An iterative method for computing the approximate inverse of a square matrix and the Moore–Penrose inverse of a non-square matrix

F. Toutounian\textsuperscript{a,b}, F. Soleymani\textsuperscript{b,*}

\textsuperscript{a} The Center of Excellence on Modelling and Control Systems, Ferdowsi University of Mashhad, Mashhad, Iran
\textsuperscript{b} Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

\textbf{ Article Info}

\textbf{Keywords:}
Moore–Penrose
Singular matrix
Approximate inverse
GMRES method
Rectangular matrix
Preconditioning

\textbf{Abstract}

In this paper, an iterative scheme is proposed to find the roots of a nonlinear equation. It is shown that this iterative method has fourth order convergence in the neighborhood of the root. Based on this iterative scheme, we propose the main contribution of this paper as a new high-order computational algorithm for finding an approximate inverse of a square matrix. The analytical discussions show that this algorithm has fourth-order convergence as well. Next, the iterative method will be extended by theoretical analysis to find the pseudo-inverse (also known as the Moore–Penrose inverse) of a singular or rectangular matrix. Numerical examples are also made on some practical problems to reveal the efficiency of the new algorithm for computing a robust approximate inverse of a real (or complex) matrix.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

In numerical analysis and engineering applications, it is frequently required to solve the large sparse linear system

$$Ax = b, \quad x, b \in \mathbb{R}^n,$$

where $A \in \mathbb{R}^{n \times n}$, is a large sparse real matrix. The direct solvers for finding the solution of (1) are expensive because a large amount of work and storage is required. Iterative methods which combine preconditioning techniques are among the most efficient ways for solving (1). More precisely, iterative methods usually involve a second matrix that transforms the coefficient matrix into one with a more favorable spectrum. The transformation matrix is called a preconditioner. Without a preconditioner, an iterative method may have a poor convergence or even fail to converge. If $M$ is a nonsingular matrix that approximates the inverse of $A$ ($M \approx A^{-1}$), then the transformed linear system $AMy = b, x = My$ will have the same solution as the system (1), but the convergence rate of iterative methods applied to the preconditioned system might be higher. This system is preconditioned from the right, but left preconditioning is also possible, i.e., $MAx = Mb$. Generally speaking, the preconditioner $M$ should be chosen so that $AM$ or $MA$ are a good approximation of the identity matrix [13]. However, it is not easy to find this kind of preconditioner for the large sparse linear systems.

In 1933, Schulz in [15] presented an efficient and stable method for computing an approximate inverse of the square matrices as follows:

$$V_{l+1} = V_l(2I - AV_l), \quad l = 0, 1, 2, \ldots,$$

* Corresponding author.

E-mail addresses: toutouni@math.um.ac.ir (F. Toutounian), fazlollah.soleymani@gmail.com (F. Soleymani).
where $I$ is an identity matrix with the same order as the matrix $A$, and $V_0$ is an initial approximation for $A^{-1}$. This iterative scheme was applied to compute the approximate inverse preconditioners $M$ for solving the linear system (1) in [10].

In 2006, Saberi Najafi and Shams Solary developed (2) for the block matrices in [14]. They also stated that the method converges quadratically under the condition: $\|I - AV_0\| < 1$, where $\| \cdot \|$ is any subordinate matrix norm.

In 2010, Li and Li [11] proposed

$$V_{l+1} = V_l(3I - 3AV_l + (AV_l)^2), \quad l = 0, 1, 2, \ldots, \quad (3)$$

where $I$ is the identity matrix. They proved that the iterative method (3) has third-order convergence and satisfies the error inequality $\|e_{l+1}\| \leq |A|^2\|e_l\|^3$, where $e_l = A^{-1} - V_l$. In addition, the authors presented a family of formulas as follows:

$$V_{l+1} = V_l\left[ k - \frac{k(k-1)}{2}AV_l + \cdots + (AV_l)^{k-1} \right], \quad k = 2, 3, \ldots, \quad (4)$$

They showed that under the condition $\|I - AV_0\| < 1$, the iterative formula (4) is convergent to $A^{-1}$ and the order of convergence for this formula is $k$.

