Commun. Korean Math. Soc. **28** (2013), No. 2, pp. 407–418
http://dx.doi.org/10.4134/CKMS.2013.28.2.407

ON A FAST ITERATIVE METHOD FOR APPROXIMATE INVERSE OF MATRICES

Fazlollah Soleymani

Abstract. This paper studies a computational iterative method to find accurate approximations for the inverse of real or complex matrices. The analysis of convergence reveals that the method reaches seventh-order convergence. Numerical results including the comparison with different existing methods in the literature will also be considered to manifest its superiority in different types of problems.

1. Introduction

The well-known method of Schulz [16], which is defined by

$$V_{n+1} = V_n(2I - AV_n), \quad n = 0, 1, 2, \ldots, \tag{1.1}$$

for the inversion of a matrix $A \in \mathbb{C}^{m\times m}$, was proposed in 1933. This method is also referred to the *Hotelling-Bodewig algorithm* [7], while it is sometimes called as the *hyper-power* iterative method [6].

In fact, Schulz in his fundamental work does not recommend (1.1) for arbitrary matrices. It is interesting that he considered an example of a Toeplitz matrix, since its structure admits cheap matrix operations. That is, a clear saving in the computational burden could be done for matrices possessing special structures such as Vandermonde, Cauchy, and Hankel matrices.

The scheme (1.1) is a matrix-by-matrix multiplication iterative method. Displacement representations of a special structured matrix enable its fast multiplication by a vector. In the literature, it is known a great number of iterative methods for computation of the usual matrix inverse, the Moore-Penrose inverse and their applications (see e.g. [1], [11], [12] and [14]).

On the other hand, matrix inversion plays a significant role in computer graphics, particularly in 3D graphics rendering and 3D simulations. Examples include screen-to-world ray casting, world-to-subspace-to-world object transformations, and physical simulations. All such needs along the preconditioning for both dense and sparse matrices, encouraged mathematicians to develop new

Received March 6, 2012.

2010 *Mathematics Subject Classification.* 15A09, 65F10, 65F50.

Key words and phrases. Hotelling-Bodewig algorithm, ill-conditioned, approximate inverse, initial matrix.

schemes or to modify the existing ones. For example, dense matrices arise, in numerical solution of multidimensional integral equations, and their approximate inverses are often of interest either themselves or as preconditioners in iterative methods (see for preconditioning [3] or [18]), and the size of matrices occurs to be about a few hundred of thousands.

As another application of matrix inversion based on [5], we have the role of matrix inverse in the MIMO (Multiple-Input, Multiple-Output) technology in wireless communications. The MIMO system consists of N transmit and M receive antennas. Unique signals, occupying the same frequency band, are sent via N transmit antennas and are received via M receive antennas. The signal arriving at each receive antenna will be a linear combination of the N transmitted signals forming a $N \times M$ transmission matrix H. It is crucial for the matrix H to be invertible for the receiver to be able to figure out the transmitted information. Therefore, an efficient algorithm to find robust approximate inverses of this type of matrices is really a need.

A straightforward way to determine A^{-1} is to compute it explicitly. Using such an approach, the complexity of the method is $O(m^3)$, even when the sparsity is not exploited. Exploiting sparsity, ill-conditioning, and clever reordering techniques would yield a much more favorable complexity, [15]. This approach can be quite expensive for large matrices or matrices arising from a 3-D mesh problem particulary in solving Partial Differential Equations (PDEs). Luckily, in many cases the dense matrices possess some structures suggesting a way to make them tractable; and for sparse matrices, some threshold-based approaches can be some remedy to reduce the computational burden of matrix multiplications.

