/h^2$*. To this end, we took step sizes* $h = 0.1$ *and* $k = 0.01$*, and obtained the following linear system:*

$$MU = b, \tag{22}$$

where

$$M = \begin{bmatrix} B_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ B_2 & B_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & B_2 & B_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & B_2 & B_1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 \end{bmatrix},$$

$\mathbf{0}$ *is a zero matrix of order* 9×9*,*

$$B_2 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix},$$

$U = [U_{1,1}, U_{2,1}, U_{3,1}, U_{4,1}, \ldots, U_{9,1}, U_{1,2}, U_{2,2}, U_{2,3}, \ldots, U_{2,9}, \ldots, U_{1,10}, U_{2,10}, \ldots, U_{9,10}]^t$, *and* $b = [\sin 0.2\pi, \sin 0.1\pi + \sin 0.3\pi, \sin 0.2\pi + \sin 0.4\pi, \sin 0.3\pi + \sin 0.5\pi, \sin 0.4\pi + \sin 0.6\pi, \sin 0.5\pi + \sin 0.7\pi, \sin 0.6\pi + \sin 0.8\pi, \sin 0.7\pi + \sin 0.9\pi, \sin 0.8\pi, 0, 0, \ldots, 0]^t$, *where t signifies to transpose of matrix or vector.*

*In order to check the applicability of the proposed iterative methods (**NM1** and **NM2**) for solving PDE* (21)*, we examined the resulting linear system* (22)*. The numerical outcomes were calculated using the coefficient matrices with a tolerance of* $\tau = 10^{-50}$ *and are displayed in Table* 3*. The experimental data revealed that the proposed scheme provided superior results in comparison to the existing methods of the same order for each considered parametric value. In addition, the final approximate values of U up to four decimal places obtained using the method **NM1** were equal to* $\big[0.2802, 0.5329, 0.7335, 0.8623, 0.9067, 0.8623, 0.7335, 0.5329, 0.2802, 0.2540, 0.4832, 0.6651, 0.7818, 0.8221, 0.7818, 0.6651, 0.4832, 0.2540, 0.2303, 0.4381, 0.6030, 0.7089, 0.7453, 0.7089, 0.6030, 0.4381, 0.2303, 0.2088, 0.3972, 0.5467, 0.6427, 0.6758, 0.6427, 0.5467, 0.3972, 0.2088, 0.1893, 0.3602, 0.4957, 0.5827, 0.6127, 0.5827, 0.4957, 0.3602, 0.1893, 0.1717, 0.3265, 0.4494, 0.5284, 0.5556, 0.5284, 0.4494, 0.3265, 0.1717, 0.1557, 0.2961, 0.4075, 0.4791, 0.5037, 0.4791, 0.4075, 0.2961, 0.1557, 0.1411, 0.2684, 0.3695, 0.4344, 0.4567, 0.4345, 0.3695, 0.2684, 0.1411, 0.1280, 0.2434, 0.3350, 0.3938, 0.4141, 0.3938, 0.3350, 0.2434, 0.1280, 0.1160, 0.2207, 0.3037, 0.3571, 0.3754, 0.3571, 0.3037, 0.2207, 0.1160\big]$*. Overall, we can conclude that the developed scheme can be used as a better alternative to the existing cubic-convergent iterative methods.*

Table 3. Experimental data obtained from iterative methods, for Example 3.

Method	i	e_1	e_2	e_3	e_4	ρ	Time
SM [30]	16	1.7 (−90)	9.2 (−90)	0	0	2.0000	128.719
CM [51]	10	1.2 (−81)	6.5 (−81)	0	0	3.0068	63.071
MP [51]	10	5.3 (−131)	2.8 (−130)	0	0	3.0015	66.297
HM [51]	9	1.1 (−67)	6.1 (−67)	0	0	3.0671	109.813
NM1 (18)	9	2.4 (−134)	1.3 (−133)	0	0	3.0589	54.781
NM2 (19)	9	2.7 (−111)	1.5 (−110)	0	0	3.0493	61.610
HP4 [53]	8	1.7 (−90)	9.2 (−90)	0	0	4.1557	54.875

In addition to validating the proposed scheme for accuracy, we investigated the computational convergence behavior of the newly developed third-order iterative schemes and compared them with existing schemes reported in the literature. The comparison is presented in Figure 1, which was drawn using Examples 1–3. These plots demonstrate the performance of the various iterative approaches in terms of computational order of convergence with respect to the number of iterations.

