

Research Article

A New Iterative Method for Finding Approximate Inverses of Complex Matrices

M. Kafei Razavi,¹ A. Kerayechian,¹ M. Gachpazan,¹ and S. Shateyi²

¹ Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

² Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 24 May 2014; Revised 25 July 2014; Accepted 28 July 2014; Published 14 September 2014

Academic Editor: Juan R. Torregrosa

Copyright © 2014 M. Kafei Razavi et al. 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.

This paper presents a new iterative method for computing the approximate inverse of nonsingular matrices. The analytical discussion of the method is included to demonstrate its convergence behavior. As a matter of fact, it is proven that the suggested scheme possesses tenth order of convergence. Finally, its performance is illustrated by numerical examples on different matrices.

1. Introduction

The solution of linear algebraic systems of the form $Ax = b$, where $A = [a_{i,j}]$ is an $m \times m$ matrix (all matrices in this paper are of the same dimension, unless it is clearly stated) and b is a given vector, is the center to many numerical simulations and is often the most time-consuming part of a computation. Direct methods, which work essentially based on finding the inverse of the coefficients matrix A , are very robust, and they tend to require a predictable amount of resources in terms of time and storage. Their only problem, in fact, lies in the massive need of time and memory in calculations, which normally put them out of interest especially for the cases when A is sparse.

At this time, iterative methods can be taken into account. The relaxation and Krylov subspace methods are of such type, which are reliable in solving large scale sparse systems [1]. On the other hand, we have another type of iterative methods, which are called Schulz-type iterations in the literature (see, e.g., [2]). These techniques are based on finding robust approximate inverses of a given matrix.

The oldest technique of this type is the Schulz method [3] defined by

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

where I is the identity matrix. In the general case, it is known to converge with $V_0 := \alpha A^*$, where $0 < \alpha < 2/\rho(A^*A)$ and $\rho(\cdot)$ denotes the spectral radius. Such schemes are also useful for sensitivity analysis when accurate approximate inverses are needed for both square and rectangular matrices. Notice that, for rectangular matrices, one may obtain their generalized inverse using such iterative methods [4].

These solvers are also the method of choice in certain areas such as circuits, power system networks, and chemical plant modeling [5]. A practical application of Schulz-type methods, which recently attracted some numerical analysts to itself [6], is in preconditioning. In fact, by having an initial value [7], one is able to produce any approximate inverse preconditioners up to the desired accuracy and then solve the preconditioned linear system as rapidly as possible.

Hence, we are interested in finding a new iterative method belonging to the class of Schulz-type methods for finding approximate inverses in this work.

Remark 1. We use matrix norm 2 in all subsequent derivations and discussions unless the other forms are clearly stated.

The rest of this paper is organized in what follows. In the next section, we briefly review some of the existing Schulz-type methods and provide a new mathematical proof for one of the unproved iterative methods. Another contribution

of this paper is presented in Section 3. That section is also devoted to the analysis of convergence. Section 4 covers the numerical simulations and some notes in order to have the best feedback in practical implementations. And finally, concluding remarks are drawn in Section 5.

2. Preliminaries

Li et al. in [6] presented

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

and also proposed another third-order iterative method for finding A^{-1} as comes next (right-product form):

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

W. Li and Z. Li in [8] proposed the following fourth-order iteration method:

$$V_{n+1} = V_n(4I - 6AV_n + 4(AV_n)^2 - (AV_n)^3), \quad (4)$$

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

In fact, a family of methods developed in [8]. We draw attention to the point that the iterative methods (2) and (4) can also be found in the textbook [9]. Moreover, we suggest that the iterative method (3) can be rewritten as (left-product form)

$$V_{n+1} = \frac{1}{4}V_n(13I - AV_n(15I - AV_n(7I - AV_n))). \quad (5)$$

This structure is easier due to the use of Horner-like multiplications and, subsequently, lowers round-off errors by avoiding the calculation of matrix power, which is costly. It should be remarked that authors in [6] did not provide a mathematical proof for (3) or its other form (5). Thus, herein we prove its order of convergence to first complete the paper [6].

Theorem 2. Assume that $A = [a_{i,j}]_{m \times m}$ is an invertible matrix with real or complex components. If the initial value V_0 satisfies

$$\|I - AV_0\| < 1, \quad (6)$$

then, the iteration (5) converges cubically to A^{-1} .