Now let us review some of the known methods. In 2010, Li and Li in [10] proposed

$$V_{n+1} = V_n(3I - 3AV_n + (AV_n)^2),\ n = 0, 1, 2, \dots, \tag{1.2}$$

wherein I denoted the identity matrix of the appropriate dimension. In 2011, Li et al. in [9] re-presented (1.2) in the following form

$$V_{n+1} = V_n(3I - AV_n(3I - AV_n)),\ n = 0, 1, 2, \dots, \tag{1.3}$$

while (1.2) and (1.3) are equal and they have previously studied in [8]. They also proposed another iterative method for finding A^{-1} as comes next:

$$V_{n+1} = \left[I + \frac{1}{4}(I - V_nA)(3I - V_nA)^2\right] V_n,\ n = 0, 1, 2, \dots. \tag{1.4}$$

Inspired and motivated by the classical method (1.1) for finding the inverse of a matrix A, we will study a high order method. The rest of this paper is organized as follows. The main study of this article is given in Section 2. It is also devoted to the analysis of convergence which shows that the method reaches a high rate of convergence. Extension of the method for singular or rectangular matrices will also be given. Subsequently, the method is examined in Section 3 numerically. Finally, concluding remarks will be presented in Section 4.

2. Main results

This section contains a new high order algorithm for finding A^{-1} numerically. In order to deal with ill-conditioned linear systems, or to find robust approximate inverses, one remedy is to rely on high-order Schulz-type methods. Applying the following four-step nonlinear solver

$$\begin{cases} y_n = x_n - f'(x_n)^{-1}f(x_n), \\ z_n = y_n - (f[y_n, x_n])^{-1}f(y_n), \\ w_n = z_n - (f[z_n, y_n])^{-1}f(z_n), \\ x_{n+1} = w_n - (f[w_n, y_n])^{-1}f(w_n), \ n = 0, 1, 2, \dots, \end{cases} \tag{2.1}$$

on the matrix equation $F(V) = V^{-1} - A$, where e.g. $f[z_n, y_n] = \frac{f(z_n)-f(y_n)}{z_n-y_n}$, is the two-point divided difference, gives us the aimed seventh-order method.

Hence, we suggest the following iteration method

$$\begin{aligned} V_{n+1} = V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I \\ +AV_n(21I + AV_n(-7I + AV_n)))))), \end{aligned} \tag{2.2}$$

for any $n = 0, 1, 2, \dots$, where I is the identity matrix, and the sequence of iterates $\{V_n\}_{n=0}^{n=\infty}$ converges to A^{-1} for a proper matrix V_0.

Theorem 2.1. *Let $A = [a_{ij}]_{m\times m}$ be a nonsingular real or complex matrix. If the initial approximation V_0 satisfies*

$$\|I - AV_0\| < 1, \tag{2.3}$$

then the iterative method (2.2) *converges with seventh order to A^{-1}.*

Proof. Let $\|I - AV_0\| < 1$, and also $E_0 = I - AV_0$. We thus have $E_n = I - AV_n$. Then

$$\begin{aligned} E_{n+1} &= I - AV_{n+1} \\ &= I - A(V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I \\ &\quad + AV_n(21I + AV_n(-7I + AV_n))))))) \\ &= I - [AV_n(7I - 21AV_n + 35(AV_n)^2 - 35(AV_n)^3 \\ &\quad + 21(AV_n)^4 - 7(AV_n)^5 + (AV_n)^6)] \\ &= I - (7AV_n - 21(AV_n)^2 + 35(AV_n)^3 - 35(AV_n)^4 \\ &\quad + 21(AV_n)^5 - 7(AV_n)^6 + (AV_n)^7 \\ &= \ - (I - AV_n)^7 \\ &= \ - E_n^7. \end{aligned} \tag{2.4}$$