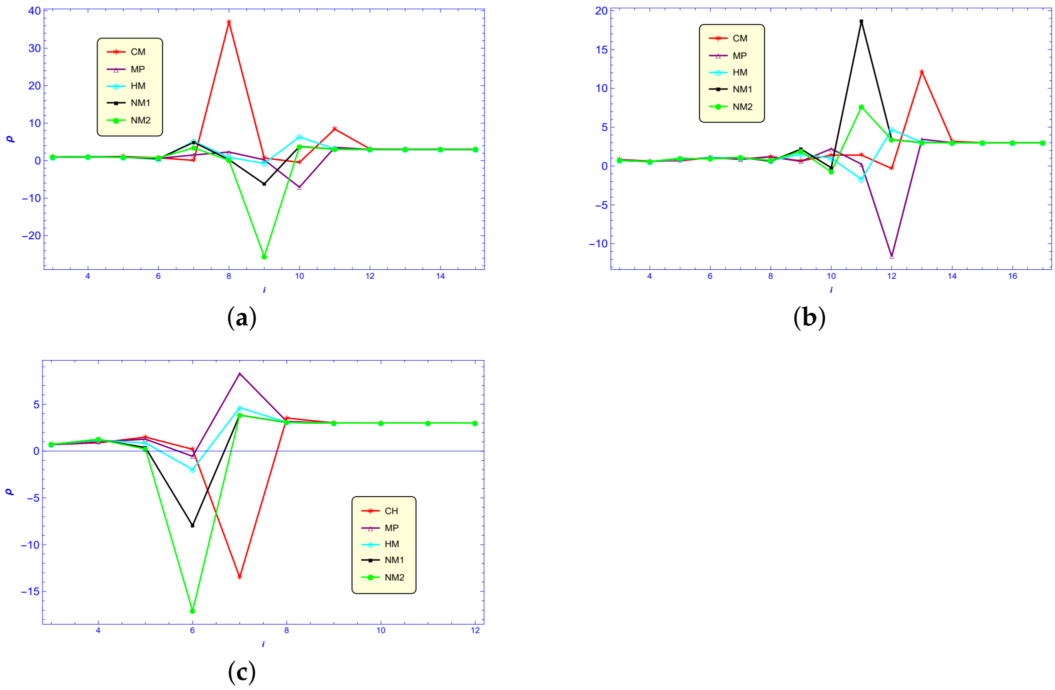


Figure 1. Iterations (i) versus computational convergence order (ρ). (**a**) Example 1. (**b**) Example 2. (**c**) Example 3.

Figure 1a shows that the **CM**, **MP**, **HM**, **NM1**, and **NM2** approaches reached the convergence phase after 12, 11, 11, 10, and 10 iterations, respectively. Figure 1b,c further demonstrate that the developed iterative procedure achieved theoretical convergence order relatively earlier than the others. Furthermore, as evidenced by the data presented in Tables 1–3, the performance of **NM1** and **NM2** was superior in terms of both convergence phase and solution accuracy compared to the **CM**, **MP**, and **HM** approaches in each of the considered examples.

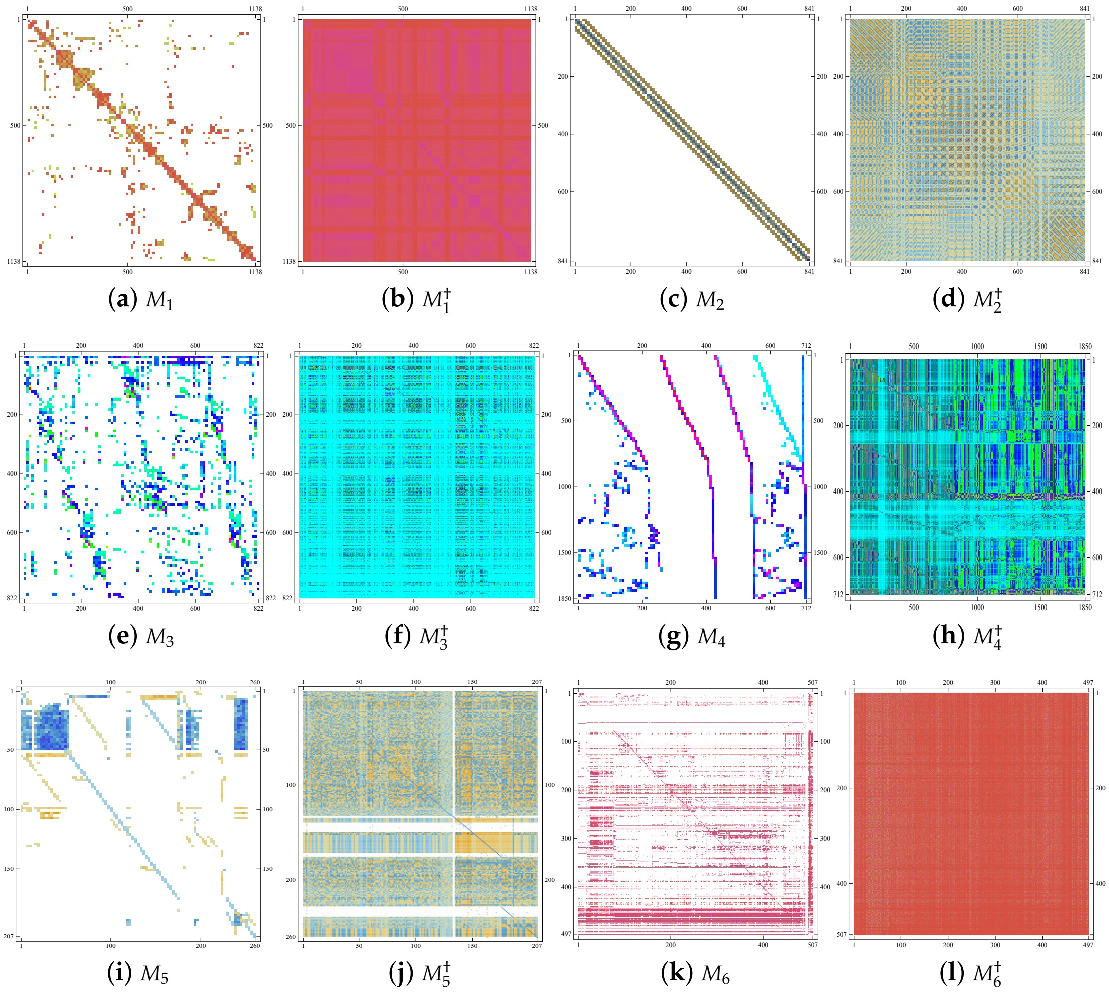
Example 4. *To investigate the applicability of the presented matrix methods to real-world problems, we examined various mathematical models available in the Matrix-Market Library [55]. We evaluated the performance of the developed methods on the matrices mentioned in Table 4, which included different types of matrices, such as ill-conditioned square matrices, rectangular matrices, and rank-deficient matrices. Using the same initial guess and stopping criteria with a tolerance of* 10^{-5}*, a comparison of different methods was derived. The corresponding results are displayed in Table 5,* *where the third and last columns clearly indicate that the new method's implementations were comparatively more satisfying than the other conventional iterative techniques. Additionally, the representation of the considered matrices listed in Table 4 and their corresponding Moore–Penrose inverse are visualized in Figure 2.*