In 2011, based on the Chebyshev's method [17], Li et al. [10] obtained the iterative process (3). Then, by using the midpoint rule method, and the Homeier's method, they also obtained the iterative processes

$$V_{l+1} = \left[ I + \frac{1}{4}(I - VA)(3I - VA)^2 \right] V_l, \quad l = 0, 1, 2, \ldots, \quad (5)$$

and

$$V_{l+1} = V_l\left[ I + \frac{1}{2}(I - AV_l)(I + (2I - AV_l)^2) \right], \quad l = 0, 1, 2, \ldots, \quad (6)$$

respectively, for computing an approximate inverse of a matrix $A$. They stated again that under the condition $\|I - AV_0\| < 1$, the iterative formulas (5) and (6) are at least cubic convergent to $A^{-1}$.

Motivated by the research in this field and also inspired by the recent works and computational algorithms found in [3], we first present a new fourth-order iterative scheme to solve the nonlinear equation $f(x) = 0$. Then, by using this iterative scheme, we obtain a sequence of approximations $\{V_l\}$ for calculating the inverse of a matrix $A$. We show that under some conditions, this sequence is at least fourth-order convergent to $A^{-1}$. Furthermore, the new method will be extended to find the generalized inverse of a rectangular matrix (the Moore–Penrose inverse) via theoretical analysis.

The rest of this paper is organized as follows. Section 2 is devoted to present a new iterative scheme for the solution of nonlinear functions. In Section 3, based on this iterative scheme, we obtain a new fourth-order computational algorithm for finding an approximate inverse of a square matrix. The convergence analysis is carried out to establish the order of convergence. Discussions on finding the pseudo-inverse using the new iterative method will be carried out in Section 4. In Section 5, some numerical examples are given to show the performance of the presented method compared with some known second and third order methods and also to illustrate the application of the new method in preconditioning of partial differential equations (PDEs). Section 5 gives further new finding for obtaining the local convergence orders numerically. Finally, we offer some concluding remarks in Section 6.

2. An iterative method for finding the solution of $f(x) = 0$

In this section, we assume that $f(x)$ has a simple root at $x$ and $x_0$ is an initial guess sufficiently close to $x$. For solving the equation $f(x) = 0$, we suggest the three-step method

$$\begin{cases}
y_1 = x_1 - f'(x_1)^{-1}f(x_1), \\
z_1 = x_1 - 2^{-1}f(x_1)f(x_1)^{-1} + f'(y_1)^{-1}, \\
x_{1+1} = z_1 - f[z_1, x_1]^{-1}f(z_1), \quad l = 0, 1, 2, \ldots,
\end{cases} \quad (7)$$

where $f[z_1, x_1] = (z_1 - x_1)^{-1}(f(z_1) - f(x_1))$ is the two-point divided difference. This three-step iterative method is based on the two-step cubically iterative method of Homeier (the relation (28) in [7]) and secant method. In fact, for increasing the convergence order of Homeier's method from three to four, we have performed a secant approach at the third step of (7).

Note that we used this new iterative method in contrast to the existing iterative fourth-order methods in the literature [17], since only some particular nonlinear solvers have the ability to be extended for matrix inversion. To be more precise, the extensive recent findings of various orders are not always applicable for matrix inversion and this makes them interesting only from theoretical point of view, while the new method has also the ability to be used for matrix inversion and thus it is interesting from the theoretical and the application point of view.

The convergence analysis of this method is studied in the following theorem.
Theorem 2.1. Let \( x \in D \) be a simple zero of a sufficiently differentiable function \( f: D \subseteq \mathbb{R} \rightarrow \mathbb{R} \) for an open interval \( D \), which contains \( x_0 \) as an initial approximation of \( x \). If \( x_0 \) is sufficiently close to \( x \), then the three-step method defined by (7), has fourth-order convergence.