Proof. In order to prove the convergence of (5), we consider first that $\|I - AV_0\| < 1$, $E_0 = I - AV_0$, and $E_n = I - AV_n$. For (5), we get that

$$\begin{aligned} E_{n+1} &= I - AV_{n+1} \\ &= I - A \left[\frac{1}{4}V_n(13I - AV_n(15I - AV_n(7I - AV_n))) \right] \\ &= I - A \left[-\frac{1}{4}V_n(-13I + 15AV_n - 7(AV_n)^2 + (AV_n)^3) \right] \end{aligned}$$

$$\begin{aligned} &= \frac{1}{4}(4I - AV_n)(I - AV_n)^3 \\ &= \frac{1}{4}(3I + I - AV_n)(I - AV_n)^3 \\ &= \frac{1}{4}(3I + E_n)(E_n)^3 = \frac{1}{4}(3E_n^3 + E_n^4). \end{aligned} \quad (7)$$

Thus, we obtain $\|E_{n+1}\| = (1/4)(\|3E_n^3 + E_n^4\|) \leq (1/4)(3\|E_n^3\| + \|E_n^4\|)$. Moreover, since $\|E_0\| < 1$ and $\|E_1\| \leq \|E_0\|^3 < 1$, we get that

$$\|E_{n+1}\| \leq \|E_n\|^3 \leq \|E_{n-1}\|^{3^2} \leq \dots \leq \|E_0\|^{3^{n+1}} < 1, \quad (8)$$

where (8) tends to zero when $n \rightarrow \infty$. That is, $I - AV_n \rightarrow 0$, when $n \rightarrow \infty$, and thus for (5), we attain $V_n \rightarrow A^{-1}$, as $n \rightarrow \infty$. This shows the convergence.

Now, we must show its third order. Toward this end, we denote by $\varepsilon_n = A^{-1} - V_n$ the error matrix in the iterative procedure (5). We have (using a similar methodology as in [10])

$$I - AV_{n+1} = \frac{1}{4} [3(I - AV_n)^3 + (I - AV_n)^4]. \quad (9)$$

We can now obtain

$$\begin{aligned} &A(A^{-1} - V_{n+1}) \\ &= \frac{1}{4} [3(A(A^{-1} - V_n))^3 + (A(A^{-1} - V_n))^4], \end{aligned} \quad (10)$$

$$A\varepsilon_{n+1} = \frac{1}{4} [3(A\varepsilon_n)^3 + (A\varepsilon_n)^4]. \quad (11)$$

Equation (11) results in

$$A\varepsilon_{n+1} = \frac{1}{4} [3(A\varepsilon_n)(A\varepsilon_n)^2 + (A\varepsilon_n)(A\varepsilon_n)^3], \quad (12)$$

and subsequently

$$\varepsilon_{n+1} = \frac{1}{4} [3\varepsilon_n(A\varepsilon_n)^2 + \varepsilon_n(A\varepsilon_n)^3], \quad (13)$$

which yields by taking norm from both sides

$$\|\varepsilon_{n+1}\| \leq \frac{1}{4} [3\|\varepsilon_n\| \|A\varepsilon_n\|^2 + \|\varepsilon_n\| \|A\varepsilon_n\|^3], \quad (14)$$

and consequently

$$\|\varepsilon_{n+1}\| \leq \left(\frac{1}{4} [3\|A\|^2 + \|A\|^3 \|\varepsilon_n\|] \right) \|\varepsilon_n\|^3. \quad (15)$$

This reveals that the iterative method (5) converges to A^{-1} with at least third order of convergence. The proof is complete. \square

3. A Novel Method

Let I be the identity matrix and $n = 0, 1, 2, \dots$. We aim at constructing an iterative method in which the sequence of iterates $\{V_n\}_{n=0}^{\infty}$ converges to A^{-1} for an appropriate initial guess. We suggest our proposed method as follows:

$$\begin{aligned} \zeta_n &= -11I + \psi_n (25I + \psi_n \\ &\quad \times (-30I + \psi_n (20I + \psi_n (-7I + \psi_n))))), \\ V_{n+1} &= -\frac{1}{4}V_n \zeta_n (4I + \psi_n \zeta_n), \quad n = 0, 1, 2, \dots, \end{aligned} \tag{16}$$

while $\psi_n = AV_n$ and its derivation will be pointed out in Section 4. In numerical mathematics, it is essential to know the theoretical behavior of an approximate method. In what follows, we prove the convergence order of (16).