Table 4. Information of matrices considered from the Matrix-Market Library [55].

$M_\#$	Name of Problem	Description
M_1	1138 BUS	Order: (1138, 1138), rank =1138, condition number (est.): 1(+2)
M_2	YOUNG1C	Order: (841, 841), rank = 841, condition number (est.): 2.9(+2)
M_3	BP_600	Order: (822, 822), rank = 822, condition number (est.): 5.1(+06)
M_4	ILLC1850	Order: (1850, 712), rank = 712
M_5	WM3	Order: (207, 260), rank = 207
M_6	BEAUSE	Order: (497, 507), rank = 459

Table 5. Performance of iterative methods for the different matrices defined in Table 4.

$M_\#$	Method	i	e_1	e_2	e_3	e_4	Time
M_1	**SM** [30]	51	6.5 (−7)	4.1 (−10)	1.2 (−10)	1.3 (−6)	113.688
	CM [51]	32	6.0 (−7)	3.3 (−9)	1.3 (−10)	1.6 (−6)	71.687
	MP [51]	30	5.3 (−7)	9.7 (−10)	1.2 (−10)	1.3 (−6)	68.750
	HM [51]	28	7.8 (−7)	1.6 (−6)	1.6 (−10)	1.6 (−6)	63.297
	NM1 (18)	26	7.5 (−7)	2.4 (−9)	1.7 (−10)	1.3 (−6)	51.437
	NM2 (19)	27	5.2 (−7)	4.2 (−10)	1.2 (−10)	1.6 (−6)	53.750
	HP4 [53]	26	5.3 (−7)	4.3 (−10)	1.2 (−10)	1.4 (−6)	42.548
M_2	**SM** [30]	21	5.8 (−6)	4.5 (−6)	3.7 (−14)	2.7 (−13)	47.203
	CM [51]	14	9.7 (−12)	7.7 (−13)	3.5 (−14)	2.9 (−13)	34.015
	MP [51]	13	1.5 (−10)	1.2 (−10)	3.7 (−10)	2.7 (−13)	31.344
	HM [51]	12	9.6 (−8)	7.4 (−8)	4.6 (−14)	2.5 (−13)	28.749
	NM1 (18)	11	1.2 (−7)	9.4 (−8)	3.6 (−14)	3.4 (−13)	21.016
	NM2 (19)	11	6.3 (−6)	4.9 (−6)	3.8 (−14)	2.9 (−13)	21.781
	HP4 [53]	11	3.0 (−11)	2.3 (−11)	3.8 (−14)	3.6 (−13)	20.843
M_3	**SM** [30]	46	8.4 (−12)	8.8 (−11)	1.2 (−12)	3.3 (−11)	131.891
	CM [51]	29	1.3 (−11)	1.9 (−10)	1.1 (−12)	2.3 (−11)	57.125
	MP [51]	27	1.6 (−11)	3.3 (−8)	1.6 (−12)	2.0 (−11)	49.343
	HM [51]	26	1.8 (−11)	1.9 (−12)	1.3 (−12)	1.9 (−10)	48.515
	NM1 (18)	24	1.7 (−11)	4.1 (−12)	1.6 (−12)	4.2 (−11)	25.845
	NM2 (19)	24	1.9 (−11)	2.8 (−9)	1.5 (−12)	2.0 (−10)	25.844
	HP4 [53]	23	1.5 (−11)	9.0 (−11)	1.4 (−12)	4.3 (−11)	26.984
M_4	**SM** [30]	26	3.0 (−13)	6.3 (−12)	8.7 (−13)	3.9 (−12)	117.156
	CM [51]	16	5.1 (−13)	2.2 (−7)	7.1 (−13)	2.7 (−12)	67.656
	MP [51]	15	1.1 (−12)	4.6 (−7)	7.1 (−13)	2.9 (−12)	66.499
	HM [51]	15	3.0 (−13)	1.3 (−11)	9.9 (−13)	2.7 (−12)	66.439
	NM1 (18)	13	2.0 (−12)	8.7 (−7)	7.4 (−13)	3.0 (−12)	54.313
	NM2 (19)	14	3.4 (−13)	4.3 (−12)	8.1 (−13)	4.2 (−12)	56.781
	HP4 [53]	13	6.1 (−13)	1.0 (−11)	9.3 (−13)	2.4 (−12)	55.281
M_5	**SM** [30]	27	4.1 (−14)	3.3 (−11)	1.2 (−13)	7.6 (−14)	3.009
	CM [51]	17	4.5 (−14)	9.8 (−11)	4.7 (−14)	8.7 (−14)	2.094
	MP[51]	16	4.8 (−14)	4.8 (−11)	7.0 (−14)	6.8 (−14)	2.048
	HM [51]	15	1.3 (−13)	2.4 (−9)	6.0 (−14)	7.1 (−14)	1.922
	NM1 (18)	14	5.0 (−14)	2.5 (−12)	5.5 (−14)	9.4 (−14)	1.890
	NM2 (19)	14	1.3 (−12)	2.4 (−8)	8.1 (−14)	1.2 (−13)	1.891
	HP4 [53]	14	3.9 (−14)	1.8 (−13)	7.1 (−14)	8.5 (−14)	1.806
M_6	**SM** [30]	36	2.1 (−12)	2.0 (−9)	1.7 (−12)	1.1 (−12)	18.656
	CM [51]	23	1.9 (−12)	2.0 (−19)	1.8 (−12)	1.0 (−12)	12.688
	MP [51]	21	2.6 (−12)	2.4 (−7)	1.8 (−12)	9.9 (−13)	12.094
	HM [51]	20	2.1 (−12)	2.0 (−9)	2.2 (−12)	1.1 (−12)	12.094
	NM1 (18)	19	3.2 (−12)	3.9 (−9)	1.7 (−12)	1.2 (−12)	11.652
	NM2 (19)	19	1.7 (−12)	1.8 (−9)	1.7 (−12)	1.7 (−12)	11.922
	HP4 [53]	18	1.7 (−12)	1.3 (−9)	1.2 (−12)	1.2 (−12)	10.982