In addition, since $\|E_0\| < 1$, by relation (2.4) we obtain that

$$\|E_{n+1}\| \leq \|E_n\|^7 \leq \|E_{n-1}\|^{7^2} \leq \cdots \leq \|E_0\|^{7^{n+1}}, \tag{2.5}$$

where (2.5) tends to zero when $n \to \infty$. That is, $I - AV_n \to 0$ when $n \to \infty$, and thus $V_n \to A^{-1}$ as $n \to \infty$. Now we show the seventh order of convergence using the sequence $\{V_n\}_{n=0}^{n=\infty}$. To do this, we denote $e_n = V_n - A^{-1}$, as the error matrix in the iterative procedure (2.2). We have

$$I - AV_{n+1} = -(I - AV_n)^7. \tag{2.6}$$

Hence, we could get that

$$A(A^{-1} - V_{n+1}) = -(A(A^{-1} - V_n))^7, \tag{2.7}$$

which yields

$$\|e_{n+1}\| \leq \|A\|^6 \|e_n\|^7. \tag{2.8}$$

Thus the iteration (2.2) converges with seventh order to A^{-1}. This concludes the proof. □

We now give another property about the scheme (2.2) under a tight condition.

Theorem 2.2. *Let $A = [a_{ij}]_{m\times m}$ be a nonsingular real or complex matrix. If $AV_0 = V_0A$ is valid, then for the sequence $\{V_n\}_{n=0}^{n=\infty}$ of* (1.4)*, we have that $AV_n = V_nA$ holds for all $n = 1, 2, \ldots$.*

Proof. First, since $AV_0 = V_0A$, we have from (2.2) that

$$\begin{aligned} AV_1 &= A(V_0(7I + AV_0(-21I + AV_0(35I + AV_0(-35I + AV_0(21I \\ &\quad + AV_0(-7I + AV_0))))))) \\ &= (V_0A(7I + V_0A(-21I + V_0A(35I + V_0A(-35I \\ &\quad + V_0A(21I + V_0A(-7I + V_0A))))))) \\ &= (V_0(7I + AV_0(-21I + AV_0(35I \\ &\quad + AV_0(-35I + AV_0(21I + AV_0(-7I + AV_0)))))))A \\ &= V_1A. \end{aligned}$$

That is, when $n = 1$, the conditions of the theorem hold. Now, we use the mathematical induction for proving the rest of the procedure. Suppose that $AV_n = V_nA$ is true. Then a straightforward calculation shows that, for all $n \geq 2$:

$$\begin{aligned} AV_{n+1} &= A(V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I + AV_n(21I \\ &\quad + AV_n(-7I + AV_n))))))) \\ &= (V_nA(7I + V_nA(-21I + V_nA(35I \\ &\quad + V_nA(-35I + V_nA(21I + V_nA(-7I + V_nA))))))) \\ &= (V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I \end{aligned}$$

$$\begin{aligned}&+ AV_n(21I + AV_n(-7I + AV_n)))))))A\\ &= V_{n+1}A.\end{aligned}$$

The proof is now complete. □

Note that the following initial matrix $V_0 = \alpha I$, where I is the identity matrix, and $\alpha \in \mathbb{R}$ must adaptively be determined such that $\|I - \alpha A\| < 1$, is one of the only ways, which guarantees the tight conditions of Theorem 2.2.

Let us now discuss on some other aspects of (2.2) in finding generalized inverse. The Moore-Penrose inverse of a complex matrix $A \in \mathbb{C}^{m\times k}$ (also called pseudo-inverse), denoted by $A^\dagger \in \mathbb{C}^{k\times m}$, is a matrix $X \in \mathbb{C}^{k\times m}$ satisfying the following conditions

$$AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA, \tag{2.9}$$

where A^* is the conjugate transpose of A. Note that such an inverse is uniquely exist (see [2] and [4] for more details). It is known that (1.1) converges to the pseudo-inverse in the general case if $V_0 := \alpha A^*$, where $0 < \alpha < 2/\rho(A^*A)$ and $\rho(\cdot)$ denotes the spectral radius.

The scheme (2.2) converges to the Moore-Penrose inverse, when having singular matrix as well. In order validate this analytically, we give the following theorem.