Proof. Let \( x \) be a simple zero of \( f \). Since \( f \) is sufficiently differentiable, by expanding \( f(x_k) \) and \( f'(x_k) \) about \( x \), we get
\[
f(x_k) = f'(x)(x - x_k) + O(e_k^5),
\] (8)
and
\[
f'(x_k) = f'(x)(1 + 2c_k x + c_k^2 x^2 + 4c_k^4 x^4 + O(e_k^5)),
\] (9)
where \( e_k = x_k - x \) and \( c_k = (1)_{k=0}^{n-1} \). Now, from (8) and (9), we have
\[
f(x_k) = e_k + c_k x + (2c_k^2 - c_k) x^2 + (4c_k^4 - 4c_k^2 + 3c_k) x^3 + O(e_k^5),
\] (10)
By substituting (10) into \( y_k \) of (7), we obtain
\[
y_k = x + c_k x^2 + (2c_k^2 - 2c_k) x^3 + (4c_k^4 - 4c_k^2 + 3c_k) x^4 + O(e_k^5).
\] (11)
By expanding \( f(y_k) \) about \( x \) and using (11), we have
\[
f(y_k) = f'(x)(2c_k x + 4c_k^2 x^2 + 4c_k^4 x^3 + c_k f'(x) x^4 + 11c_k^3 x^5 - 11c_k^2 x^4 - 6c_k x^3 + O(e_k^5)).
\] (12)
From (8), (10), and (12), we get
\[
\frac{f(x_k)}{f'(x_k)} + \frac{f(x_k)}{f'(y_k)} = 2e_k - c_k x^2 + (2c_k^2 - 3c_k) x^3 + (4c_k^4 - 2c_k^2 - 3c_k) x^4 + O(e_k^5).
\] (13)
Now, the use of (7) and (13) implies that
\[
z_k = x + c_k x^2 + (2c_k^2 - 3c_k) x^3 + (4c_k^4 - 2c_k^2 - 3c_k) x^4 + O(e_k^5).
\] (14)
Again, by expanding \( f(z_k) \) about \( x \) and using (14), we have
\[
f(z_k) = f'(x)
\left[
\frac{1}{2} c_k x^3 + \left( c_k^2 - \frac{3c_k x}{2} + c_k^4 \right) x^4
\right] + O(e_k^5).
\] (15)
From (8) and (15), we get
\[
f(z_k) = f'(x)
\left[
\frac{1}{2} c_k x^3 + \left( c_k^2 - \frac{3c_k x}{2} + c_k^4 \right) x^4
\right] + O(e_k^5).
\] (16)
Finally, using (7), (14) and (15), we obtain
\[
e_{k+1} = \frac{1}{2} c_k x^4 + O(e_k^5).
\] (17)
Therefore, the iteration (7) has fourth-order convergence. \( \Box \)

3. A method for matrix inversion

In this section, we propose a high-order computational method for matrix inversion. If we apply (7) to the matrix equation \( F(V) = V^{-1} - A = 0 \), then we obtain the following efficient iterative process
\[
V_{i+1} = \frac{1}{2} V_i [9I - AV_i(16I - AV_i(14I - AV_i(6I - AV_i)))] = 0, 1, 2, \ldots,
\] (18)
where \( V_0 \) is an initial approximation for \( A^{-1} \) and \( I \) is the identity matrix with the same dimension as the matrix \( A \).

We show that the sequence of iterates \( \{ V_i \}_{i=0}^{\infty} \) converges to \( A^{-1} \) with fourth-order provided that \( \| I - AV_0 \| < 1 \), where \( \| \cdot \| \) is any subordinate matrix norm.

Theorem 3.1. Let \( A = [a_{ij}] \) be a nonsingular real (complex) matrix. If the initial approximation \( V_0 \) satisfies
\[
\| E_0 \| = \| I - AV_0 \| < 1,
\] (19)
then, the iterative method (18) converges to \( A^{-1} \) and the order of this method is at least four.
Proof. Let $E_i = I - AV_i$, then

\[ E_{i+1} = I - AV_{i+1} = I - A \left( \frac{1}{2} V_i(9I - AV_i(16I - AV_i(14I - AV_i(6I - AV_i)))) \right) \]

\[ = I - \frac{1}{2} AV_i(9I - 16AV_i + 14(AV_i)^2 - 6(AV_i)^3 + (AV_i)^4) = \frac{1}{2} \left[ (I - AV_i)^4(2I - AV_i) \right] \]

\[ = \frac{1}{2} \left[ (I - AV_i)^4(I + (I - AV_i)) \right] = \frac{1}{2} [E_i^4 + E_i^5]. \quad (20) \]

