

## Research Article

# A New High-Order Stable Numerical Method for Matrix Inversion

F. Khaksar Haghani<sup>1</sup> and F. Soleymani<sup>2</sup>

<sup>1</sup> Department of Mathematics, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran

<sup>2</sup> Department of Mathematics, Zahedan Branch, Islamic Azad University, Zahedan, Iran

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

Received 22 August 2013; Accepted 23 October 2013; Published 6 February 2014

Academic Editors: N. Herisanu and N. I. Mahmudov

Copyright © 2014 F. K. Haghani and F. Soleymani. 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 stable numerical method is proposed for matrix inversion. The new method is accompanied by theoretical proof to illustrate twelfth-order convergence. A discussion of how to achieve the convergence using an appropriate initial value is presented. The application of the new scheme for finding Moore-Penrose inverse will also be pointed out analytically. The efficiency of the contributed iterative method is clarified on solving some numerical examples.

## 1. Introduction and Preliminary Notes

It is well known that the inverse of a square matrix  $A_{m \times m}$ , which is also known as a reciprocal matrix, is a matrix  $A^{-1}$  such that  $AA^{-1} = I$ , where  $I$  is the identity matrix. A regular nonsingular matrix  $A_{m \times m}$  can be inverted using methods such as the Gauss-Jordan elimination or Gaussian elimination method. Such schemes fall in the category of direct methods for this purpose.

The direct methods cannot properly handle sparse matrices possessing sparse inverses arising in the numerical solution of integral equations [1]. On the other hand, methods such as conjugate gradient for symmetric positive definite matrices and GMRES are effective for large sparse linear systems. However, there is a problem when the coefficient matrix (when solving a linear system of equations) is ill-conditioned. To remedy this, one may apply a preconditioner to the system in which its construction is not an easy task [2].

An iterative method for preconditioning is the SPAI (sparse approximate inverse preconditioner) algorithm [3]. Given a sparse matrix  $A$  the SPAI algorithm computes a sparse approximate inverse  $M$  by minimizing  $\|AM - I\|$  in the Frobenius norm. Then, the approximate inverse is computed explicitly and can be applied as a preconditioner to an iterative method.

There are other types of schemes, which can be considered as iteration methods while they have different structures; see, for example, [4, 5]. In such iterative methods, at each iteration an approximate inverse of a matrix (if it is rectangular, one can find Moore-Penrose inverse) may easily be attained. And consequently, the users have the ability to solve the linear systems (with multiple right-hand side vectors) iteratively or use the approximate inverses in sensitivity analysis and the preconditioning of a linear system. This type of methods is in focus here.

Several known methods were proposed for approximating matrix inverse, such as those based on the so-called minimum residual iterations and Hotelling-Bodewig algorithm [6]. The Hotelling-Bodewig algorithm is defined by

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

where  $I$  is the identity matrix. Schulz in [7] found that the eigenvalues of  $I - AV_0$  must have magnitudes less than 1 to ensure the convergence, which is a key element in designing higher efficient Schulz-type iterative methods.

In 2011, Li et al. in [8] theoretically investigated

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



*Proof.* Let  $\|I - AV_0\| < 1$ , and for the sake of simplicity assume that  $E_0 = I - AV_0$  and  $E_n = I - AV_n = I - \psi_n$  stand for the symmetric residual matrix. It is straightforward to have

$$\begin{aligned}
 E_{n+1} &= I - AV_{n+1} \\
 &= I - A \left( \frac{1}{64} V_n \left( 816I - 4812\psi_n + 17393\psi_n^2 - 43044\psi_n^3 \right. \right. \\
 &\quad \left. \left. + 77154\psi_n^4 - 103312\psi_n^5 + 105039\psi_n^6 \right. \right. \\
 &\quad \left. \left. - 81576\psi_n^7 + 48268\psi_n^8 - 21516\psi_n^9 \right. \right. \\
 &\quad \left. \left. + 7071\psi_n^{10} - 1652\psi_n^{11} \right. \right. \\
 &\quad \left. \left. + 258\psi_n^{12} - 24\psi_n^{13} + \psi_n^{14} \right) \right) \\
 &= -\frac{1}{64} (-4I + \psi_n)^3 (-I + \psi_n)^{12} \\
 &= \frac{1}{64} (3I + I - \psi_n)^3 (I - \psi_n)^{12} \\
 &= \frac{1}{64} (3I + E_n)^3 E_n^{12} \\
 &= \frac{1}{64} (27E_n^{12} + 27E_n^{13} + 9E_n^{14} + E_n^{15}).
 \end{aligned} \tag{9}$$

Hence, we attain

$$\|E_{n+1}\| \leq \frac{1}{64} (27\|E_n\|^{12} + 27\|E_n\|^{13} + 9\|E_n\|^{14} + \|E_n\|^{15}). \tag{10}$$

In addition, since  $\|E_0\| < 1$ , by relation (10) and using mathematical induction, we obtain that  $\|E_1\| \leq (1/64)(27\|E_0\|^{12} + 27\|E_0\|^{13} + 9\|E_0\|^{14} + \|E_0\|^{15}) \leq \|E_0\|^{12} < 1$ . If we consider  $\|E_n\| < 1$ , therefore

$$\begin{aligned}
 \|E_{n+1}\| &\leq \frac{1}{64} (27\|E_n\|^{12} + 27\|E_n\|^{13} + 9\|E_n\|^{14} + \|E_n\|^{15}) \\
 &\leq \|E_n\|^{12}.
 \end{aligned} \tag{11}$$

Furthermore, we get that

$$\|E_{n+1}\| \leq \|E_n\|^{12} \leq \dots \leq \|E_0\|^{12^{n+1}} < 1. \tag{12}$$

That is,  $I - AV_n \rightarrow 0$ , when  $n \rightarrow \infty$ , and thus  $V_n \rightarrow A^{-1}$ , as  $n \rightarrow \infty$ .