Theorem 2.3. *For the sequence $\{V_n\}_{n=0}^{n=\infty}$ generated by the iterative Schulz-type method* (2.2)*, for any $n \geq 0$, it holds that*

$$(AV_n)^* = AV_n, \quad (V_nA)^* = V_nA, \quad V_nAA^\dagger = V_n, \quad AA^\dagger V_n = V_n. \tag{2.10}$$

Proof. We will prove the conclusion by induction on n. For $n = 0$, by $V_0 = \alpha A^*$, the first two equations can be demonstrated simply, and we only give a verification to the last two equations in what follows:

$$V_0AA^\dagger = \alpha A^*AA^\dagger = \alpha A^*(AA^\dagger)^* = \alpha A^*(A^\dagger)^*A^* = \alpha(AA^\dagger A)^* = \alpha A^* = V_0,$$

$$A^\dagger AV_0 = (A^\dagger A)\alpha A^* = \alpha(A^\dagger A)^*A^* = \alpha A^*(A^\dagger)^*A^* = \alpha(AA^\dagger A)^* = \alpha A^* = V_0.$$

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 (2.2), one has

$$\begin{aligned}(AV_{n+1})^* &= (A(V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I\\ &\quad + AV_n(21I + AV_n(-7I + AV_n))))))))^*\\ &= [7AV_n - 21(AV_n)^2 + 35(AV_n)^3 - 35(AV_n)^4\\ &\quad + 21(AV_n)^5 - 7(AV_n)^6 + (AV_n)^7]^*\\ &= 7(AV_n)^* - 21((AV_n)^*)^2 + 35((AV_n)^*)^3 - 35((AV_n)^*)^4\\ &\quad + 21((AV_n)^*)^5 - 7((AV_n)^*)^6 + ((AV_n)^*)^7\\ &= A(V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I\\ &\quad + AV_n(21I + AV_n(-7I + AV_n)))))))\end{aligned}$$

$$= AV_{n+1},$$

where the following fact $(AV_n)^* = AV_n$ has been used. Thus, the first equality in (2.10) holds for $n+1$, and the second equality can be proved in a similar way. For the third equality in (2.10), using the assumption that $V_nAA^\dagger = V_n$ and the iterative method (2.2), we could write down

$$\begin{aligned}
V_{n+1}AA^\dagger &= (V_n(7I + AV_n(-21I + V_nAV_n(35I + V_nAV_n(-35I \\
&\quad + V_nAV_n(21I + AV_n(-7I + AV_n)))))))AA^\dagger \\
&= 7V_nAA^\dagger - 21AV_nAA^\dagger + 35V_n(AV_n)^2AA^\dagger - 35V_n(AV_n)^3AA^\dagger \\
&\quad + 21V_n(AV_n)^4AA^\dagger - 7V_n(AV_n)^5AA^\dagger + V_n(AV_n)^6AA^\dagger \\
&= 7V_n - 21AV_n + 35V_n(AV_n)^2 - 35V_n(AV_n)^3 \\
&\quad + 21V_n(AV_n)^4 - 7V_n(AV_n)^5 + V_n(AV_n)^6 \\
&= V_n(7I + AV_n(-21I + AV_n(35I + AV_n(-35I \\
&\quad + V_nAV_n(21I + AV_n(-7I + AV_n)))))) \\
&= V_{n+1}.
\end{aligned}$$

Hence the third equality in (2.10) holds for $n+1$. The fourth equality can similarly be proved, and the desired result follows. □

3. Applications

The applicability of such Schulz-type iterative methods could be restricted since global convergence is not inherent to all initial matrices. This motivated many authors to propose general or special ways for selecting the initial matrix in order to achieve the convergence properties. For instance authors in [13] proposed

$$V_0 = \frac{A^T}{\|A\|_1\|A\|_\infty}, \tag{3.1}$$

for an $m \times m$ matrix A, where T stands for transpose, $\|A\|_1 = \max_j(\Sigma_{i=1}^m|a_{ij}|)$ and $\|A\|_\infty = \max_i(\Sigma_{j=1}^m|a_{ij}|)$. This choice yields $\|I - AV_0\|_2 \leq 1 - 1/(n\kappa^2)$, where $\kappa = \sigma_{\max}/\sigma_{\min}$. Moreover, for an $m \times m$ symmetric positive definite matrix A, by choosing