So, for any subordinate matrix norm, we have

\[ \|E_{i+1}\| \leq \frac{1}{2} [\|E_i^4\| + \|E_i^5\|] \leq \left( \frac{1}{2} \right) (\|E_i\| + \|E_i\|^5). \quad (21) \]

This equation and the assumption $\|E_0\| < 1$ imply that

\[ \|E_i\| \leq \left( \frac{1}{2} \right) (\|E_0\|^4 + \|E_0\|^5) \leq \|E_0\|^4 < 1. \quad (22) \]

Now, if we use the principle of mathematical induction and suppose that

\[ \|E_l\| \leq \|E_{l-1}\|^4 < 1, \quad (23) \]

then, from (21), we have

\[ \|E_{l+1}\| \leq \left( \frac{1}{2} \right) (\|E_{l-1}\|^4 + \|E_{l-1}\|^5) \leq \|E_{l-1}\|^4. \quad (24) \]

Since $\|E_0\| < 1$, then from (24), we have that

\[ \|E_{l+1}\| \leq \|E_l\|^4 \leq \cdots \leq \|E_0\|^{4^l-1} \to 0, \quad \text{as} \; l \to \infty. \quad (25) \]

Namely, $I - AV_i \to 0$, as $l \to \infty$, and $V_i \to A^{-1}$, as $l \to \infty$. \quad (26)

Now, we prove that the order of convergence for the sequence $\{V_i\}_{i=0}^\infty$ is at least four. Let $e_l$ denote the error matrix $e_l = A^{-1} - V_i$, then $Ae_l = I - AV_i = E_i$. This together with (20) implies that

\[ Ae_{i+1} = \frac{1}{2} \left[ (Ae_i)^4 + (Ae_i)^5 \right]. \quad (27) \]

Therefore, it follows immediately that

\[ e_{i+1} = \frac{1}{2} [e_i(Ae_i)^3 + e_i(Ae_i)^4]. \quad (28) \]

By taking any subordinate norm, we obtain

\[ \|e_{i+1}\| \leq \left( \frac{1}{2} \right) \left[ (|A|)^3 + (|A|)^4 \|e_i\| \right] \|e_i\|^4. \quad (29) \]

Consequently, it has been proved that the iterative formula (18) converges to $A^{-1}$ and the order of this method is at least four. \quad \Box

Next, we give a property about the scheme (18). This property shows that $V_i(1 > 0)$ of (18), under a certain condition, may be applied to not only the left preconditioned linear system $V_iAx = V_ib$ but also to the right preconditioned linear system $AV_iy = b$, where $x = V_iy$.

**Theorem 3.2.** Let $A = [a_{ij}]$ be a nonsingular real (complex) matrix. If $AV_0 = V_0A$ is valid, then for the sequence of $\{V_i\}_{i=0}^\infty$ of (18), we have that

\[ AV_i = V_iA, \quad (30) \]

holds, for all $i = 1, 2, \ldots$.

**Proof.** First, since $AV_0 = V_0A$, from (18), we have

\[ AV_1 = \frac{1}{2} AV_0(9I - AV_0(16I - AV_0(14I - AV_0(6I - AV_0)))) = \left( \frac{1}{2} V_0A(9I - V_0A(16I - V_0A(14I - V_0A(6I - V_0A)))) \right) \]

\[ = \left( \frac{1}{2} V_0(9I - AV_0(16I - AV_0(14I - AV_0(6I - AV_0)))) \right) A = V_1A. \]
That is, when \( l = 1 \), the Eq. (30) holds. Now, we use the mathematical induction to prove the Eq. (30). Suppose that \( AV_l = V_lA \) is true, then a straightforward calculation using (18) shows that, for all \( l \geq 1 \), we have
\[
AV_{l+1} = A \left( \frac{1}{2} V_l \left( 9I - AV_l(16I - AV_l(14I - AV_l(6I - AV_l)))) \right) \right) = \left( \frac{1}{2} V_l A \left( 9I - V_l A(16I - V_l A(14I - V_l A (6I - V_l A)))) \right) \right) A = V_{l+1} A.
\]

The proof is now complete. \( \square \)

3.1. The choice for the initial value \( V_0 \)

As it is well known, the choices for the initial value \( V_0 \) on the iterative processes (2), (6) and (18) are very important to preserve the convergence. There exist many different choices for the initial value \( V_0 \) in the literature. Some of the suitable choices of \( V_0 \) are as follows:

(I) Pan and Schreiber [12] assumed that the approximations \( V_l \) share singular vectors with \( A^T \), and both the largest (\( \sigma_{\text{max}} \)) and the smallest singular values (\( \sigma_{\text{min}} \)) of \( A \) are available, then for a general matrix \( A \), one can choose
\[
V_0 = \frac{2A^T}{\sigma_{\text{min}}^2 + \sigma_{\text{max}}^2}.
\]

It is not difficult to show that for this choice, we have \( \| I - AV_0 \|_2 < 1 \).

(II) Ben-Israel and Greville [2] proposed an initial approximate inverse of any nonsingular matrix by the formula
\[
V_0 = \alpha A^*.
\]

where \( 0 < \alpha < \frac{1}{\| A \|_\infty^2} \) and \( A^* \) is the conjugate transpose matrix of \( A \). They also proved that \( \| I - V_0 A \| < 1 \).

(III) Grosz [6] suggested the diagonal matrix
\[
V_0 = \text{diag}(1/a_{11}, 1/a_{22}, \ldots, 1/a_{nn}),
\]

as the initial approximation, where \( a_{ii} \) is the \( i \)th diagonal entry of \( A \). For this choice, when \( A \) is strictly diagonal dominant, we have \( \| I - V_0 A \|_\infty < 1 \).

(IV) Pan and Schreiber [12] proposed the choice
\[
V_0 = \frac{A^T}{\| A \|_1 \| A \|_\infty},
\]

for an \( n \times n \) nonsingular matrix \( A \), where \( T \) stands for transpose, \( \| A \|_1 = \max_{i} (\sum_{j=1}^{n} \left| a_{ij} \right|) \) and \( \| A \|_\infty = \max_{i} (\sum_{j=1}^{n} \left| a_{ij} \right|) \). This choice yields \( \| I - AV_0 \|_2 \leq 1 - 1/(m\kappa^2) \), where \( \kappa = \sigma_{\text{max}}/\sigma_{\text{min}} \). Moreover, Codevico et al. in [5] for an \( n \times n \) symmetric positive definite matrix \( A \), by choosing
\[
V_0 = I/(\| A \|_F),
\]

they obtained the bound \( \| I - AV_0 \|_2 \leq 1 - 1/(\sqrt{n}\kappa) \).

(V) We can also choose \( V_0 = \alpha I \), where \( I \) is the identity matrix, and \( \alpha \in \mathbb{R} \) is a fixed number such that \( \max_{i} |1 - \alpha \lambda_{ii}| < 1 \), where \( \lambda_{ii}, i = 1, \ldots, n \) are eigenvalues of \( A \).

4. Moore–Penrose inverse

The Moore–Penrose inverse of a matrix \( A \in \mathbb{C}^{m \times k} \), denoted by \( A^\dagger \in \mathbb{C}^{k \times m} \), is a matrix \( X \) satisfying the following four Penrose equations
\[
AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA,
\]

where \( A^* \) is the conjugate transpose of \( A \). It is well known that for any matrix \( A \in \mathbb{C}^{m \times k} \), its Moore–Penrose inverse exists. For computing the Moore–Penrose inverse of a matrix, various numerical solution methods were developed, e.g., [2], [3], and [16]. In this section, we will show that the scheme (18), with the starting value (32), can be used to compute the Moore–Penrose inverse of a matrix and it converges with fourth-order convergence.

Lemma 4.1. For the sequence \( \{ V_l \}_{l=0}^{\infty} \) generated by the iterative Schulz-type method (18) and \( V_0 = \alpha A^* \), for any \( l \geq 0 \), it holds that
\[
(AV_l)^* = AV_l, \quad (V_l A)^* = V_l A, \quad V_l A A^\dagger = V_l, \quad A^\dagger A V_l = V_l.
\]
Proof. By using the fact that \( V_{t+1} = \frac{1}{2} V_t \left( 9I - 16AV_t + 14(AV_t)^2 - 6(AV_t)^3 + (AV_t)^4 \right) \), the proof is similar to the proof of Lemma 2.1 in [3]. \( \square \)

Lemma 4.2. For \( A \in \mathbb{C}^{m \times n} \) with the singular values \( \sigma_1 > \sigma_2 > \cdots > \sigma_r > 0 \), and the initial approximation \( V_0 = zA^\dagger \) with \( 0 < z < 2/\sigma_r^2 \), it holds that

\[
\|A(V_0 - A^\dagger)\| < 1.
\] (38)

Proof. The proof is similar to that given in Theorem 2.1 in [3].