Now, we must show the twelfth order using the sequence  $\{V_n\}_{n=0}^{n=\infty}$ . To do this, we denote  $\epsilon_n = V_n - A^{-1}$  as the error matrix in the iterative procedure (7). We have

$$\begin{aligned}
 I - AV_{n+1} &= \frac{1}{64} [27(I - AV_n)^{12} + 27(I - AV_n)^{13} \\
 &\quad + 9(I - AV_n)^{14} + (I - AV_n)^{15}].
 \end{aligned} \tag{13}$$

Hence, we could get that

$$\begin{aligned}
 A(A^{-1} - V_{n+1}) &= \frac{1}{64} [27A^{12}(A^{-1} - V_n)^{12} + 27A^{13}(A^{-1} - V_n)^{13} \\
 &\quad + 9A^{14}(A^{-1} - V_n)^{14} + A^{15}(A^{-1} - V_n)^{15}].
 \end{aligned} \tag{14}$$

By multiplying  $A^{-1}$  by the left side, we have

$$\begin{aligned}
 A^{-1} - V_{n+1} &= \frac{1}{64} [27A^{11}(A^{-1} - V_n)^{12} + 27A^{12}(A^{-1} - V_n)^{13} \\
 &\quad + 9A^{13}(A^{-1} - V_n)^{14} + A^{14}(A^{-1} - V_n)^{15}],
 \end{aligned} \tag{15}$$

which now by taking an arbitrary matrix norm results in

$$\begin{aligned}
 \|\epsilon_{n+1}\| &\leq \frac{1}{64} [27\|A\|^{11}\|\epsilon_n\|^{12} + 27\|A\|^{12}\|\epsilon_n\|^{13} \\
 &\quad + 9\|A\|^{13}\|\epsilon_n\|^{14} + \|A\|^{14}\|\epsilon_n\|^{15}].
 \end{aligned} \tag{16}$$

And thus

$$\begin{aligned}
 \|\epsilon_{n+1}\| &= \left( \frac{1}{64} [27\|A\|^{11} + 27\|A\|^{12}\|\epsilon_n\|^1 \right. \\
 &\quad \left. + 9\|A\|^{13}\|\epsilon_n\|^2 + \|A\|^{14}\|\epsilon_n\|^3] \right) \|\epsilon_n\|^{12}.
 \end{aligned} \tag{17}$$

That is, the iteration (7) converges with at least twelfth order to  $A^{-1}$ . This concludes the proof.  $\square$

At this time, we discuss an application of (7) for finding the generalized inverses. The Moore-Penrose inverse of a complex matrix  $A \in \mathbb{C}^{m \times k}$  (also called pseudoinverse), denoted by  $A^\dagger \in \mathbb{C}^{k \times m}$ , is a unique matrix  $V \in \mathbb{C}^{k \times m}$  satisfying the following four Penrose equations:

$$\begin{aligned}
 AVA &= A, & VAV &= V, \\
 (AV)^* &= AV, & (VA)^* &= VA,
 \end{aligned} \tag{18}$$

wherein  $A^*$  is the conjugate transpose of  $A$ . Ben-Israel and his colleagues in [14, 15] used the method (1) with the starting value

$$V_0 := \alpha A^*, \tag{19}$$

where  $0 < \alpha < 2/\rho(A^*A)$  and  $\rho(\cdot)$  denotes the spectral radius. The authors in [15] further investigated that the sequence generated by

$$V_n = \alpha \sum_{i=0}^n A^* (I - \alpha A A^*)^i, \quad n = 0, 1, 2, \dots, \tag{20}$$

converges to the pseudoinverse.

In the following theorem, we show analytically that in case of having singular or rectangular matrices, scheme (7) by considering the initial approximation (19) converges to the Moore-Penrose generalized inverse.

**Theorem 2.** For the sequence  $\{V_n\}_{n=0}^{n=\infty}$  generated by the iterative Schulz-type method (7), and any  $n \geq 0$ , it holds that

$$\begin{aligned} (AV_n)^* &= AV_n, & (V_nA)^* &= V_nA, \\ V_nAA^\dagger &= V_n, & A^\dagger AV_n &= V_n. \end{aligned} \tag{21}$$

*Proof.* We will prove the conclusion by induction on  $n$ . For  $n = 0$ , and considering (19), the first two equations of (21) can be demonstrated simply. And thus, we only give a verification to the last two equations as follows:

$$\begin{aligned} V_0AA^\dagger &= \alpha A^* AA^\dagger = \alpha A^* (AA^\dagger)^* = \alpha A^* (A^\dagger)^* A^* \\ &= \alpha (AA^\dagger A)^* = \alpha A^* = V_0, \\ A^\dagger AV_0 &= (A^\dagger A) \alpha A^* = \alpha (A^\dagger A)^* A^* = \alpha A^* (A^\dagger)^* A^* \\ &= \alpha (AA^\dagger A)^* = \alpha A^* = V_0. \end{aligned} \tag{22}$$