$$V_0 = I/\|A\|_F, \tag{3.2}$$

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

For numerical comparisons in this section, we have used the methods (1.1), (1.3), (1.4) and (2.2). Now we implement the above algorithms on the following examples. We used Mathematica 8 [17] in calculations. In this paper, the computer specifications are Intel(R) Core(TM) 2 Quad CPU, Q9550 @ 2.83GHz with 2.00GB of RAM. The elapsed time in seconds for running the Mathematica 8 codes will be reported using the command `AbsoluteTiming[]`.

Example 1. Let us consider finding the approximate inverse of a 100×100 Hankel matrix (a Hankel matrix or catalecticant matrix, named after Hermann Hankel, is a square matrix with constant skew-diagonals, i.e., positive sloping diagonals). If the i, j element of A is denoted $A_{i,j}$, then we have $A_{i,j} = A_{i-1,j+1}$. In MATHEMATICA, it could be defined by `n = 100`; `A = N[HankelMatrix[n]]`;.

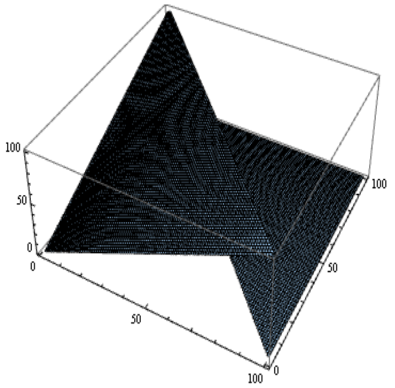

FIGURE 1. The `ListPlot3D` of the Hankel matrix in Example 1.

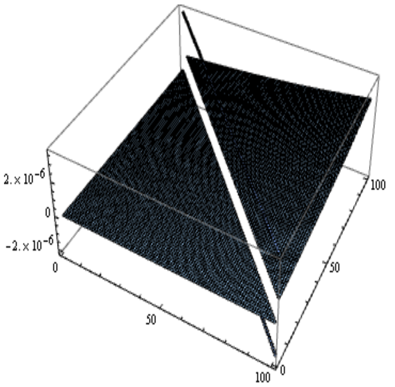

FIGURE 2. The `ListPlot3D` of the approximate inverse in Example 1.

The Hankel matrix is closely related to the Toeplitz matrix. In fact, a Hankel matrix is an upside-down Toeplitz matrix. Hankel matrices are formed when given a sequence of output data and a realization of an underlying state-space or hidden Markov model is desired.

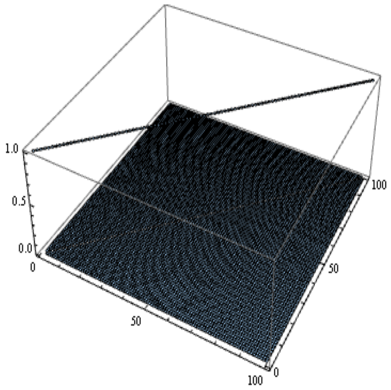

FIGURE 3. The `ListPlot3D` of the matrix `Chop[A.V, 10`$^{-8}$`]` in Example 1.

The list 3D plots of the A, its approximate inverse and the multiplication of these two matrices to observe an identity-like matrix output are given in Figures 1-3, respectively. In this example, the stopping criterion is $\|I - V_n A\|_1 \leq 1.E{-6}$ in double precision arithmetic. The results of comparisons are reported in Table 1. It shows the number of iterations for different methods in order to reveal the efficiency of the proposed iteration. Note that the initial guess has been constructed using (3.1).

