

Research Article

A Rapid Numerical Algorithm to Compute Matrix Inversion

F. Soleymani

Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

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

Received 22 March 2012; Revised 1 June 2012; Accepted 1 June 2012

Academic Editor: Taekyun Kim

Copyright © 2012 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.

The aim of the present work is to suggest and establish a numerical algorithm based on matrix multiplications for computing approximate inverses. It is shown theoretically that the scheme possesses seventh-order convergence, and thus it rapidly converges. Some discussions on the choice of the initial value to preserve the convergence rate are given, and it is also shown in numerical examples that the proposed scheme can easily be taken into account to provide robust preconditioners.

1. Introduction

Let us consider the square matrix $A_{N \times N}$ with real or complex elements which is nonsingular. It is well known that its inverse is available and could be found by the direct methods such as LU or QR decompositions, see for example [1]. When the matrix inverse is computed, the method of choice should be probably Gaussian elimination with partial pivoting (GEPP). The resulting residual bounds and possible backward errors may be much smaller in this case, see [2] (subsection on the "Use and abuse of the matrix inverse").

An effective tool to compute approximate inverses of the matrix A is to use iteration-type methods for this purpose which are based on matrix multiplications and are of great interest and accuracy when implementing on parallel machines. In fact, one way is to construct iterative methods of high order of convergence to find matrix inversion numerically for all types of matrices (especially for ill-conditioned ones).

A clear use of such schemes is that one may apply them to find A^{-1} and then, by an easy matrix-vector multiplication, compute the solution of the linear system of the equations $Ax = b$. However another use is in constructing approximate inverse preconditioners; that is,

a very robust approximate preconditioner can easily be constructed using one, two, or three steps of such iterative methods, and the resulting left preconditioned systems would be

$$\bar{A}x = \bar{b}, \quad (1.1)$$

wherein $\bar{A} = P^{-1}A$, $\bar{b} = P^{-1}b$, and $P^{-1} \approx A^{-1}$.

Such obtained approximate inverse preconditioners could be robust competitors to the classical or modern methods such as AINV or FAPINV; see for example [3, 4]. The approximate inverse (AINV) and the factored approximate inverse (FAPINV) are two known algorithms in the field of preconditioning of linear systems of equations. Both of these algorithms compute a sparse approximate inverse of matrix A in the factored form and are based on computing two sets of vectors which are A biconjugate.

In this paper, in order to challenge with very ill-conditioned matrices or to find the preconditioner P^{-1} in less number of iterations and having high accuracy, we will propose an efficient iterative method for finding A^{-1} numerically. Theoretical analysis and numerical experiments show that the new method is more effective than the existing ones in the case of constructing approximate inverse preconditioners.

The rest of the paper is organized as follows. Section 2 is devoted to a brief review of the available literature. The main contribution of this paper is given in Section 3. Subsequently, the method is examined in Section 4. Finally, concluding remarks are presented in Section 5.

2. Background

Several methods of various orders were proposed for approximating (rectangular or square) matrix inverses, such as those according to the minimum residual iterations [5] and Hotelling-Bodewig algorithm [6].

The Hotelling-Bodewig algorithm [6] is defined as

$$V_{n+1} = V_n(2I - AV_n), \quad n = 0, 1, 2, \dots, \quad (2.1)$$

where I is the identity matrix. Note that throughout this paper we consider matrices of the same dimension unless it is stated obviously.

In 2011, Li et al. in [7] presented the following third-order method:

$$V_{n+1} = V_n(3I - AV_n(3I - AV_n)), \quad n = 0, 1, 2, \dots, \quad (2.2)$$

and also proposed another third-order iterative method for approximating A^{-1} as comes next

$$V_{n+1} = \left[I + \frac{1}{4}(I - V_n A)(3I - V_n A)^2 \right] V_n, \quad n = 0, 1, 2, \dots \quad (2.3)$$

It is interesting to mention that the method (2.2) can be found in the Chapter 5 of [8].

As an another method from the existing literature, Krishnamurthy and Sen suggested the following sixth-order iteration method [8] for the above purpose:

$$\begin{aligned}
 V_{n+1} &= V_n(2I - AV_n)(3I - AV_n(3I - AV_n))(I - AV_n(I - AV_n)) \\
 &= V_n(I + (I - AV_n)(I + (I - AV_n))))))),
 \end{aligned}
 \tag{2.4}$$

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

For further reading refer to [9, 10].