Assume now that the conclusion holds for some  $n > 0$ . We now show that it continues to hold for  $n+1$ . Using the iterative method (7), one has

$$\begin{aligned} (AV_{n+1})^* &= \left[ A \left( \frac{1}{64} V_n \phi(V_n) \right) \right]^* \\ &= \frac{1}{64} \left( 816\psi_n^* - 4812(\psi_n^*)^2 + 17393(\psi_n^*)^3 \right. \\ &\quad - 43044(\psi_n^*)^4 + 77154(\psi_n^*)^5 - 103312(\psi_n^*)^6 \\ &\quad + 105039(\psi_n^*)^7 - 81576(\psi_n^*)^8 + 48268(\psi_n^*)^9 \\ &\quad - 21516(\psi_n^*)^{10} + 7071(\psi_n^*)^{11} - 1652(\psi_n^*)^{12} \\ &\quad \left. + 258(\psi_n^*)^{13} - 24(\psi_n^*)^{14} + (\psi_n^*)^{15} \right) \\ &= \frac{1}{64} \left[ 816\psi_n - 4812\psi_n^2 + 17393\psi_n^3 - 43044\psi_n^4 \right. \\ &\quad + 77154\psi_n^5 - 103312\psi_n^6 + 105039\psi_n^7 \\ &\quad - 81576\psi_n^8 + 48268\psi_n^9 - 21516\psi_n^{10} \\ &\quad \left. + 7071\psi_n^{11} - 1652\psi_n^{12} + 258\psi_n^{13} - 24\psi_n^{14} + \psi_n^{15} \right] \\ &= A \left( \frac{1}{64} V_n \phi(V_n) \right) \\ &= AV_{n+1}, \end{aligned} \tag{23}$$

where the following fact  $(AV_n)^* = AV_n$  has been used. Thus, the first equality in (21) holds for  $n+1$ , and the second equality can be proved in a similar way. For the third equality in (21),

using the assumption that  $V_nAA^\dagger = V_n$  and the iterative method (7), we could write down

$$\begin{aligned} V_{n+1}AA^\dagger &= \left[ \frac{1}{64} V_n \phi(V_n) \right] AA^\dagger \\ &= \frac{1}{64} \left( 816V_nAA^\dagger - 4812V_n\psi_nAA^\dagger + 17393V_n\psi_n^2AA^\dagger \right. \\ &\quad - 43044V_n\psi_n^3AA^\dagger + 77154V_n\psi_n^4AA^\dagger \\ &\quad - 103312V_n\psi_n^5AA^\dagger + 105039V_n\psi_n^6AA^\dagger \\ &\quad - 81576V_n\psi_n^7AA^\dagger + 48268V_n\psi_n^8AA^\dagger \\ &\quad - 21516V_n\psi_n^9AA^\dagger + 7071V_n\psi_n^{10}AA^\dagger \\ &\quad - 1652V_n\psi_n^{11}AA^\dagger + 258V_n\psi_n^{12}AA^\dagger \\ &\quad \left. - 24V_n\psi_n^{13}AA^\dagger + V_n\psi_n^{14}AA^\dagger \right) \\ &= \frac{1}{64} \left( 816V_n - 4812V_n\psi_n + 17393V_n\psi_n^2 \right. \\ &\quad - 43044V_n\psi_n^3 + 77154V_n\psi_n^4 - 103312V_n\psi_n^5 \\ &\quad + 105039V_n\psi_n^6 - 81576V_n\psi_n^7 + 48268V_n\psi_n^8 \\ &\quad - 21516V_n\psi_n^9 + 7071V_n\psi_n^{10} - 1652V_n\psi_n^{11} \\ &\quad \left. + 258V_n\psi_n^{12} - 24V_n\psi_n^{13} + V_n\psi_n^{14} \right) \\ &= \frac{1}{64} V_n \phi(V_n) \\ &= V_{n+1}. \end{aligned} \tag{24}$$

Hence, the third equality in (21) holds for  $n + 1$ . The fourth equality can similarly be proved, and the desired result follows.  $\square$

The iterative method (7) is a matrix multiplication rich scheme. So, in order to reduce the computational load of matrix multiplications, it is enough to use `SparseArray[mat]` to avoid unnecessary multiplications to the nonzero elements when dealing with large sparse matrices.

*Remark 3.* The (inverse finder) informational efficiency index is defined as  $p/\eta$  where  $p$  and  $\eta$  stand for the local order and the number of matrix-matrix products per cycle. The proposed method of this paper requires 8 matrix-matrix multiplications to achieve the convergence order 12. This implies that  $12/8 = 1.5$  as its informational index, which is much better than  $2/2 = 1$ ,  $3/3 = 1$ ,  $3/4 = 0.75$ , and  $12/12 = 1$  of schemes (1), (2), (3), and (4), respectively.

Before stating the main theorem for finding Moore-Penrose inverse, it is required to recall that for  $A \in \mathbb{C}^{m \times k}$

with the singular values  $\sigma_1 > \sigma_2 > \dots > \sigma_r > 0$  and the initial approximation  $V_0 = \alpha A^*$  with  $0 < \alpha < 2/\sigma_1^2$ , it holds that

$$\|A(V_0 - A^\dagger)\| < 1. \tag{25}$$

We are about to use this fact in the following theorem so as to find the theoretical order of the proposed method (7) for finding the Moore-Penrose inverse (see [16] for more details).

**Theorem 4.** For  $A \in \mathbb{C}^{m \times k}$ , with the singular values  $\sigma_1 > \sigma_2 > \dots > \sigma_r > 0$ , the sequence  $\{V_n\}_{n=0}^{\infty}$  generated by (7) and using the initial approximation  $V_0 = \alpha A^*$  converges to the Moore-Penrose inverse  $A^\dagger$  in twelfth order provided that  $0 < \alpha < 2/\sigma_1^2$ .