The piece of code for implementing the scheme (2.2) in MATHEMATICA 8 is given as follows:

```
Clear["Global‘*"]
n = 100; A = N[HankelMatrix[n]];
Id = SparseArray[{{i_, i_} -> 1.}, {n, n}, 0.];
V = Transpose[A]/(Norm[A, 1]*Norm[A, Infinity]);
k1 = N[Norm[Id - A.V]];
k2 = N[Norm[Id - A.V, 1]];
k3 = N[Norm[Id - A.V, Infinity]];
If[k1 <= 1 || k2 <= 1 || k3 <= 1,
   {k = 1; While[k <= 75 && N[Norm[Id - V.A, 1]] >= 10^-6,
     V1 = SparseArray[V]; VV = A.V1;
     V = Chop[V1.SparseArray[7 Id + VV.(-21 Id + VV.(35 Id
              + VV.(-35 Id + VV.(21 Id + VV.(-7 Id + VV)))))],
       10^-8]; k++];
    Print["The approximate inverse is:" SparseArray[V]];
    Print["The number of iterations is:" (k-1)];},
   Print["The method will mostly fail to converge due to badly
    chosen initial guess."]]; // AbsoluteTiming
```

Table 1. Results of comparisons for Example 1.

Methods	(1.1)	(1.3)	(1.4)	(2.2)
Iteration number	18	11	11	7
The running time	0.34	0.21	0.22	0.15

Example 2. In order to deal with ill conditioned matrices, in this example we use 32 digits floating point arithmetics in our calculations so as to keep the effect of round-off error at minimum. Let us consider Hilbert matrix of the size $n = 14$ with the condition number 4.53776×10^{19}.

The results of comparisons are reported in Table 2 for finding its inverse by different Schulz-type iterative methods. The initial guess is constructed by (3.1). In this example, the stopping criterion is $\|I - V_n A\|_1 \leq 1.E-6$. Similar to Example 1, our method for such ill-conditioned matrices could be coded in what follows:

```
Clear["Global`*"]
n = 14; digits = 32;
A = SetAccuracy[HilbertMatrix[n], digits];
Id = SetAccuracy[SparseArray[{{i_, i_} -> 1.}
    , {n, n}, 0], digits];
V = SetAccuracy[Transpose[A]/(Norm[A, 1]
      *Norm[A, Infinity]), digits];
k1 = N[Norm[Id - A.V]];
k2 = N[Norm[Id - A.V, 1]];
k3 = N[Norm[Id - A.V, Infinity]];
If[k1 <= 1 || k2 <= 1 || k3 <= 1,
  {k = 1; While[k <= 1000 && N[Norm[Id - V.A, 1]] >= 10^-6,
    VV = SetAccuracy[A.V, digits];
    V = SetAccuracy[V.(7 Id + VV.(-21 Id + VV.(35 Id
              + VV.(-35 Id + VV.(21 Id
                   + VV.(-7 Id + VV)))))), digits]; k++];
   Print["The number of iterations is:" (k - 1)];},
  Print["The method will mostly fail to converge due to badly
   chosen initial guess."]]; // AbsoluteTiming
```

Table 2. Results of comparisons for Example 2.

Methods	(1.1)	(1.3)	(1.4)	(2.2)
Iteration number	134	85	79	48
The running time	2.29	1.98	2.26	2.10

Example 3. This test is devoted to the application of the Schulz-type iterative methods in finding the pseudo-inverse of some large random matrices defined as follows.

```
k = 301; m = 300; number = 10;
Table[A[j] = RandomReal[{-100, 10}, {m, k}], {j,
    number}]; // AbsoluteTiming
Id = SparseArray[{{i_, i_} -> 1.}, {m, m}, 0.];
```

The results of comparisons for 10 different random matrices in double precision arithmetics of the size $m \times k = 300 \times 301$ are reported in Figures 4-5 in terms of the number of iterations and the computational times. The compared methods are, (1.1) denoted by "Schulz", (1.3) denoted by "KMS" and the new iterative scheme (2.2) denoted by "PM". A clear saving in the elapsed time by considering the stopping criterion as $||\mathtt{V}_{\mathtt{n+1}} - \mathtt{V}_{\mathtt{n}}||_1 \leq \mathtt{10}^{-6}$ can be observed for the studied method (2.2). In this test, the initial matrix has been computed for each random matrix by $\mathtt{V}_0[\mathtt{j}] = \mathtt{ConjugateTranspose}[\mathtt{A}[\mathtt{j}]] * (\mathtt{1.}/(\mathtt{Norm}[\mathtt{A}[\mathtt{j}], \mathtt{2}]^{\mathtt{2}}))$.