(**a**) M_1 (**b**) $M_1^\dagger$ (**c**) M_2 (**d**) $M_2^\dagger$

(**e**) M_3 (**f**) $M_3^\dagger$ (**g**) M_4 (**h**) $M_4^\dagger$

(**i**) M_5 (**j**) $M_5^\dagger$ (**k**) M_6 (**l**) $M_6^\dagger$

Figure 2. Visual representations of matrices and their Moore–Penrose inverses.

Example 5. *Let us consider the rectangular matrix*

$$M = \begin{pmatrix} 5 & 1 & 1 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}. \tag{23}$$

The exact pseudoinverse of M is

$$M^\dagger = \begin{pmatrix} 1/5 & -1/25 & -1/25 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/5 & 0 \end{pmatrix}. \tag{24}$$

In this example, the numerical results listed in Table 6 were computed with a tolerance of $tol = 10^{-1000}$. The purpose of evaluating a higher accurate solution is to discuss the behavior of residual norms of third-order iterative schemes with an increase in iterations, as shown in Figure 3. It is important to note that the figure results were obtained under the same environment for the initial guess and stopping criteria. Moreover, the computational results were evaluated by fixing the 3000 significant digits, to minimize the round-off errors and enhance the computing speed.

Table 6. Experimental data obtained from iterative methods, for Example 5.

Method	i	e_1	e_2	e_3	e_4	ρ	Time
SM [30]	12	2.1 (−1497)	1.1 (−1498)	0	0	2.000	0.562
CM [51]	8	1.7 (−2398)	8.9 (−2400)	0	0	3.0000	0.500
MP [51]	8	1.2 (−2673)	6.5 (−2675)	0	0	3.0000	0.548
HM [51]	7	3.0 (−1009)	1.6 (−1010)	0	0	3.0000	0.470
NM1 (18)	7	5.3 (−1359)	2.8 (−1360)	0	0	3.0000	0.453
NM2 (19)	7	2.6 (−1229)	1.4 (−1230)	0	0	3.0000	0.480
HP4 [53]	6	2.1 (−1497)	1.1 (−1498)	0	0	4.0000	0.484

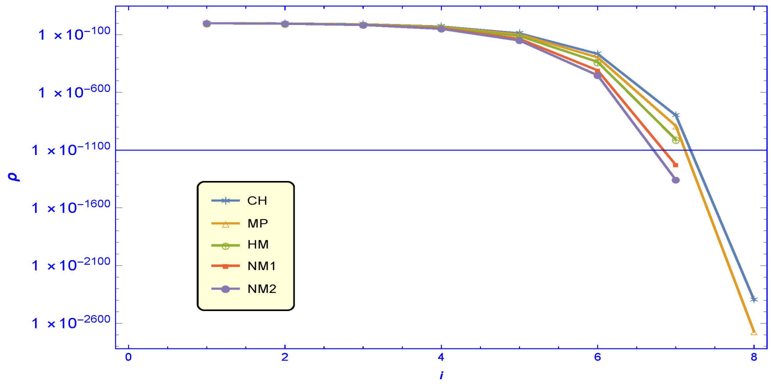


Figure 3. Number of iterations versus residual norms of cubic order methods, for Example 5.