3. An Accurate Seventh-Order Method

This section contains a new high-order algorithm for finding A^{-1} numerically. In order to deal with very ill-conditioned linear systems, to find efficient preconditioners rapidly, or to find robust approximate inverses, we suggest the following matrix multiplication-based iterative method:

$$\begin{aligned}
 V_{n+1} &= \frac{1}{16} V_n(120I + AV_n(-393I + AV_n(735I + AV_n(-861I \\
 &\quad + AV_n(651I + AV_n(-315I + AV_n(93I + AV_n(-15I + AV_n))))))),
 \end{aligned}
 \tag{3.1}$$

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

Theorem 3.1. Assume that $A = [a_{i,j}]_{N \times N}$ be an invertible matrix with real or complex elements. If the initial guess V_0 satisfies

$$\|I - AV_0\| < 1,
 \tag{3.2}$$

then the iteration (3.1) converges to A^{-1} with at least seventh convergence order.

Proof. In order to prove the convergence behavior of (3.1), we assume that $\|I - AV_0\| < 1$, $E_0 = I - AV_0$, and $E_n = I - AV_n$. We then have

$$\begin{aligned}
 E_{n+1} &= I - AV_{n+1} \\
 &= I - A \left[\frac{1}{16} V_n(120I + AV_n(-393I + AV_n(735I + AV_n(-861I + AV_n(651I + AV_n(-315I \right. \\
 &\quad \left. + AV_n(93I + AV_n(-15I + AV_n)))))) \right] \\
 &= I - A \left[\frac{1}{16} V_n \left(120I - 393AV_n + 735(AV_n)^2 - 861(AV_n)^3 + 651(AV_n)^4 - 315(AV_n)^5 \right) \right]
 \end{aligned}$$

$$\begin{aligned}
& +93(AV_n)^6 - 15(AV_n)^7 + (AV_n)^8 \Big] \\
& = -\frac{1}{16}(-4I + AV_n)^2(-I + AV_n)^7 \\
& = \frac{1}{16}(3I + I - AV_n)^2(I - AV_n)^7 \\
& = \frac{1}{16}(3I + E_n)^2(E_n)^7 \\
& = \frac{1}{16}(9I + 6E_n + E_n^2)(E_n)^7 \\
& = \frac{1}{16}(E_n^9 + 6E_n^8 + 9E_n^7).
\end{aligned} \tag{3.3}$$

Hence, it is now and according to the above simplifications easily to have

$$\|E_{n+1}\| \leq \frac{1}{16} \left(\|E_n^9 + 6E_n^8 + 9E_n^7\| \right) \leq \frac{1}{16} \left(\|E_n\|^9 + 6\|E_n\|^8 + 9\|E_n\|^7 \right). \tag{3.4}$$

Furthermore, since $\|E_0\| < 1$, and $\|E_1\| \leq \|E_0\|^7 < 1$, we get that

$$\|E_{n+1}\| \leq \|E_n\|^7 \leq \|E_{n-1}\|^{7^2} \leq \dots \leq \|E_0\|^{7^{n+1}} < 1, \tag{3.5}$$

where (3.5) tends to zero when $n \rightarrow \infty$. That is to say

$$I - AV_n \rightarrow 0, \tag{3.6}$$

when $n \rightarrow \infty$, and thus for (3.6), we obtain

$$V_n \rightarrow A^{-1}, \quad \text{as } n \rightarrow \infty. \tag{3.7}$$

We must now illustrate the seventh order of convergence for (3.1) due to the obtained theoretical discussions that (3.1) converges under the assumption made in Theorem 3.1 to A^{-1} . To do this aim, we denote $\varepsilon_n = V_n - A^{-1}$, as the error matrix in the iterative procedure (3.1). We have

$$I - AV_{n+1} = \frac{1}{16} \left((I - AV_n)^9 + 6(I - AV_n)^8 + 9(I - AV_n)^7 \right). \tag{3.8}$$

Equation (3.8) yields in

$$A(A^{-1} - V_{n+1}) = \frac{1}{16} \left(A^9 (A^{-1} - V_n)^9 + 6A^8 (A^{-1} - V_n)^8 + 9A^7 (A^{-1} - V_n)^7 \right), \quad (3.9)$$

$$A^{-1} - V_{n+1} = \frac{1}{16} \left(A^8 (A^{-1} - V_n)^9 + 6A^7 (A^{-1} - V_n)^8 + 9A^6 (A^{-1} - V_n)^7 \right). \quad (3.10)$$