Theorem 4.3. For \( A \in \mathbb{C}^{m \times n} \), with the singular values \( \sigma_1 > \sigma_2 > \cdots > \sigma_r > 0 \), the sequence \( \{V_t\}_{t=0}^\infty \) generated by (18) and using the initial approximation \( V_0 = zA^\dagger \), converges to the Moore–Penrose inverse \( A^\dagger \) in fourth-order provided that \( 0 < z < 2/\sigma_r^2 \).

Proof. Set \( E_t = V_t - A^\dagger \), and \( E_t = I - AV_t \), then, using (20), we have

\[
A_{t+1} = AV_t - AA^\dagger = AV_t - I + I - AA^\dagger = -E_t - I - AA^\dagger = -\frac{1}{2} [E_t^4 + E_t^3] + I - AA^\dagger
\] (39)

On the other hand, from the definitions of the Moore–Penrose inverse \( A^\dagger \), we have

\[
(I - AA^\dagger)^k = I - AA^\dagger, \quad k = 1, 2, \ldots; \quad (I - AA^\dagger)A_{t+1} = 0.
\]

The use of these relations implies that

\[
-E_t^4 + (I - AA^\dagger) = -(I - AV_t)^4 + (I - AA^\dagger) = -(I - AA^\dagger + AA^\dagger - AV_t)^4 + (I - AA^\dagger) = (I - AA^\dagger - AE_t)^4 + (I - AA^\dagger)
\]

\[
= -\left( (I - AA^\dagger) - 4(I - AA^\dagger)AE_t + 6(I - AA^\dagger)(AE_t)^2 - 4(I - AA^\dagger)(AE_t)^3 + (AE_t)^4 \right) + (I - AA^\dagger)
\]

\[
= -\left( (I - AA^\dagger) - 4(I - AA^\dagger)AE_t + 6(I - AA^\dagger)(AE_t)^2 - 4(I - AA^\dagger)(AE_t)^3 + (AE_t)^4 \right) + (I - AA^\dagger)
\]

\[
= -\left( (I - AA^\dagger) - 4(I - AA^\dagger)AE_t + 6(I - AA^\dagger)(AE_t)^2 - 4(I - AA^\dagger)(AE_t)^3 + (AE_t)^4 \right) + (I - AA^\dagger)
\]

Thus, we have

\[ A_{t+1} = \frac{1}{2} \left[ -(AE_t)^4 + (AE_t)^5 \right]. \] (40)

Similarly, we can show that

\[ -E_t^5 + (I - AA^\dagger) = (AE_t)^5. \] (41)

Thus, we have

\[ A_{t+1} = \frac{1}{2} \left[ -(AE_t)^4 + (AE_t)^5 \right]. \] (42)

So, for any matrix norm \( \| \cdot \| \), we obtain

\[ \|A_{t+1}\| \leq \frac{1}{2} [\|AE_t\|^4 + \|AE_t\|^5]. \] (43)

By using Lemma 4.2 which implies that \( \|A_{t=0}\| < 1 \), and a similar reasoning as in (22)–(25), one can obtain

\[ \|A_{t+1}\| \leq \frac{1}{2} \left[ \|AE_t\|^4 + \|AE_t\|^5 \right] \leq \|AE_t\|^4 \leq \|A\|^4 \|E_t\|^4. \] (44)

Finally, using the properties of the Moore–Penrose inverse \( A^\dagger \) and Lemma 4.1, it would be now easy to find the error inequality of the new scheme (18) as follows:

\[ \|V_{t+1} - A^\dagger\| = \|A^\dagger AV_{t+1} - A^\dagger AA^\dagger\| \leq \|A^\dagger\|\|AV_{t+1} - AA^\dagger\| = \|A^\dagger\|\|AE_{t+1}\| \leq \|A^\dagger\|^4 \|E_t\|^4. \] (45)