According to the computational data listed in Tables 1–6, a clear conclusion can be drawn that **NM1** and **HP4** exhibited the best performances in each aspect of the comparisons. Undoubtedly, the **HP4** method demonstrated equivalent or, in some cases, superior results compared to **NM1** and **NM2**. However, in comparison to the cubic order convergence methods, the new method demonstrated more favorable outcomes.

Study of Different Parametric Values

In this subsection, we conducted a comprehensive analysis of the behavior of the proposed scheme by varying the value of the parameter β. To achieve this, we utilized selected test matrices to evaluate the performance of the scheme under different values of β. We considered a range of values for β from 0 to 1, with a step size of 0.01. This allowed us to systematically investigate the scheme's performance under different parameter values, providing valuable insights into how the scheme behaves and performs in various scenarios.

Example 6. *Let us consider a randomly generated real matrix of order* 10×11 *denoted as* $M1$*, which was generated using the following code:*

```
SeedRandom[123]
M1=RandomReal[{-2, 2}, {10, 11}];
```
(25)

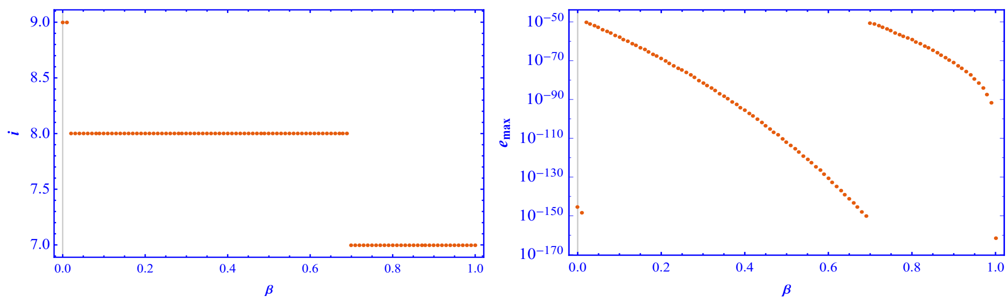
We also refer to the matrix M *defined in Example 5, and in this subsection, this is denoted by* $M2$

$$M2 = \begin{pmatrix} 5 & 1 & 1 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}. \quad (26)$$

The iterations and computational time of the proposed scheme for the matrix $M1$ defined in (25) were determined under the same initial guess and stopping criteria, with a tolerance of 10^{-50}. The obtained experimental data are illustrated in Figures 4 and 5. Similarly, the behavior of the matrix $M2$ defined in (26) with a tolerance of 10^{-100} is shown in Figures 6 and 7. Note that the maximum error norm $e_{\max} = \max\{e_1, e_2, e_3, e_4\}$. Based on the observations from these figures, the following conclusions can be drawn:

1. The graph of the number of iterations shows that, as the value of β increased, the number of iterations did not necessarily decrease. In fact, it can be observed that the presented scheme used fewer iterations for values of β close to one compared to values of β close to zero, indicating that the scheme converged faster for higher values of β.
2. To achieve a more precise matrix inverse, the maximum error norm should be lower. However, it was observed that the scheme (4), which resulted in fewer iterations as depicted in Figures 4 and 6, corresponded to a higher error norm. Nevertheless, when the accuracy of the solutions obtained from each iterative method was evaluated for a particular iteration, it was found that the scheme with β that required fewer iterations yielded a comparatively more accurate and precise matrix inverse.
3. On the other hand, the same trend did not necessarily hold for a computational time, due to fluctuations. For example, for matrix $M1$ of (25), the time taken for computation with a β close to one was comparatively less than the β near to zero. However, such behavior of β was not observed for the matrix $M2$ in (26). Therefore, the computational time varied depending on the characteristics of the matrices used.

In summary, it can be concluded that the scheme (4) demonstrated a greater efficiency for values of β near to one compared to values of β close to zero.



(**a**) Number of iterations (**b**) Maximum error

Figure 4. Number of iterations and corresponding maximum error norm $e_{\max}$ attained by the proposed scheme for different parametric values β for the matrix defined in (25).

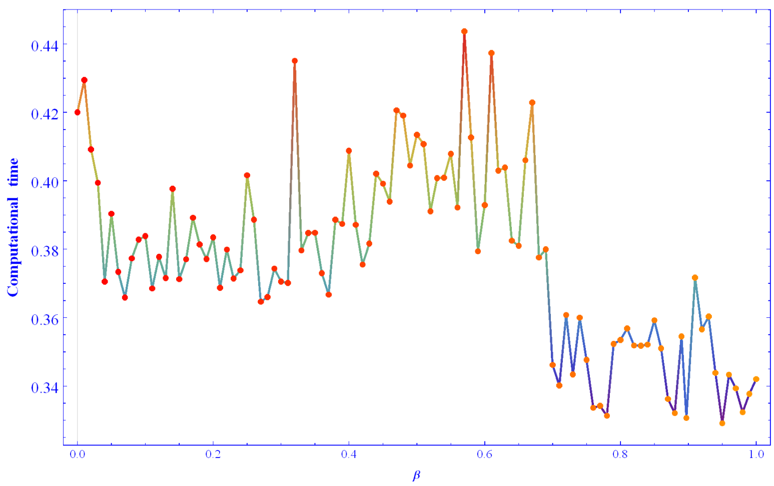
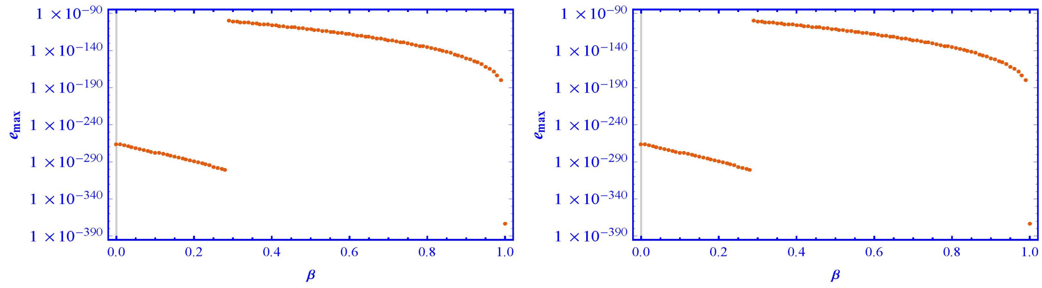