*Proof.* Set  $E_n = V_n - A^\dagger$  and  $E_n = I - AV_n$ ; we have

$$\begin{aligned} AE_{n+1} &= AV_{n+1} - AA^\dagger \\ &= AV_{n+1} - I + I - AA^\dagger \\ &= -E_{n+1} + I - AA^\dagger \\ &= -\frac{1}{16} [27E_n^{12} + 27E_n^{13} + 9E_n^{14} + E_n^{15}] + I - AA^\dagger. \end{aligned} \tag{26}$$

On the other hand, from the definitions of the Moore-Penrose inverse  $A^\dagger$ , we have

$$\begin{aligned} (I - AA^\dagger)^k &= I - AA^\dagger, \quad k = 1, 2, \dots; \\ (I - AA^\dagger)AE_n &= 0. \end{aligned} \tag{27}$$

The use of these relations implies that

$$\begin{aligned} AE_{n+1} &= \frac{1}{16} [-27(AE_n)^{12} + 27(AE_n)^{13} \\ &\quad - 9(AE_n)^{14} + (AE_n)^{15}]. \end{aligned} \tag{28}$$

So, for any matrix norm  $\|\cdot\|$ , we obtain

$$\begin{aligned} \|AE_{n+1}\| &\leq \frac{1}{16} [27\|AE_n\|^{12} + 27\|AE_n\|^{13} \\ &\quad + 9\|AE_n\|^{14} + \|AE_n\|^{15}]. \end{aligned} \tag{29}$$

Applying (25), which implies that  $\|AE_0\| < 1$ , and a similar reasoning as in (10)–(12), one can obtain

$$\begin{aligned} \|AE_{n+1}\| &\leq \frac{1}{16} [27\|AE_n\|^{12} + 27\|AE_n\|^{13} + 9\|AE_n\|^{14} + \|AE_n\|^{15}] \\ &\leq \|AE_n\|^{12} \leq \|A\|^{12} \|E_n\|^{12}. \end{aligned} \tag{30}$$

Finally, using the properties of the Moore-Penrose inverse  $A^\dagger$  and Theorem 2, it would be now easy to find the error inequality of the new scheme (7) as follows:

$$\begin{aligned} \|V_{n+1} - A^\dagger\| &= \|A^\dagger AV_{n+1} - A^\dagger AA^\dagger\| \leq \|A^\dagger\| \|AV_{n+1} - AA^\dagger\| \\ &= \|A^\dagger\| \|AE_{n+1}\| \leq \|A^\dagger\| \|A\|^{12} \|E_n\|^{12}. \end{aligned} \tag{31}$$

Thus,  $\|V_n - A^\dagger\| \rightarrow 0$ ; that is, the sequence of (7) converges to the Moore-Penrose inverse in twelfth order as  $n \rightarrow +\infty$ . This ends the proof.  $\square$

### 3. Stability

We investigate the stability of (7) for finding  $A^\dagger$  (or the simplified case  $A^{-1}$ ) in a neighborhood of the solution of equation  $AV = I$ . Note that if the iteration is not self-correcting, that is, if errors made at one stage are not subsequently damped, then the inevitable rounding errors introduced into the iteration may accumulate to the point where they overwhelm the answer. Thus, we should either show that the proposed method is self-correcting or must furnish an analysis showing that rounding errors remain under control. This will be done in what follows. In fact, we analyze how a small perturbation at the  $n$ th iterate is amplified or damped along the iterates. Note that this procedure has recently been applied on a general family of methods for matrix inversion in [17].

**Theorem 5.** The sequence  $\{V_n\}_{n=0}^{\infty}$  generated by (7) with the initial approximation (19) is asymptotically stable for finding the Moore-Penrose generalized inverse.

*Proof.* Let  $\Delta V_n$  be the numerical perturbation introduced at the  $n$ th iterate of (7). Next, one has

$$\tilde{V}_n = V_n + \Delta V_n. \tag{32}$$

Here, we perform a first-order error analysis; that is, we formally neglect quadratic or higher terms such as  $(\Delta V_n)^2$ . This formal manipulation is meaningful if  $\Delta V_n$  is sufficiently small and further yields to  $V_n \cdot \Delta V_n \approx \Delta V_n \cdot V_n$ . We have