Using (3.10), we attain

$$\varepsilon_{n+1} = \frac{1}{16} \left(A^8 (\varepsilon_n)^9 - 6A^7 (\varepsilon_n)^8 + 9A^6 (\varepsilon_n)^7 \right), \quad (3.11)$$

which simplifies by taking norm from both sides

$$\|\varepsilon_{n+1}\| \leq \frac{1}{16} \left[\|A^8 \varepsilon_n^9\| + \|6A^7 \varepsilon_n^8\| + \|9A^6 \varepsilon_n^7\| \right], \quad (3.12)$$

and consequently

$$\|\varepsilon_{n+1}\| \leq \left(\frac{1}{16} \left[9\|A\|^6 + 6\|A\|^7 \|\varepsilon_n\| + \|A\|^8 \|\varepsilon_n\|^2 \right] \right) \|\varepsilon_n\|^7. \quad (3.13)$$

This shows that the method (3.1) converges to A^{-1} with at least seventh order of convergence. This concludes the proof. \square

Remark 3.2. For the above Theorem 3.1, we can conclude some points as follows.

- (1) From (3.4), one knows that the condition (3.2) may be weakened. In fact, we only need that the spectral radius of AV_0 be less than one for the convergence of the above new method (3.1). In this case, the choice of V_0 may be obtained according to the estimate formulas for the spectral radius $\rho(AV_0)$ (see, e.g., [11])
- (2) In some experiments and to reduce the computational cost, we may solve the matrix multiplications, based on the vector and parallel processors.
- (3) Finally, for the choice of V_0 , there exist many of different forms. We will describe this problem after Theorem 3.3 based on some literatures.

We now give a property about the scheme (3.1). This property shows that $\{V_n\}_{n=0}^{n=\infty}$ of (3.1) may be applied to not only the left preconditioned linear system $V_n Ax = V_n b$ but also to the right preconditioned linear system $AV_n y = b$, where $y = V_n x$.

Theorem 3.3. Let again $A = [a_{i,j}]_{N \times N}$ be a nonsingular real or complex matrix. If

$$AV_0 = V_0 A, \quad (3.14)$$

is valid, then, for the sequence of $\{V_n\}_{n=0}^{n=\infty}$ of (3.1), one has that

$$AV_n = V_n A, \quad (3.15)$$

holds, for all $n = 1, 2, \dots$

Proof. The mathematical induction is taken into account herein. First, since $AV_0 = V_0A$, we have

$$\begin{aligned}
 AV_1 &= A \left(\frac{1}{16} V_0 (120I + AV_0(-393I + AV_0(735I + AV_0(-861I + AV_0(651I \right. \\
 &\quad \left. + AV_0(-315I + AV_0(93I + AV_0(-15I + AV_0)))))) \right) \\
 &= \frac{1}{16} V_0 A (120I + V_0 A (-393I + V_0 A (735I + V_0 A (-861I + V_0 A (651I \\
 &\quad + V_0 A (-315I + V_0 A (93I + V_0 A (-15I + V_0 A)))))) \tag{3.16} \\
 &= \frac{1}{16} V_0 (120I + V_0 A (-393I + V_0 A (735I + V_0 A (-861I + V_0 A (651I \\
 &\quad + V_0 A (-315I + V_0 A (93I + V_0 A (-15I + V_0 A)))))) A \\
 &= V_1 A.
 \end{aligned}$$

Equation (3.16) shows that when $n = 1$, (3.15) is true. At the moment, suppose that $AV_n = V_n A$ is true, and then a straightforward calculation using (3.16) shows that, for all $n \geq 1$,

$$\begin{aligned}
 AV_{n+1} &= A \left[\frac{1}{16} V_n (120I + AV_n(-393I + AV_n(735I + AV_n(-861I + AV_n(651I \right. \\
 &\quad \left. + AV_n(-315I + AV_n(93I + AV_n(-15I + AV_n)))))) \right] \\
 &= \frac{1}{16} V_n A (120I + V_n A (-393I + V_n A (735I + V_n A (-861I + V_n A (651I \\
 &\quad + V_n A (-315I + V_n A (93I + V_n A (-15I + V_n A)))))) \tag{3.17} \\
 &= V_{n+1} A.
 \end{aligned}$$

This concludes the proof. \square

Note that according to the literatures [7, 9, 12, 13] and to find an initial value V_0 to preserve the convergence order of such iterative methods, we need to fulfill at least the condition given in Remark 3.2. or Theorem 3.1. We list some ways for this purpose in what follows.