Figure 5. Computational time used by the proposed scheme for different parametric values β for the matrix defined in (25).



(**a**) Maximum error (**b**) Maximum error

Figure 6. Number of iterations and corresponding maximum error norms $e_{\max}$ attained by the proposed scheme for different parametric values β for the matrix defined in (26).

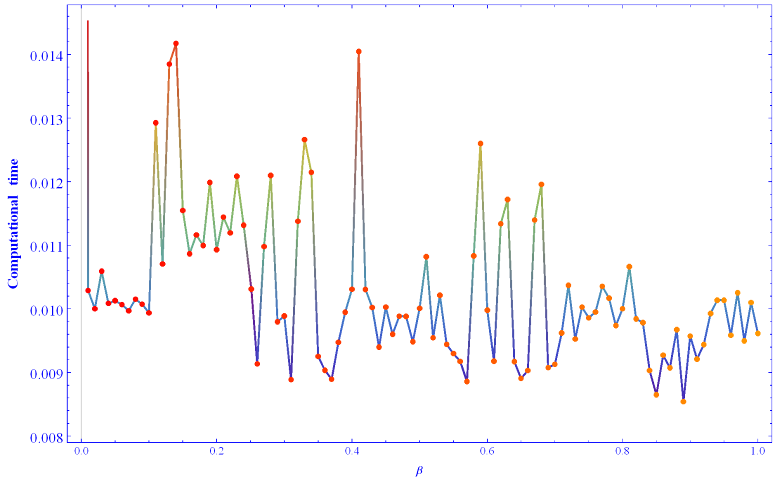


Figure 7. Computational time used by the proposed scheme for different parametric values the β for matrix defined in (26).

6. Conclusions

This paper presented an iterative scheme for obtaining the Moore–Penrose inverse of a given complex matrix. The behavior of the proposed scheme was thoroughly analyzed and investigated in this study. The theoretical analysis showed that under specific conditions and with a restricted parametric value, the new scheme converged to $M^{\dagger}$ with a third-order convergence rate. The existing three- and four-order schemes could be defined for a specific parameter. Nevertheless, we aimed to identify the best parametric value to define a comparatively efficient third-order method. We demonstrated through numerical investigations that the proposed scheme had a superior accuracy, despite not being as efficient as the other methods in terms of a higher efficiency index. Different types of matrices, such as random real and complex matrices, realistic problems, ill-conditioned matrices, a larger sparse matrix, and academic problems were inspected, to authenticate the validity of the new scheme. Based on numerical analysis, we concluded that the presented scheme yielded more accurate results as the value of β gradually increased towards one. In conclusion, the proposed iterative scheme is a viable method for obtaining the Moore–Penrose inverse of a complex matrix, and it can be applied to various practical problems in mathematics, engineering, and other fields. Several possible directions for further research can be described. The presented work could be extended to include a stability analysis of the proposed techniques. This would include exploring the stability properties and performance bounds of the presented methods. Furthermore, one could investigate the characteristics of the proposed scheme for computation of the Drazin inverse and Bott–Duffin inverse.