$$\begin{aligned} \tilde{V}_{n+1} &= \frac{1}{64} \tilde{V}_n \phi(\tilde{V}_n) \\ &= \frac{1}{64} \tilde{V}_n (816I - 4812A\tilde{V}_n + 17393(A\tilde{V}_n)^2 - 43044(A\tilde{V}_n)^3 \\ &\quad + 77154(A\tilde{V}_n)^4 - 103312(A\tilde{V}_n)^5 + 105039(A\tilde{V}_n)^6 \\ &\quad - 81576(A\tilde{V}_n)^7 + 48268(A\tilde{V}_n)^8 - 21516(A\tilde{V}_n)^9 \\ &\quad + 7071(A\tilde{V}_n)^{10} - 1652(A\tilde{V}_n)^{11} + 258(A\tilde{V}_n)^{12} \\ &\quad - 24(A\tilde{V}_n)^{13} + (A\tilde{V}_n)^{14}) \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{64} (V_n + \Delta V_n) \\
 &\times \left( 816I - 4812A(V_n + \Delta V_n) + 17393(A(V_n + \Delta V_n))^2 \right. \\
 &\quad - 43044(A(V_n + \Delta V_n))^3 + 77154(A(V_n + \Delta V_n))^4 \\
 &\quad - 103312(A(V_n + \Delta V_n))^5 + 105039(A(V_n + \Delta V_n))^6 \\
 &\quad - 81576(A(V_n + \Delta V_n))^7 + 48268(A(V_n + \Delta V_n))^8 \\
 &\quad - 21516(A(V_n + \Delta V_n))^9 + 7071(A(V_n + \Delta V_n))^{10} \\
 &\quad - 1652(A(V_n + \Delta V_n))^{11} + 258(A(V_n + \Delta V_n))^{12} \\
 &\quad \left. - 24(A(V_n + \Delta V_n))^{13} + (A(V_n + \Delta V_n))^{14} \right) \\
 &\approx \frac{1}{64} (V_n + \Delta V_n) \\
 &\times \left( 816I - 4812(AV_n + A\Delta V_n) \right. \\
 &\quad + 17393 \left( (AV_n)^2 + 2A(AV_n)^1 \Delta V_n \right) \\
 &\quad - 43044 \left( (AV_n)^3 + 3A(AV_n)^2 \Delta V_n \right) \\
 &\quad + 77154 \left( (AV_n)^4 + 4A(AV_n)^3 \Delta V_n \right) \\
 &\quad - 103312 \left( (AV_n)^5 + 5A(AV_n)^4 \Delta V_n \right) \\
 &\quad + 105039 \left( (AV_n)^6 + 6A(AV_n)^5 \Delta V_n \right) \\
 &\quad - 81576 \left( (AV_n)^7 + 7A(AV_n)^6 \Delta V_n \right) \\
 &\quad + 48268 \left( (AV_n)^8 + 8A(AV_n)^7 \Delta V_n \right) \\
 &\quad - 21516 \left( (AV_n)^9 + 9A(AV_n)^8 \Delta V_n \right) \\
 &\quad + 7071 \left( (AV_n)^{10} + 10A(AV_n)^9 \Delta V_n \right) \\
 &\quad - 1652 \left( (AV_n)^{11} + 11A(AV_n)^{10} \Delta V_n \right) \\
 &\quad + 258 \left( (AV_n)^{12} + 12A(AV_n)^{11} \Delta V_n \right) \\
 &\quad - 24 \left( (AV_n)^{13} + 13A(AV_n)^{12} \Delta V_n \right) \\
 &\quad + \left( (AV_n)^{14} + 14A(AV_n)^{13} \Delta V_n \right) \\
 &\approx V_{n+1} + \frac{51}{4} V_n - \frac{1203}{16} (AV_n)^1 V_n + \frac{17393}{64} (AV_n)^2 V_n \\
 &\quad - \frac{10761}{16} (AV_n)^3 V_n + \frac{38577}{32} (AV_n)^4 V_n \\
 &\quad - \frac{6457}{4} (AV_n)^5 V_n + \frac{105039}{64} (AV_n)^6 V_n - \frac{10197}{8} (AV_n)^7 V_n \\
 &\quad + \frac{12067}{16} (AV_n)^8 V_n - \frac{5379}{16} (AV_n)^9 V_n
 \end{aligned}$$

$$\begin{aligned}
 &+ \frac{7071}{64} (AV_n)^{10} V_n - \frac{413}{16} (AV_n)^{11} V_n + \frac{129}{32} (AV_n)^{12} V_n \\
 &\quad - \frac{3}{8} (AV_n)^{13} V_n + \frac{1}{64} (AV_n)^{14} V_n + \frac{51}{4} \Delta V_n - \frac{1203}{8} AV_n \Delta V_n \\
 &\quad + \frac{52179}{64} (AV_n)^2 \Delta V_n - \frac{10761}{4} (AV_n)^3 \Delta V_n \\
 &\quad + \frac{192885}{32} (AV_n)^4 \Delta V_n - \frac{19371}{2} (AV_n)^5 \Delta V_n \\
 &\quad + \frac{735273}{64} (AV_n)^6 \Delta V_n - 10197 (AV_n)^7 \Delta V_n \\
 &\quad + \frac{108603}{16} (AV_n)^8 \Delta V_n - \frac{26895}{8} (AV_n)^9 \Delta V_n \\
 &\quad + \frac{77781}{64} (AV_n)^{10} \Delta V_n - \frac{1239}{4} (AV_n)^{11} \Delta V_n \\
 &\quad + \frac{1677}{32} (AV_n)^{12} \Delta V_n - \frac{21}{4} (AV_n)^{13} \Delta V_n \\
 &\quad + \frac{15}{64} (AV_n)^{14} \Delta V_n,
 \end{aligned} \tag{33}$$

where  $(\Delta V_n)^i \approx 0, i \geq 2$  has been used, since they are very close to the zero (matrix). After some algebraic manipulation and using  $\Delta V_{n+1} \approx \tilde{V}_{n+1} - V_{n+1}$ , we have

$$\begin{aligned}
 &\Delta V_{n+1} \\
 &\approx \frac{3}{64} (4I - AV_n)^2 (I - AV_n)^{11} (17I - 5AV_n) \Delta V_n \\
 &\approx \frac{3}{64} \left[ (3I + I - AV_n)^2 (I - AV_n)^{11} (12I + 5(I - AV_n)) \right] \Delta V_n, \\
 &\approx \frac{3}{64} \left[ 108(I - AA^\dagger)^{11} + 117(I - AA^\dagger)^{12} \right. \\
 &\quad \left. + 42(I - AA^\dagger)^{13} + 5(I - AA^\dagger)^{14} \right] \Delta V_n,
 \end{aligned} \tag{34}$$

by using the fact that for enough large  $n$ , we have  $V_n \approx A^\dagger$ . We attain

$$\begin{aligned}
 \|\Delta V_{n+1}\| &\leq \frac{3}{64} \left[ 108\|I - AA^\dagger\|^{11} + 117\|I - AA^\dagger\|^{12} \right. \\
 &\quad \left. + 42\|I - AA^\dagger\|^{13} + 5\|I - AA^\dagger\|^{14} \right] \|\Delta V_n\|.
 \end{aligned} \tag{35}$$