WAY 1 If a matrix is strictly diagonally dominant, then choose V_0 as $V_0 = \text{diag}(1/a_{11}, 1/a_{22}, \dots, 1/a_{nn})$, where a_{ii} are the diagonal elements of A .

WAY 2 For the matrix A , choose V_0 as $V_0 = A^T / (\|A\|_1 \|A\|_\infty)$, wherein T stands for transpose, $\|A\|_1 = \max_j \{\sum_{i=1}^n |a_{ij}|\}$ and $\|A\|_\infty = \max_i \{\sum_{j=1}^n |a_{ij}|\}$.

WAY 3 If the ways 1-2 fail, then use $V_0 = \alpha I$, where I is the identity matrix, and $\alpha \in \mathbb{R}$ should adaptively be determined such that $\|I - \alpha A\| < 1$.

4. Numerical Testing

In this section, experiments are presented to demonstrate the capability of the suggested method. For solving a square linear system of equations of the general form $Ax = b$, wherein $A \in \mathbb{C}^{N \times N}$, we can now propose the following efficient algorithm: $x_{n+1} = V_{n+1}b$ while $n = 0, 1, 2, \dots$.

The programming package MATHEMATICA 8, [14, 15], has been used in this section. For numerical comparisons, we have used the methods (2.1), (2.2), the sixth-order method of Krishnamurthy and Sen (2.4), and the new algorithm (3.1).

As noted in Section 1, such high order schemes are so much fruitful in providing robust preconditioners to the linear systems. In fact, we believe that only one full iteration of the scheme (3.1) for even large sparse systems and by defining *SparseArray[mat]* to reduce the computational load of matrix multiplications is enough to find acceptable preconditioners to the linear systems. Anyhow, using parallel computations with simple commands for this purpose in MATHEMATICA 8 may reduce the computational burden much more.

Test Problem 1

Consider the linear system $Ax = b$, wherein A is the large sparse complex matrix defined by

```
n = 1000;
A = SparseArray[{Band[{1, 120}] -> -2., Band[{950, 1}] -> 2. - I,
Band[{-700, 18}] -> 1., Band[{1, 1}] -> 23.,
Band[{1, 100}] -> 0.2, Band[{214, -124}] -> 1.,
Band[{6, 800}] -> 1.1}, {n, n}, 0.].
```

The dimension of this matrix is 1000 with complex elements in its structure. The right hand side vector is considered to be $b = (1, 1, \dots, 1)^T$. Although page limitations do not allow us to provide the full form of such matrices, the structure of such matrices can easily be drawn. Figure 1 illustrates the plot and also the array plot of this matrix.

Now, we expect to find robust approximate inverses for A in less iterations by the high-order iterative methods. Furthermore, as we described in the previous sections, the approximate inverses could be considered for left or right preconditioned systems.

Table 1 clearly shows the efficiency of the proposed iterative method (3.1) in finding approximate inverses by manifesting the number of iterations and the obtained residual norm. When working with sparse matrices, an important factor which affects clearly on the computational cost of the iterative method is the number of nonzero elements. The considered sparse complex matrix A , in test problem 1, has 3858 nonzero elements at the beginning. We list the number of nonzero elements for different iteration matrix multiplication-based schemes in Table 1. It is obvious that the new scheme is much better than (2.1) and (2.2), but when comparing to (2.4) its number of nonzero elements are higher; however, this is completely satisfactory due to the better numerical results we have obtained for the residual norm. In fact, if one let the (2.4) to cycle more in order to reach the residual norm which is correct up to seven decimal places, then the obtained nonzero elements will be more than the corresponding one of (3.1).