Author Contributions: M.K. (Munish Kansal): Conceptualization; methodology; validation; supervision; M.K. (Manpreet Kaur): writing—original draft preparation; software; investigation; L.R.: software; L.J.: formal analysis; resources; data curation; writing—review and editing; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to sincerely thank the referees for their valuable comments and suggestions.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Stanimirović, P.S.; Chountasis, S.; Pappas, D.; Stojanović, I. Removal of blur in images based on least squares solutions. *Math. Methods Appl. Sci.* **2013**, *36*, 2280–2296. [CrossRef]
2. Meister, S.; Stockburger, J.T.; Schmidt, R.; Ankerhold, J. Optimal control theory with arbitrary superpositions of waveforms. *J. Phys. A Math. Theor.* **2014**, *47*, 495002. [CrossRef]
3. Wang, J.Z.; Williamson, S.J.; Kaufman, L. Magnetic source imaging based on the minimum-norm least-squares inverse. *Brain Topogr.* **1993**, *5*, 365–371. [CrossRef] [PubMed]
4. Lu, S.; Wang, X.; Zhang, G.; Zhou, X. Effective algorithms of the Moore–Penrose inverse matrices for extreme learning machine. *Intell. Data Anal.* **2015**, *19*, 743–760. [CrossRef]
5. Pavlíková, S.; Ševčovič, D. On the Moore–Penrose pseudo-inversion of block symmetric matrices and its application in the graph theory. *Linear Algebra Appl.* **2023**, *673*, 280–303. [CrossRef]
6. Feliks, T.; Hunek, W.P.; Stanimirović, P.S. Application of generalized inverses in the minimum-energy perfect control theory. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 4560–4575. [CrossRef]
7. Doty, K.L.; Melchiorri, C.; Bonivento, C. A theory of generalized inverses applied to robotics. *Int. J. Robot. Res.* **1993**, *12*, 1–19. [CrossRef]
8. Soleimani, F.; Stanimirović, P.S.; Soleymani, F. Some matrix iterations for computing generalized inverses and balancing chemical equations. *Algorithms* **2015**, *8*, 982–998. [CrossRef]
9. Moore, E.H. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.* **1920**, *26*, 394–395.
10. Penrose, R. A generalized inverse for matrices. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [CrossRef]
11. Rado, R. Note on generalized inverses of matrices. *Math. Proc. Camb. Philos. Soc.* **1956**, *52*, 600–601. [CrossRef]
12. Ben-Israel, A. Generalized inverses of matrices: A perspective of the work of Penrose. *Math. Proc. Camb. Philos. Soc.* **1986**, *100*, 407–425. [CrossRef]
13. Lee, M.; Kim, D. On the use of the Moore–Penrose generalized inverse in the portfolio optimization problem. *Finance Res. Lett.* **2017**, *22*, 259–267. [CrossRef]
14. Kučera, R.; Kozubek, T.; Markopoulos, A.; Machalová, J. On the Moore–Penrose inverse in solving saddle-point systems with singular diagonal blocks. *Numer. Linear Algebra Appl.* **2012**, *19*, 677–699. [CrossRef]
15. Kyrchei, I. Weighted singular value decomposition and determinantal representations of the quaternion weighted Moore–Penrose inverse. *Appl. Math. Comput.* **2017**, *309*, 1–16. [CrossRef]
16. Long, J.; Peng, Y.; Zhou, T.; Zhao, L.; Li, J. Fast and Stable Hyperspectral Multispectral Image Fusion Technique Using Moore–Penrose Inverse Solver. *Appl. Sci.* **2021**, *11*, 7365. [CrossRef]
17. Zhuang, G.; Xia, J.; Feng, J.E.; Wang, Y.; Chen, G. Dynamic compensator design and Hinf admissibilization for delayed singular jump systems via Moore–Penrose generalized inversion technique. *Nonlinear Anal. Hybrid Syst.* **2023**, *49*, 101361. [CrossRef]
18. Zhang, W.; Wu, Q.M.J.; Yang, Y.; Akilan, T. Multimodel Feature Reinforcement Framework Using Moore–Penrose Inverse for Big Data Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5008–5021. [CrossRef]
19. Castaño, A.; Fernández-Navarro, F.; Hervás-Martínez, C. PCA-ELM: A Robust and Pruned Extreme Learning Machine Approach Based on Principal Component Analysis. *Neural Process. Lett.* **2013**, *37*, 377–392. [CrossRef]
20. Lauren, P.; Qu, G.; Zhang, F.; Lendasse, A. Discriminant document embeddings with an extreme learning machine for classifying clinical narratives. *Neurocomputing* **2018**, *277*, 129–138. [CrossRef]
21. Koliha, J.; Djordjević, D.; Cvetković, D. Moore–Penrose inverse in rings with involution. *Linear Algebra Its Appl.* **2007**, *426*, 371–381. [CrossRef]
22. Jäntschi, L. The Eigenproblem Translated for Alignment of Molecules. *Symmetry* **2019**, *11*, 1027. [CrossRef]
23. Baksalary, O.M.; Trenkler, G. The Moore–Penrose inverse: A hundred years on a frontline of physics research. *Eur. Phys. J. H* **2021**, *46*, 9. [CrossRef]