From (35), we can conclude that the perturbation at the iterate  $n + 1$ , is bounded. Therefore, the sequence  $\{V_n\}_{n=0}^{n=\infty}$  generated by (7) is asymptotically stable. This ends the proof.  $\square$

**Corollary 6.** *By using the matrix identity in (27), it would be possible to further simplify bound (35) as follows:*

$$\|\Delta V_{n+1}\| \leq \left( \frac{51}{4} \|I - AA^\dagger\| \right) \|\Delta V_n\|. \tag{36}$$

- (1) Given  $A \in \mathbb{C}^{m \times m}$
- (2) construct the matrix  $V_0 = (1/(\|A\|_F^2)) A^*$
- (3) obtain the LU decomposition of the matrix  $A$ , for example  $LU = \text{LinearSolve}[A]$
- (4) for  $l = 1, \dots, j$  ( $j \ll n$ ), and when  $e_l$  is the  $l$ th column of the identity matrix
- (5) update the columns of  $V_0$  by finding the columns of the real  $A^{-1}$  as follows:  $v_l = LU[e_l]$
- (6) end for.

ALGORITHM 1: An algorithm for constructing a rapid and robust initial approximation for  $A^{-1}$ .

Using (36) recursively, one may attain the following very simple bound:

$$\|\Delta V_{n+1}\| \leq \left[ \left( \frac{51}{4} \right)^{n+1} \|I - AA^\dagger\| \right] \|\Delta V_0\|. \quad (37)$$

*Remark 7.* In case of finding the regular inverse of nonsingular matrices, that is, when  $A^\dagger = A^{-1}$ , according to (37), we have  $\Delta V_{n+1} \approx 0$ , and so the matrix method is strongly numerically stable. Consequently, in case of finding the  $A^\dagger$ , the matrix method (7), is asymptotically stable. Of course, since the iteration is not self-correcting in the general case, proceeding beyond convergence may cause a serious increase in error.

#### 4. Initial Value

The iterative methods that have been discussed up to now are sensitive upon choosing the initial guess/value to start the process. As a matter of fact, the high accuracy and efficiency of such types of iterative algorithms are guaranteed only if the initial value satisfies the appropriate condition given in Theorem 1. Thus, in order to preserve the convergence order, we present some ways from the literature to remedy this flaw, although an efficient way for square or rectangular matrices is the way (19).

For a symmetric positive definite (SPD) matrix  $A$ , one can easily use the Householder-John Theorem to attain

$$V_0 = P^{-1} \quad (38)$$

as the initial value, wherein the matrix  $P$  is any of the matrices such that  $P + P^T - A$  is SPD [8].

If the square matrix  $A$  is diagonally dominant, one may apply the approach given in [18] and use

$$V_0 = \text{diag} \left( \frac{1}{a_{11}}, \frac{1}{a_{22}}, \dots, \frac{1}{a_{mm}} \right), \quad (39)$$

wherein  $a_{ii}$  is the diagonal entry of  $A$ . Note that this choice is so much fruitful in solving PDEs resulting from discretizations. Some further generalizations of such an initial matrix are given in [19].

Although the two abovementioned ways are efficient, they cannot be applied for finding an initial guess/value to the inverse of general input matrices. For instance, for large-scale matrices which do not satisfy the above structures, they may fail to provide the convergence. Hence, we here take into

account the suboptimal way of producing  $V_0$  as given by Pan and Schreiber in [20] as follows:

$$V_0 = \frac{A^T}{\|A\|_1 \|A\|_\infty}. \quad (40)$$

We should note that choosing the initial value as given above can satisfy the necessary condition of arriving to the convergence phase. Some ways for updating the initial matrix for sparse matrices are brought forward by [21].

In what follows, we provide an algorithm for improving an initial matrix for square matrices rapidly. In fact, the derivation of LU factorization for almost all kinds of square nonsingular matrices could be done rapidly in the linear algebra programming packages. In the Mathematica, the one argument command `LinearSolve[]` provides an LU factorization of  $A$  too much fast. Then, by applying the LU decomposition on the columns of a identity matrix recursively, one could update the columns of a derived initial matrix produced by other strategies such as (40). Such a procedure is illustrated in Algorithm 1. We summarized this idea as in Algorithm 2.

The `initial1[A, num.]` takes the nonsingular matrix  $A$  and the number of columns that users wish to update from the real inverse into the approximate inverse from the left, while the function `initial2[A, num.]` works doubly. That is, if, for example, `num = 10`, it updates the first and last 10 columns of the approximate inverse as rapidly as possible.

Next, we conduct some numerical tests to support the theoretical results given in Section 2 using the initial value discussed herein.

#### 5. Computational Aspects

In this section, some experiments are presented to demonstrate the capability of the proposed method. The programming package MATHEMATICA 8 [22] has been used in the demonstrations. We work on the numerical aspects of the methods in machine precision.