Theorem 3. Assume that $A = [a_{i,j}]_{m \times m}$ is an invertible matrix with real or complex entries. If the initial guess V_0 satisfies

$$\|I - AV_0\| < 1, \tag{17}$$

then, the iteration (16) converges to A^{-1} with at least tenth convergence order.

Proof. In order to prove the convergence of (16), we consider the same assumptions as we did in the proof of Theorem 2. We then have

$$\begin{aligned} E_{n+1} &= I - AV_{n+1} \\ &= I - A \left[-\frac{1}{4}V_n \right. \\ &\quad \times (-11I + AV_n \\ &\quad \times (25I + AV_n (-30I + AV_n \\ &\quad \times (20I + AV_n (-7I + AV_n)))))) \\ &\quad \times (4I + AV_n \\ &\quad \times (-11I + AV_n \\ &\quad \times (25I + AV_n \\ &\quad \times (-30I + AV_n (20I + AV_n \\ &\quad \times (-7I + AV_n)))))) \left. \right] \end{aligned}$$

$$\begin{aligned} &= I - A \left[-\frac{1}{4}V_n (-11I + 25AV_n - 30(AV_n)^2 \right. \\ &\quad \left. + 20(AV_n)^3 - 7(AV_n)^4 + (AV_n)^5) \right. \\ &\quad \times (4I - 11AV_n + 25(AV_n)^2 - 30(AV_n)^3 \\ &\quad \left. + 20(AV_n)^4 - 7(AV_n)^5 + (AV_n)^6) \right] \\ &= \frac{1}{4}(-2I + AV_n)^2(-I + AV_n)^{10} \\ &= \frac{1}{4}(I + I - AV_n)^2(I - AV_n)^{10} = \frac{1}{4}(I + E_n)^2(E_n)^{10} \\ &= \frac{1}{4}(I + 2E_n + E_n^2)E_n^{10} = \frac{1}{4}(E_n^{10} + 2E_n^{11} + E_n^{12}). \end{aligned} \tag{18}$$

Thus, we obtain

$$\begin{aligned} \|E_{n+1}\| &= \frac{1}{4}(\|E_n^{10} + 2E_n^{11} + E_n^{12}\|) \\ &\leq \frac{1}{4}(\|E_n\|^{10} + 2\|E_n\|^{11} + \|E_n\|^{12}). \end{aligned} \tag{19}$$

Moreover, since $\|E_0\| < 1$ and $\|E_1\| \leq \|E_0\|^{10} < 1$, we attain

$$\|E_{n+1}\| \leq \|E_n\|^{10} \leq \|E_{n-1}\|^{10^2} \leq \dots \leq \|E_0\|^{10^{n+1}} < 1, \tag{20}$$

where (20) tends to zero when $n \rightarrow \infty$. That is, $I - AV_n \rightarrow 0$, when $n \rightarrow \infty$, and thus for (16), we obtain

$$V_n \rightarrow A^{-1}, \quad \text{as } n \rightarrow \infty. \tag{21}$$

We must now illustrate the tenth order of convergence for (16). To this aim, we denote by $\varepsilon_n = A^{-1} - V_n$ the error matrix in the iterative procedure (16). We have (using (18))

$$I - AV_{n+1} = \frac{1}{4}[(I - AV_n)^{10} + 2(I - AV_n)^{11} + (I - AV_n)^{12}]. \tag{22}$$

Equation (22) yields

$$\begin{aligned} &A(A^{-1} - V_{n+1}) \\ &= \frac{1}{4}[(A(A^{-1} - V_n))^{10} + 2(A(A^{-1} - V_n))^{11} \\ &\quad + (A(A^{-1} - V_n))^{12}], \end{aligned} \tag{23}$$

$$A\varepsilon_{n+1} = \frac{1}{4}[(A\varepsilon_n)^{10} + 2(A\varepsilon_n)^{11} + (A\varepsilon_n)^{12}]. \tag{24}$$