Thus, \( \|V_t - A^\dagger\| \to 0 \), i.e., the sequence (18) converges to the Moore–Penrose inverse in fourth-order as \( t \to +\infty \). This ends the proof. \( \square \)

Remark 1. As Chen in [4] and Alpert et al. in [1] mentioned, the iterative Schulz-type methods are fruitful for large sparse matrices having sparse (generalized) inverses, by employing a threshold to keep the sparsity of the output matrices; or when, an approximate inverse of rather low accuracy in the sparse form is needed.
5. Numerical experiments

In this section, we will make some numerical experiments on our proposed method (18) and give some numerical comparison with methods (2), (3) and the fourth-order method extracted from (4). All tests were carried out in double precision with a Matlab code, while the computer specifications are Microsoft Windows XP Intel(R), Pentium(R) 4, CPU 3.20 GHz, with 4 GB of RAM.

Example 1 ([11]). Let \( A = \text{rand}(200, 200) \), \( x = \frac{1}{|A|} \), and \( V_0 = xA^* \). The stop criterion is \( \| I - V_1 A \| < 10^{-8} \) and the maximum number of iterations allowed set to 200. We used 50 matrices of the size \( 200 \times 200 \) with random entries. The CPU times and the number of iterations required for convergence are compared in Figs. 1 and 2, respectively. In these figures, x-axis represents the random matrices and y-axis represents the computing time in Fig. 1 and the number of iterations required for convergence in Fig. 2. From the Figs. 1 and 2, we see that, the method (18) is more efficient than the methods (2) and (3), while it has a similar convergence behavior to (4).

Example 2 ([9,10]). Let us consider the linear system \( Ax = b \), where the matrix \( A \) arises from the five-point discretization of the following second order elliptic partial differential equation:

\[
-\frac{\partial}{\partial x} \left( a \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( b \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial x} (cu) + \frac{\partial}{\partial y} (du) + fu = 0,
\]

with \( a(x, y) > 0, b(x, y) > 0 \), \( c(x, y), d(x, y) \), and \( f(x, y) \) defined on a unit square region \( \Omega = (0, 1) \times (0, 1) \), and Dirichlet boundary condition \( u(x, y) = 0 \) on \( \partial \Omega \) are used, where \( \partial \Omega \) denotes the boundary of \( \Omega \).

Now, we consider the Eq. (46) with \( a(x, y) = b(x, y) = 1 \), \( c(x, y) = \cos(x/6), d(x, y) = \sin(y/6) \) and \( f(x, y) = 1 \). And we use four uniform meshes of \( h_x = h_y = 1/11, h_x = h_y = 1/21, h_x = h_y = 1/31 \) and \( h_x = h_y = 1/41 \), which lead to four matrices of orders \( n = 100 \times 100, n = 400 \times 400, n = 900 \times 900 \) and \( n = 1600 \times 1600 \), where \( h_x \) and \( h_y \) refer to the mesh sizes in the x-direction and y-direction, respectively.

We used the right preconditioned GMRES method for solving the linear system \( Ax = b \). In all of our runs, we used the zero vector as an initial approximate solution, and right-hand side vector \( b \) is chosen such that \( b = A(1, 1, \ldots, 1)^T \). The stopping criterion

\[
\| r_k \|_2 / \| r_0 \|_2 \leq 10^{-8},
\]

was used, where \( r_k = b - Ax_k \) is the \( k \)th iterated residual of the linear system to be solved. In this example, we focus our attention on comparison between the preconditioners \( V_1, V_2 \) using Eq. (18) and \( V_1, V_2 \) using Eq. (3) and Eq. (4), when \( V_0 = \text{diag}(1/a_{11}, 1/a_{22}, \ldots, 1/a_{nn}) \). Table 1 presents the computation time in seconds (Time) and the number of iterations of preconditioned GMRES algorithm (IT) required for satisfying (47). “\( V_1-(3)-\)Order-3, \( V_2-(3)-\)order-3”, “\( V_1-(4)-\)Order-4,
V2-(4)-order-4", and "V1-(18)-Order-4, V2-(18)-order-4", denote the preconditioned GMRES algorithm with the preconditioners V1, V2 using Eqs. (3), (4), and (18), respectively. The results of Table 1 show that, the convergence behavior of V1-(18)-Order-4 and V2-(18)-Order-4 algorithms (the preconditioned GMRES algorithm with the preconditioner V1 using new scheme (18) and the fourth-order method extracted from (4)), in terms of the elapsed CPU time are better than that of the GMRES with other preconditioners. These results show that the new scheme furnishes an effective algorithm for computing an approximate inverse of a matrix as a preconditioner for the Krylov subspace methods.