24. Hornick, M.; Tamayo, P. Extending Recommender Systems for Disjoint User/Item Sets: The Conference Recommendation Problem. *IEEE Trans. Know. Data Eng.* **2012**, *24*, 1478–1490. [CrossRef]
25. Chatterjee, S.; Thakur, R.S.; Yadav, R.N.; Gupta, L. Sparsity-based modified wavelet de-noising autoencoder for ECG signals. *Signal Process.* **2022**, *198*, 108605. [CrossRef]
26. Shinozaki, N.; Sibuya, M.; Tanabe, K. Numerical algorithms for the Moore–Penrose inverse of a matrix: Direct methods. *Ann. Inst. Stat. Math.* **1972**, *24*, 193–203. [CrossRef]
27. Katsikis, V.N.; Pappas, D.; Petralias, A. An improved method for the computation of the Moore–Penrose inverse matrix. *Appl. Math. Comput.* **2011**, *217*, 9828–9834. [CrossRef]
28. Stanimirović, I.P.; Tasić, M.B. Computation of generalized inverses by using the LDL^* decomposition. *Appl. Math. Lett.* **2012**, *25*, 526–531. [CrossRef]
29. Stanimirović, P.S.; Petković, M.D. Gauss—Jordan elimination method for computing outer inverses. *Appl. Math. Comput.* **2013**, *219*, 4667–4679. [CrossRef]
30. Schultz, G. Iterative berechung der reziproken matrix. *ZAMM Z. Angew. Math. Mech.* **1933**, *13*, 57–59. [CrossRef]
31. Ben-Israel, A.; Greville, T.N. *Generalized Inverses: Theory and Applications*; Springer: New York, NY, USA, 2003.
32. Pan, V.; Schreiber, R. An improved Newton iteration for the generalized inverse of a matrix, with applications. *SIAM J. Sci. Statist. Comput.* **1991**, *12*, 1109–1130. [CrossRef]
33. Kaur, M.; Kansal, M.; Kumar, S. An Efficient Matrix Iterative Method for Computing Moore–Penrose Inverse. *Mediterr. J. Math.* **2021**, *18*, 1–21. [CrossRef]
34. Li, W.; Li, Z. A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix. *Appl. Math. Comput.* **2010**, *215*, 3433–3442. [CrossRef]
35. Petković, M.D. Generalized Schultz iterative methods for the computation of outer inverses. *Comput. Math. Appl.* **2014**, *67*, 1837–1847. [CrossRef]
36. Stanimirović, P.S.; Soleymani, F. A class of numerical algorithms for computing outer inverses. *J. Comput. Appl. Math.* **2014**, *263*, 236–245. [CrossRef]
37. Liu, X.; Jin, H.; Yu, Y. Higher-order convergent iterative method for computing the generalized inverse and its application to Toeplitz matrices. *Linear Algebra Appl.* **2013**, *439*, 1635–1650. [CrossRef]
38. Soleymani, F.; Stanimirović, P.S. A higher order iterative method for computing the Drazin inverse. *Sci. World J.* **2013**, *2013*, 708647. [CrossRef]
39. Soleymani, F.; Stanimirović, P.S.; Ullah, M.Z. An accelerated iterative method for computing weighted Moore–Penrose inverse. *Appl. Math. Comput.* **2013**, *222*, 365–371. [CrossRef]
40. Sun, L.; Zheng, B.; Bu, C.; Wei, Y. Moore–Penrose inverse of tensors via Einstein product. *Linear Multilinear Algebra* **2016**, *64*, 686–698. [CrossRef]
41. Ma, H.; Li, N.; Stanimirović, P.S.; Katsikis, V.N. Perturbation theory for Moore–Penrose inverse of tensor via Einstein product. *Comput. Appl. Math.* **2019**, *38*, 111. [CrossRef]
42. Liang, M.; Zheng, B. Further results on Moore–Penrose inverses of tensors with application to tensor nearness problems. *Comput. Math. Appl.* **2019**, *77*, 1282–1293. [CrossRef]
43. Huang, B. Numerical study on Moore–Penrose inverse of tensors via Einstein product. *Numer. Algorithms* **2021**, *87*, 1767–1797. [CrossRef]
44. Zhang, Y.; Yang, Y.; Tan, N.; Cai, B. Zhang neural network solving for time-varying full-rank matrix Moore–Penrose inverse. *Computing* **2011**, *92*, 97–121. [CrossRef]
45. Wu, W.; Zheng, B. Improved recurrent neural networks for solving Moore–Penrose inverse of real-time full-rank matrix. *Neurocomputing* **2020**, *418*, 221–231. [CrossRef]
46. Miao, J.M. General expressions for the Moore–Penrose inverse of a 2×2 block matrix. *Linear Algebra Appl.* **1991**, *151*, 1–15. [CrossRef]
47. Kyrchei, I.I. Determinantal representations of the Moore–Penrose inverse over the quaternion skew field and corresponding Cramer's rules. *Linear Multilinear Algebra* **2011**, *59*, 413–431. [CrossRef]
48. Wojtyra, M.; Pekal, M.; Fraczek, J. Utilization of the Moore–Penrose inverse in the modeling of overconstrained mechanisms with frictionless and frictional joints. *Mech. Mach. Theory* **2020**, *153*, 103999. [CrossRef]
49. Zhuang, H.; Lin, Z.; Toh, K.A. Blockwise Recursive Moore–Penrose Inverse for Network Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 3237–3250. [CrossRef]
50. Sharifi, M.; Arab, M.; Haghani, F.K. Finding generalized inverses by a fast and efficient numerical method. *J. Comput. Appl. Math.* **2015**, *279*, 187–191. [CrossRef]
51. Li, H.B.; Huang, T.Z.; Zhang, Y.; Liu, X.P.; Gu, T.X. Chebyshev-type methods and preconditioning techniques. *Appl. Math. Comput.* **2011**, *218*, 260–270. [CrossRef]
52. Chun, C. A geometric construction of iterative functions of order three to solve nonlinear equations. *Comput. Math. Appl.* **2007**, *53*, 972–976. [CrossRef]
53. Altman, M. An optimum cubically convergent iterative method of inverting a linear bounded operator in Hilbert space. *Pac. J. Math.* **1960**, *10*, 1107–1113. [CrossRef]

54. Trott, M. *The Mathematica Guidebook for Programming*; Springer: New York, NY, USA, 2013.
55. Matrix Market. Available online: https://math.nist.gov/MatrixMarket (accessed on 24 April 2023).