At this time, if the user is satisfied of the obtained residual norm for solving the large sparse linear system then can be stopped, else, one can use the attained approximate inverse as a left or right preconditioner and solve the resulting preconditioned linear system with

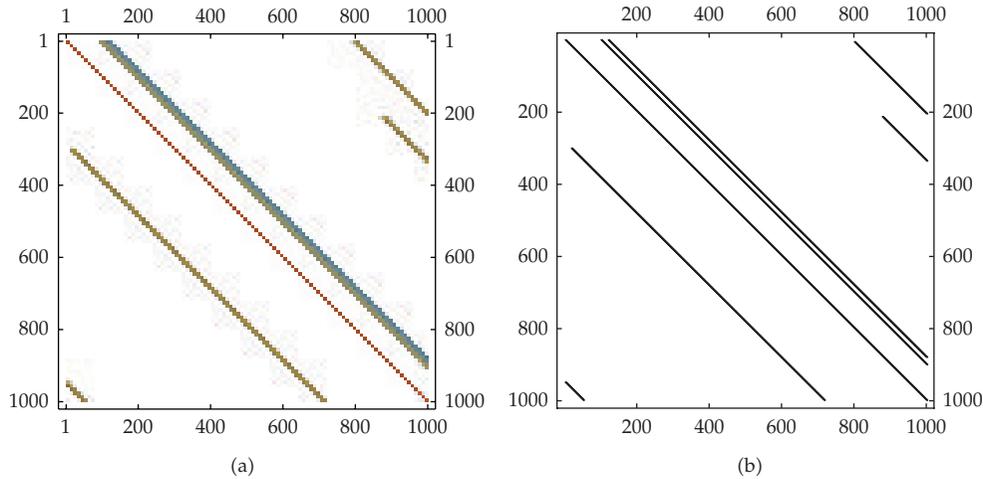


Figure 1: The matrix plot (a) and the array plot (b) of the coefficient complex matrix in test problem 1.

Table 1: Results of comparisons for the test problem 1.

Iterative methods	(2.1)	(2.2)	(2.4)	(3.1)
Number of iterations	3	2	1	1
Number of nonzero elements	126035	137616	65818	119792
Residual norm	3.006×10^{-7}	2.628×10^{-7}	1.428×10^{-5}	9.077×10^{-7}

low condition number using the *LinearSolve* command. Note that the written code in order to implement the test problem 1 for the method (3.1) is given as follows:

```

b = SparseArray[Table[1, {k, n}]]; DA = Diagonal[A];
B = SparseArray[Table[(1/DA[[i]]), {i, 1, n}]];
Id = SparseArray[{{i_, i_} -> 1}, {n, n}];
X = SparseArray[DiagonalMatrix[B]];
Do[X1 = SparseArray[X]; AV = Chop[SparseArray[A.X1]];
X = Chop[(1/16) X1.SparseArray[(120 Id + AV.(-393 Id
+ AV.(735 Id + AV.(-861 Id + AV.(651 Id + AV.(-315 Id +
AV.(93 Id + AV.(-15 Id + AV)))))))]];
Print[X]; L[i] = Norm[b - SparseArray[A.(X.b)]];
Print["The residual norm of the linear system solution is:"
Column[{i}, Frame -> All, FrameStyle -> Directive[Blue]]
Column[{L[i]}, Frame -> All, FrameStyle -> Directive[Blue]]];
, {i, 1}].

```

In what follows, we try to examine the matrix inverse-finding iterative methods of this paper on a dense matrix which is of importance in applications.

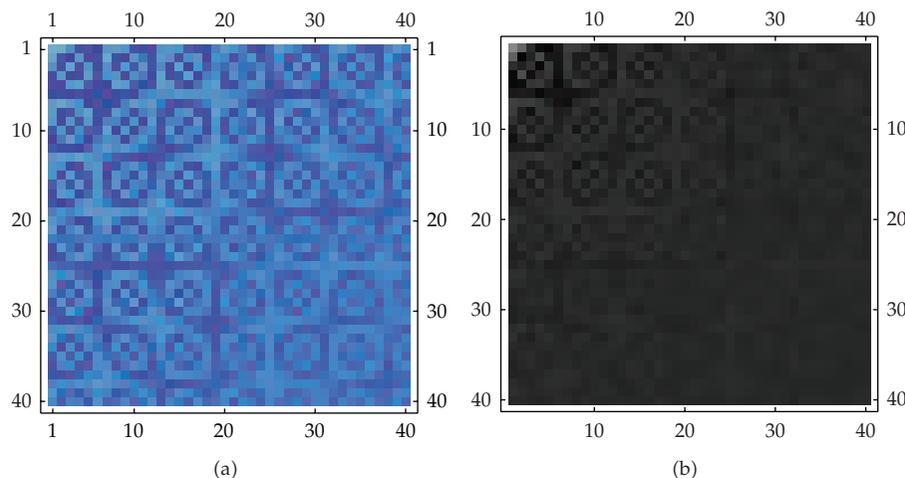


Figure 2: The matrix plot (a) and the array plot (b) of the coefficient complex matrix in test problem 2.