In fact, the preconditioner obtained by the iterative Schulz-type methods will be denser per iteration. Hence, the first approximate inverse for preconditioning is the best choice in order to reduce the condition number. This is confirmed in Table 1 and shows that the best preconditioner can be obtained by only one iterate of the new method (18). The best answers in terms of (total) CPU times are bolded in Table 1.

One might say that there is a little difference between (4) and (18), so why do we need the new method (18)? To answer this, we clearly mention that the iterative scheme (4) needs to compute matrix powers per cycle, while the new method has been written in its Horner form. To be more precise, we remind that the computation matrix powers for larger matrices produce some round-off errors which might make the iterative scheme to be asymptotically unstable while this does not happen when the iteration has been used in the Horner form (just like (18)).

Example 3. In this test, we compare the iterative methods (2)–(4) and (18) for finding the generalized inverse as discussed in Section 4. We assume that V0 = z'A' with z = 1/|A|2, is the initial matrix, and the stop criterion is

\[
\max\{|AV_iA - A|_F, \|V_iA - V_i\|_F, \|V_i\|_F, \|(V_iA)^* - AV_i\|_F, \|(V_iA)^* - V_iA\|_F\} < 10^{-8},
\]
where \( \| \cdot \|_F \) denotes the Frobenius-norm of a matrix, while the maximum number of iterations allowed set to 100. Toward this aim, for different sizes of matrices, whose elements are randomly taken from \([0, 1]\), we have performed five tests and compared the average values of the iterations and the elapsed times in seconds. The results are listed in Table 2. The fourth-order method (18) and (4) are better than the low order methods (2) and (3).

In addition, for every method and the test random matrices with the size 100 \( \times 110 \), by using (see [8,18])

\[
\text{AOC} = \frac{\ln(\|V_{i+1} - V_i\|_F)}{\ln(\|V_{i+1} - V_i\|_F/\|V_{i+1} - V_i\|_F^2)},
\]

(48)

and 40 digits floating point, we have computed the approximated order of convergence \( p \). The estimated order of convergence that appears in Table 3 is the maximum of the last four finite values of quotient (48), when the iterations converge to the desired solution.

From Table 2, we observe that the iterative method (18) and (4) have a higher efficiency than the other methods in terms of the number of iterations and the elapsed time required for convergence. Furthermore, from Table 3, we can observe that the approximated order of convergence of the iterative methods (2)–(4) are roughly 2, 3, 4, respectively, and the approximated order of convergence of the iterative method (18) is greater than 4, as we indicated in the proof of Theorem 4.3.

6. Conclusion

In this paper, we have proposed a new iterative scheme for computing the roots of an algebraic equation \( f(x) = 0 \). We have proved that this method has fourth local convergence. Based on this scheme, we have also presented a new fourth-order computational algorithm to compute an approximate inverse of a square matrix, which contained the main contribution of this paper.

The theoretical proofs and numerical experiments have showed that this iterative method is of fourth-order and effective. Furthermore, this approximate inverse can be used as a preconditioner for solving linear systems by using the preconditioned Krylov subspace methods. We observed that this preconditioner is reliable and effective for reducing the number of iterations and the computational time required for convergence.

We have also extensively discussed the extension of the new method for computing the Moore–Penrose inverse. Finally, we have also provided further findings, which are useful in high precision computing environment in order to check and observe the computational order of convergence for different Schulz-type iterative inverse-finders.

The extension of the new iterative scheme for generalized outer inverse \( A_{r,2}^{(2)} \) could be considered as a future work in this trend of research.
Acknowledgements

The authors would like to thank the anonymous referee for the valuable comments and suggestions which substantially improved the quality of this article. They would like to extend their gratitude to Ph.D. Melvin Scott for some interesting comments on the revision of this manuscript.

References