It is clear that large sparse matrices cannot be handled easily and their storage needs to be done in sparse form as in the input form to be accessible and economic in real applications. Methods like (7) are powerful in finding an approximate inverse or a robust approximate inverse preconditioner in low number of steps and computational

```

initial1[A_, num_] :=
  Quiet@Module[{n = Dimensions[A][[1]], i = 1, LU = LinearSolve[A]},
    Id = SparseArray[{{k_, k_} -> 1.}, {n, n}];
    mat = (1/Norm[A, "Frobenius"]^2)*ConjugateTranspose[A];
    While[i <= num, {ith = LU[Id[[All, n + 1 - i]]];
      mat[[All, n + 1 - i]] = ith; i++;}; mat];
initial2[A_, num_] := Quiet@Module[{n = Dimensions[A][[1]], i = 1,
  Id = SparseArray[{{k_, k_} -> 1.}, {n, n}, 0]; LU = LinearSolve[A];
  mat = Quiet[initial1[A, num]];
  While[i <= num, {ith = LU[Id[[All, i]]];
    mat[[All, i]] = ith; i++;}; mat];

```

ALGORITHM 2: Two-argument function written in the Mathematica environment.

```

m = 1500; k = 1800; number = 30; SeedRandom[1234];
Table[A[1] = SparseArray[Band[{400, 10}], {m, k} -> Random[] - I,
  Band[{10, 200}], {m, k} -> {1.1, -Random[]}],
  Band[{-60, 100}] -> -3.02, Band[{-100, 500}] -> 3.1 I], {m, k}, 0.];, {1, number}];
Id = SparseArray[{{i_, i_} -> 1.}, {m, m}, 0];

```

ALGORITHM 3

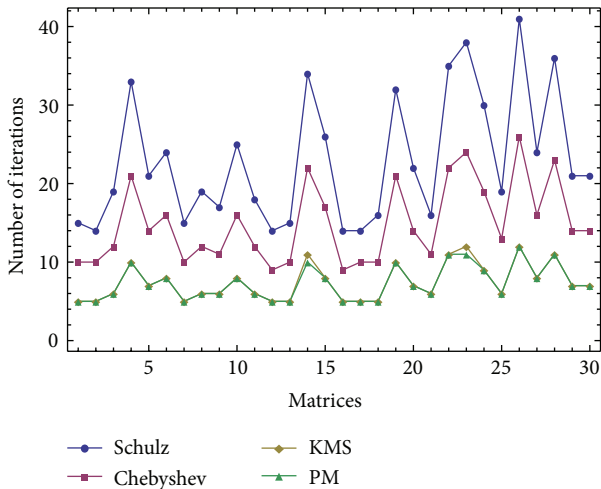


FIGURE 1: The results of comparisons for Experiment 1 in terms of the number of iterations.

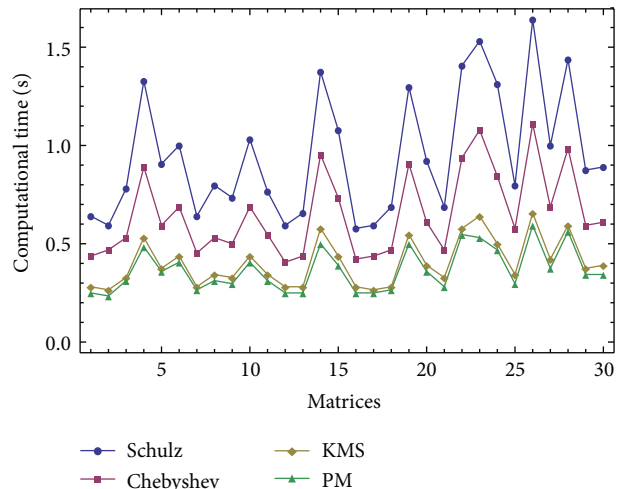


FIGURE 2: The results of comparisons for Experiment 1 in terms of the elapsed time using  $\|V_{n+1} - V_n\|_{\infty} \leq 10^{-6}$ .

time, in which the output form of the approximate inverse is also *sparse*.

In this paper, as the programs were running, we found the running time using the command `AbsoluteTiming[]` to report the elapsed CPU time (in seconds) for this experiment. In this paper, the computer specifications are Microsoft Windows XP Intel(R), Pentium(R) 4, and CPU 3.20 GHz, with 4 GB of RAM.

*Experiment 1.* This test is devoted to the application of the Schulz-type iterative methods in finding the pseudoinverse of 30 large random complex matrices defined as shown in Algorithm 3 ( $I = \sqrt{-1}$ ).

The results of comparisons for these random matrices of the size  $m \times k = 1500 \times 1800$  are reported in Figures 1, 2, and 3 in terms of the number of iterations and the computational time. The compared methods are (1) denoted by “Schulz,” (2) denoted by “Chebyshev,” (4) denoted by “KMS;” and the new iterative scheme (7) denoted by “PM.” A saving in the elapsed time by considering the stopping criteria as  $\|V_{n+1} - V_n\|_{\infty} \leq 10^{-6}$  and  $\|V_{n+1} - V_n\|_F \leq 10^{-6}$  can be observed for the studied method (7). In this test, the initial matrix has been computed for each random matrix by  $V_0[j] = \text{ConjugateTranspose}[A[j]] * (1./((\text{SingularValueList}[A[j], 1][[1]]^2))$ , while the maximum number of iterations is set to 100.