Table 2: Results of comparisons for the test problem 2 with condition number 18137.2.

Iterative methods	(2.1)	(2.2)	(2.4)	(3.1)
Number of iterations	29	18	11	10
The condition number of $V_n A$	1.00135	1.01234	1.01780	1.00114
Residual norm	6.477×10^{-7}	5.916×10^{-6}	8.517×10^{-6}	5.482×10^{-7}

Test Problem 2

Consider the linear system $Ax = b$, wherein A is the dense matrix defined by

$$n = 40;$$

$$A = \text{Table}[\text{Sin}[(x*y)]/(x + y) - 1., \{x, n\}, \{y, n\}].$$

The right hand side vector is again considered to be $b = (1, 1, \dots, 1)^T$. Figure 2 illustrates the plot and array plot of this matrix. Due to the fact that, for dense matrices, such plots will not be illustrative, we have drawn the 3-D plots of the coefficient dense matrix using the zero-order interpolation in test problem 2, alongside its approximate inverse obtained from the method (3.1) after 11 iterations in Figure 3. We further expect to have a similar identity-like behavior for the multiplication of A to its approximate inverse, and this is clear in part (c) of Figure 3.

Table 2 shows the number of iterations and the obtained residual norm for different methods in order to reveal the efficiency of the proposed iteration. There is a clear reduction in computational steps for the proposed method (3.1) in contrast to the other existing well-known methods of various orders in the literature. Table 2 further confirms the use of such iterative inverse finders in order to find robust approximate inverse preconditioners again. Since, the condition number of the coefficient matrix in Test Problem 2 is 18137.2, which is quite high for a 40×40 matrix. But as can be furnished in Table 2, the obtained condition number of the preconditioned matrix after the specified number of iterations is very small.

We should here note that the computational time requiring for implementing all the methods in Tables 1 and 2 for the Test Problems 1 and 2 is less than one second, and due