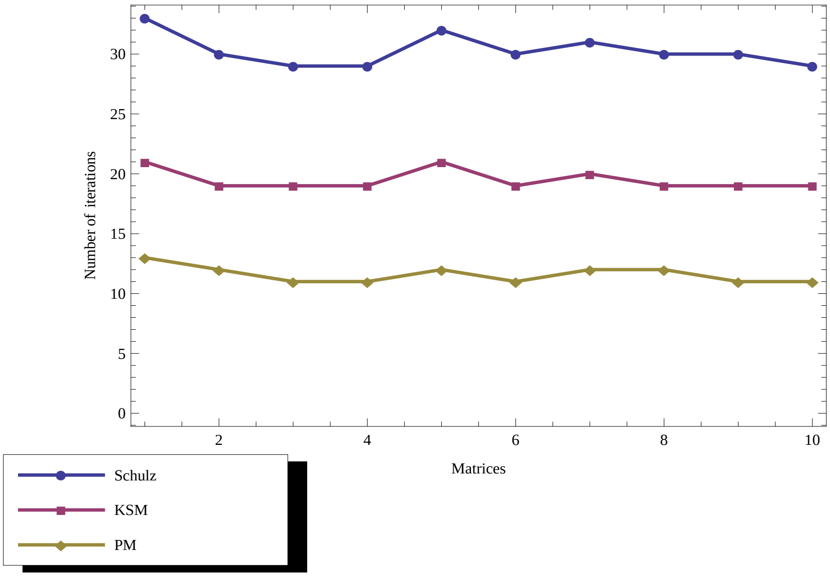

FIGURE 4. The results of comparisons for Example 3 in terms of the number of iterations.

4. Summary

In this paper, we have studied an iterative method in inverse-finding of singular or nonsingular matrices, which could be real or complex. We have shown that the suggested method (2.2) reaches seventh order of convergence,

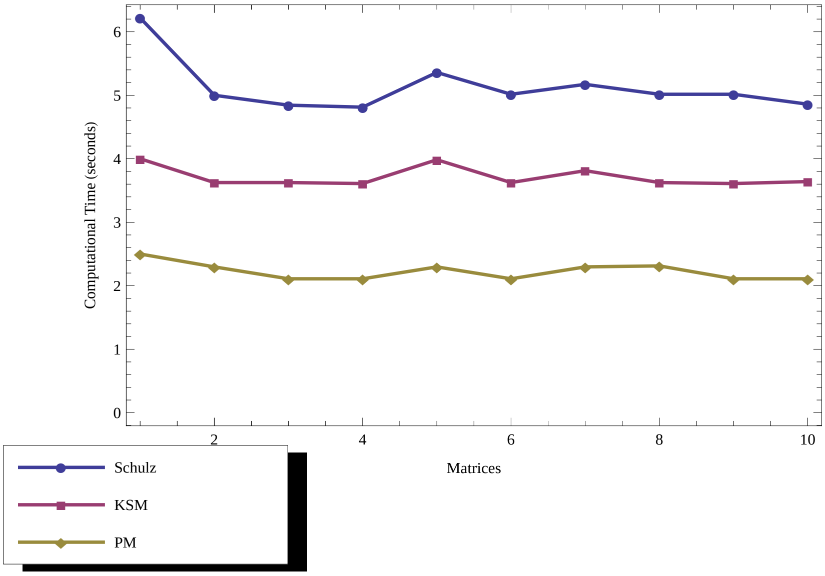

FIGURE 5. The results of comparisons for Example 3 in terms of the elapsed time.

while a discussion on finding the Moore-Penrose inverse has also been given. The applicability of the new scheme was illustrated numerically in Section 3.