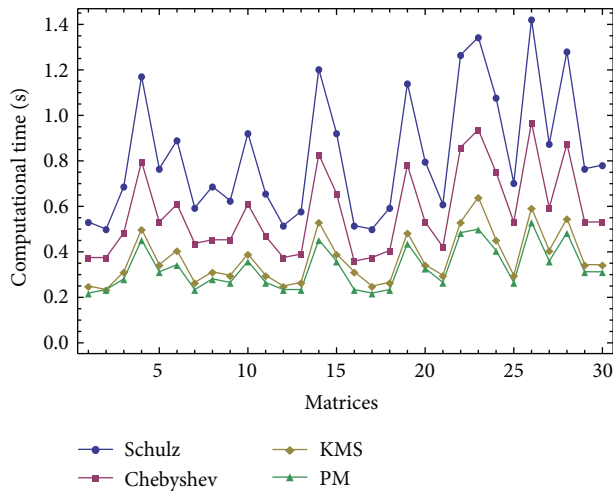


FIGURE 3: The results of comparisons for Experiment 1 in terms of the elapsed time using  $\|V_{n+1} - V_n\|_F \leq 10^{-6}$ .

## 6. Concluding Remarks

It is well known that matrix inverse and generalized inverse are important in applied fields of nature science, such as the solution to various systems of linear and nonlinear equations, eigenvalue problems, and the linear least square problems. Iterative methods are often effective especially for large-scale systems with sparsity and Schulz-type methods are great tools for preconditioning such systems or in finding pseudoinverses.

Hotelling-Bodewig algorithm is simple to describe and analyze and is numerically stable. This was the idea of developing an iterative method of this type in this paper.

In this paper, we have shown that the suggested method (7) reaches twelfth order of convergence. The stability of the new method was also studied in detail and established that the new scheme is asymptotically stable. The efficacy of the new scheme was illustrated numerically in Section 5. Finally, and based on the numerical results obtained, one can conclude that the presented method is useful. Further extensions of the new scheme for other generalized inverses (such as the ones in [23, 24]) can be done for future works.

## Conflict of Interests

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

## Acknowledgment

The research of the first author (F. Khaksar Haghani) is financially supported by Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.

## References

[1] I. V. Oseledets and E. E. Tyrtshnikov, "Approximate inversion of matrices in the process of solving a hypersingular integral equation," *Computational Mathematics and Mathematical Physics*, vol. 45, no. 2, pp. 315–326, 2005.

- [2] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd edition, 2003.
- [3] <http://www.computational.unibas.ch/software/spai/spaidoc.html>.
- [4] S. K. Sen and S. S. Prabhu, "Optimal iterative schemes for computing the Moore-Penrose matrix inverse," *International Journal of Systems Science*, vol. 7, no. 8, pp. 847–852, 1976.
- [5] F. Soleymani and P. S. Stanimirović, "A higher order iterative method for computing the Drazin inverse," *The Scientific World Journal*, vol. 2013, Article ID 708647, 11 pages, 2013.
- [6] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 7, pp. 498–520, 1933.
- [7] G. Schulz, "Iterative berechnung der reziproken matrix," *Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, no. 1, pp. 57–59, 1933.
- [8] 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.
- [9] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms: Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, India, 1986.
- [10] X. Liu and S. Huang, "Proper splitting for the generalized inverse  $A_{T,S}^{(2)}$  and its application on Banach spaces," *Abstract and Applied Analysis*, vol. 2012, Article ID 736929, 9 pages, 2012.
- [11] I. Pavaloiu and E. Catina, "Remarks on some Newton and Chebyshev-type methods for approximation eigenvalues and eigenvectors of matrices," *Computer Science Journal of Moldova*, vol. 7, pp. 3–17, 1999.
- [12] A. R. Soheili, F. Soleymani, and M. D. Petkovic, "On the computation of weighted Moore-Penrose inverse using a high-order matrix method," *Computers and Mathematics with Applications*, vol. 66, no. 11, pp. 2344–2351, 2013.
- [13] F. Soleymani and D. K. R. Babajee, "Computing multiple roots using a class of quartically convergent methods," *Alexandria Engineering Journal*, vol. 52, no. 3, pp. 531–541, 2013.
- [14] A. Ben-Israel and D. Cohen, "On iterative computation of generalized inverses and associated projections," *SIAM Journal on Numerical Analysis*, vol. 3, pp. 410–419, 1966.
- [15] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*, Springer, 2nd edition, 2003.
- [16] F. Toutounian and F. Soleymani, "An iterative method for computing the approximate inverse of a square matrix and the Moore-Penrose inverse of a non-square matrix," *Applied Mathematics and Computation*, vol. 224, pp. 671–680, 2013.
- [17] F. Soleymani and P. S. Stanimirović, "A note on the stability of a pth order iteration for finding generalized inverses," *Applied Mathematics Letters*, vol. 28, pp. 77–81, 2014.
- [18] L. Grosz, "Preconditioning by incomplete block elimination," *Numerical Linear Algebra with Applications*, vol. 7, no. 7-8, pp. 527–541, 2000.
- [19] L. Gonzalez and A. Suarez, "Improving approximate inverses based on Frobenius norm minimization," *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 9363–9371, 2013.
- [20] V. Y. Pan and R. Schreiber, "An improved Newton iteration for the generalized inverse of a matrix with applications," *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 5, pp. 1109–1131, 1991.

- [21] X. Cui and K. Hayami, "Generalized approximate inverse preconditioners for least squares problems," *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 1, pp. 1–14, 2009.
- [22] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.
- [23] F. Soleymani, P. S. Stanimirović, and M. Z. Ullah, "On an accelerated iterative method for weighted Moore-Penrose inverse," *Applied Mathematics and Computation*, vol. 222, pp. 365–371, 2013.
- [24] M. Z. Ullah, F. Soleymani, and A. S. Al-Fhaid, "An efficient matrix iteration for computing weighted Moore-Penrose inverse," *Applied Mathematics and Computation*, vol. 226, pp. 441–454, 2014.



# Hindawi

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