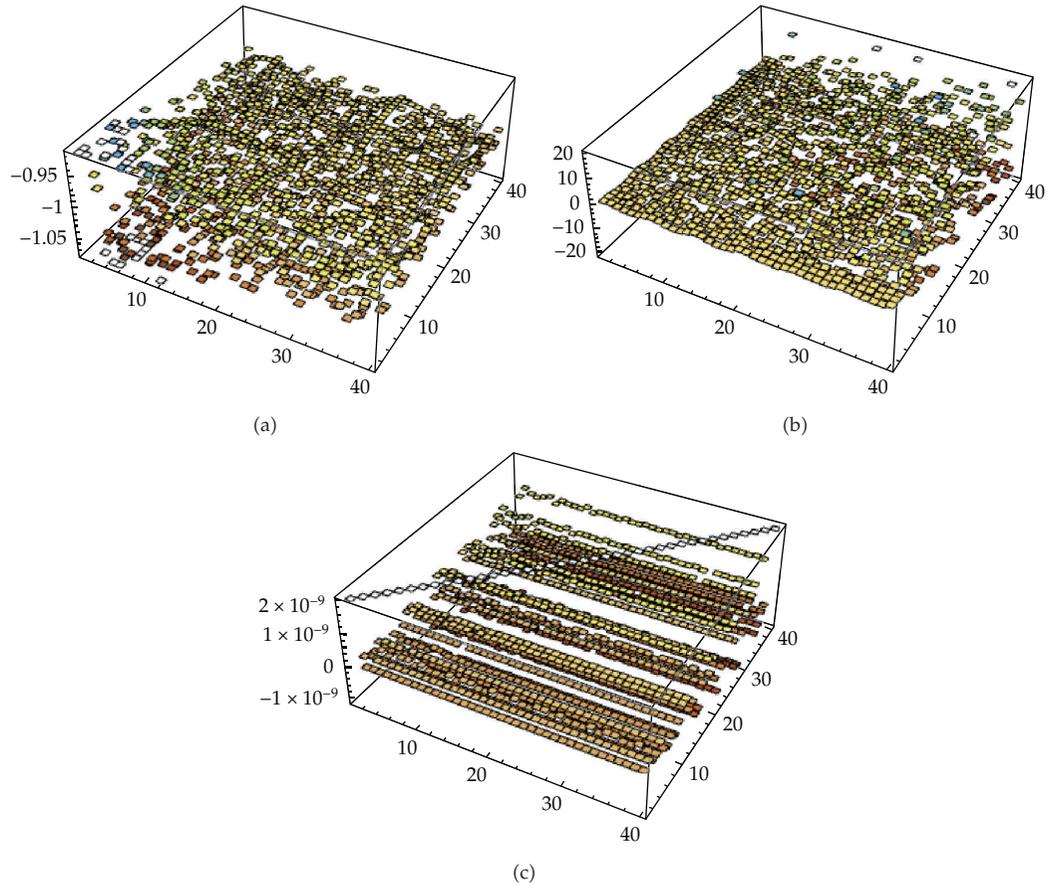


Figure 3: The 3D plot of the structure of the coefficient matrix (a) in the Test Problem 2, its approximate inverse (b) by the method (3.1) and their multiplication to obtain a similar output to the identity matrix (c).

to this we have not listed them herein. For the second test problem, we have used $V_0 = A^T / (\|A\|_1 \|A\|_\infty)$

5. Concluding Remarks

In the present paper, the author along with the helpful suggestions of the reviewers has developed an iterative method in inverse finding of matrices. Note that such high order-iterative methods are so efficient for very ill-conditioned linear systems or to find robust approximate inverse preconditioners. We have shown analytically that the suggested method (3.1), reaches the seventh order of convergence. Moreover, the efficacy of the new scheme was illustrated numerically in Section 4 by applying to a sparse matrix and an ill-conditioned dense matrix. All the numerical results confirm the theoretical aspects and show the efficiency of (3.1).

Acknowledgments

The author would like to take this opportunity to sincerely acknowledge many valuable suggestions made by the two anonymous reviewers and especially the third suggestion of reviewer 1, which have been made to substantially improve the quality of this paper.

References

- [1] T. Sauer, *Numerical Analysis*, Pearson, 2nd edition, 2011.
- [2] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, 2nd edition, 2002.
- [3] D. K. Salkuyeh and H. Roohani, "On the relation between the AINV and the FAPINV algorithms," *International Journal of Mathematics and Mathematical Sciences*, vol. 2009, Article ID 179481, 6 pages, 2009.
- [4] M. Benzi and M. Tuma, "Numerical experiments with two approximate inverse preconditioners," *BIT*, vol. 38, no. 2, pp. 234–241, 1998.
- [5] E. Chow and Y. Saad, "Approximate inverse preconditioners via sparse-sparse iterations," *SIAM Journal on Scientific Computing*, vol. 19, no. 3, pp. 995–1023, 1998.
- [6] G. Schulz, "Iterative berechnung der reziproken matrix," *Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, pp. 57–59, 1933.
- [7] H.-B. Li, T.-Z. Huang, Y. Zhang, V.-P. Liu, and T.-V. Gu, "Chebyshev-type methods and preconditioning techniques," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 260–270, 2011.
- [8] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms—Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, India, 1986.
- [9] G. Codevico, V. Y. Pan, and M. V. Barel, "Newton-like iteration based on a cubic polynomial for structured matrices," *Numerical Algorithms*, vol. 36, no. 4, pp. 365–380, 2004.
- [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," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3433–3442, 2010.
- [11] H. B. Li, T. Z. Huang, S. Shen, and H. Li, "A new upper bound for spectral radius of block iterative matrices," *Journal of Computational Information Systems*, vol. 1, no. 3, pp. 595–599, 2005.
- [12] 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. 1109–1130, 1991.
- [13] K. Moriya and T. Nodera, "A new scheme of computing the approximate inverse preconditioner for the reduced linear systems," *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 345–352, 2007.
- [14] S. Wagon, *Mathematica in Action*, Springer, 3rd edition, 2010.
- [15] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.



Hindawi

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