It should be remarked that we have used the threshold $\mathtt{Chop}[\mathtt{exp}, \mathtt{tol}]$ with $\mathtt{tol} = 10^{-8}$ in Example 1. In fact, this command provides an amazing tool at hand for making the process so much fast for sparse matrices by avoiding unnecessary multiplications. By choosing a tolerance like 0.001, we would be able to make necessary zeros in the approximate inverse and thus the sparsity of the approximate inverse would be high enough to accelerate the matrix by matrix multiplications for sparse matrices.

Finally, according to the numerical results obtained and the concrete application of such solvers, we can conclude that the method is rapid.

References

[1] A. Ben-Israel and D. Cohen, *On iterative computation of generalized inverses and associated projections*, SIAM J. Numer. Anal. **3** (1966), 410–419.

[2] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*, Springer, 2nd edition, 2003.

[3] W. Cao and B. Guo, *Preconditioning for the p-version boundary element method in three dimension with tringaular elements*, J. Korean Math. Soc. **41** (2004), no. 2, 345–368.

[4] H. Chen and Y. Wang, *A family of higher-order convergent iterative methods for computing the Moore-Penrose inverse*, Appl. Math. Comput. **218** (2011), no. 8, 4012–4016.

[5] en.wikipedia.org/wiki/Invertible_matrix.

[6] J. M. Garnett, A. Ben-Israel, and S. S. Yau, *A hyperpower iterative method for computing matrix products involving the generalized inverse*, SIAM J. Numer. Anal. **8** (1971), 104–109.

[7] H. Hotelling, *Analysis of a complerx statistocal variable into principal components*, J. Educ. Psysh. **24** (1933), 498–520.

[8] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms*, Computations in science and engineering.Affiliated East-West Press Pvt. Ltd., New Delhi, 1986.

[9] H.-B. Li, T.-Z. Huang, Y. Zhang, X.-P. Liu, and T.-X. Gu, *Chebyshev-type methods and preconditioning techniques*, Appl. Math. Comput. **218** (2011), no. 2, 260–270.

[10] W. Li and Z. Li, *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. **215** (2010), no. 9, 3433–3442.

[11] M. Monsalve and M. Raydan, *A new inversion-free method for a rational matrix equation*, Linear Algebra Appl. **433** (2010), no. 1, 64–71.

[12] Y. Nakatsukasa, Z. Bai, and F. Gygi, *Optimizing Halley's iteration for computing the matrix polar decomposition*, SIAM. J. Matrix Anal. Appl. **31** (2010), no. 5, 2700–2720.

[13] V. Pan and R. Schreiber, *An improved Newton iteration for the generalized inverse of a matrix, with applications*, SIAM J. Sci. Statist. Comput. **12** (1991), no. 5, 1109–1130.

[14] I. Pavaloiu and E. Catina, *Remarks on some Newton and Chebyshev-type methods for approximation eigenvalues and eigenvectors of matrices*, Comput. Sci. J. Moldova **7** (1999), no. 1, 3–17.

[15] S. M. Rump, *Inversion of extremely ill-conditioned matrices in floating-point*, Japan J. Indust. Appl. Math. **26** (2009), no. 2-3, 249–277.

[16] G. Schulz, *Iterative Berechnung der Reziproken matrix*, Z. Angew. Math. Mech. **13** (1933), 57–59.

[17] S. Wolfram, *The Mathematica Book*, 5th edition, Wolfram Media, 2003.

[18] J. H. Yun, *Comparison results for the preconditioned Gauss-Seidel methods*, Commun. Korean Math. Soc. **27** (2012), no. 1, 207–215.

Department of Mathematics
Islamic Azad University
Zahedan Branch, Zahedan 98168, Iran
E-mail address: fazlollah.soleymani@gmail.com