Using (24), we attain

$$\begin{aligned} A\varepsilon_{n+1} &= \frac{1}{4}[(A\varepsilon_n)(A\varepsilon_n)^9 + 2(A\varepsilon_n)(A\varepsilon_n)^{10} \\ &\quad + (A\varepsilon_n)(A\varepsilon_n)^{11}], \end{aligned} \tag{25}$$

$$\varepsilon_{n+1} = \frac{1}{4}[\varepsilon_n(A\varepsilon_n)^9 + 2\varepsilon_n(A\varepsilon_n)^{10} + \varepsilon_n(A\varepsilon_n)^{11}],$$

which simplifies, by taking norm from both sides, the following:

$$\|\varepsilon_{n+1}\| \leq \frac{1}{4} \left[\|\varepsilon_n\| \|A\varepsilon_n\|^9 + 2\|\varepsilon_n\| \|A\varepsilon_n\|^{10} + \|\varepsilon_n\| \|A\varepsilon_n\|^{11} \right], \quad (26)$$

and consequently

$$\|\varepsilon_{n+1}\| \leq \left(\frac{1}{4} \left[\|A\|^9 + 2\|A\|^{10} \|\varepsilon_n\| + \|A\|^{11} \|\varepsilon_n\|^2 \right] \right) \|\varepsilon_n\|^{10}. \quad (27)$$

This shows that the method (16) converges to A^{-1} with at least tenth order of convergence. The proof is now complete. \square

A simple corollary from Theorems 2 and 3 is that, using the same conditions and initial conditions, the higher order methods arrive at the convergence phase faster than lower order methods and this reduces the number of iterations.

Note that the sequence $\{V_n\}_{n=0}^{\infty}$ of (16) may be applied to not only the left preconditioned linear system $V_n Ax = V_n b$ but also the right preconditioned linear system $AV_n y = b$, where $y = V_n x$, only if the initial matrix satisfies $AV_0 = V_0 A$.

The iterative methods that have been discussed up to now are sensitive for choosing the initial guess 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 Theorems 2 and 3.

Thus, in order to preserve the convergence order, we remind the reader that the efficient way of producing V_0 as given in [11] is as follows: $V_0 = A^T / (\|A\|_1 \|A\|_\infty)$. Another adaptive way is $V_0 = \rho I$, where I is the identity matrix, and $\rho \in \mathbb{R}$ should be determined so that $\|I - \rho A\| < 1$.

Remark 4. The new iteration (16) reaches 10th order by using 8 matrix-matrix multiplications, while the schemes (1), (2), and (5) reach 2nd, 3rd, and 4th orders, respectively, by consuming 2, 3, and 4 matrix-matrix multiplications. Hence, the contributed method has less computational cost than its competitors. This superiority will be clarified in Section 4. It should also be remarked that the convergence of any order for nonsingular square matrices is generated in Section 6 of Chapter 2 of the book [12], whereas the general way for the rectangular matrices is discussed in Chapter 5 of [9] and the recent paper [13]. In fact, in those constructions a convergence order ρ will always be attained by ρ times of matrix-matrix products, such as (1) which reaches the order 2 using two matrix-matrix multiplications.

Remark 5. Two important matters must be mentioned at this moment to ease up the perception of why a higher order (efficient) method such as (16) with 8 matrix-matrix products to reach at least the convergence order 10 is practical. First, by following the comparative index of informational efficiency index of inverse-finders [14], defined by $E = \rho/\theta$, wherein ρ and θ stand for the convergence order and the number of matrix-matrix products, the informational efficiency for (16),

that is, $10/8 \approx 1.25$, beats its other competitive, $2/2 = 1$ of (1), $3/3 = 1$ of (2) and (4), and $3/4 = 0.75$ of (3). And second, the significance of the new scheme will be displayed in the implementation of such schemes. To illustrate further, such iterations are totally dependent on the initial matrices. Though there are certain and efficient ways for finding V_0 , in general such initial approximations take a high number of iterations to arrive at the convergence phase. On the other hand, each cycle of the implementation of such Schulz-type methods includes one stopping criterion based on the use of a matrix norm, and this would impose further burden and load in general for the low order methods in contrast to the high order methods such as (5). Because the computation of a matrix norm (usually $\|\cdot\|_2$ for dense matrices and $\|\cdot\|_F$ for large sparse matrices) takes a reasonable time, therefore higher number of steps/iterations (which is the result of lower order methods) will be costlier than the lower number of steps/iterations.

Remark 6. The index that we are defining is different from the classical efficiency index as defined in [15]. Traub in [15] discussed why informational efficiency index is needed. In fact, the kind of efficiency index that the users apply depends on the situation which is dealt with. Note that the cost of each iteration (step) is governed by the number of matrix-matrix products, order, and computing of a stopping criterion. Thus, the informational index has meaning in this case, because it measures the gain brought each time a matrix-matrix product along with the order and the stopping criterion is computed.

4. Numerical Reports

In this section, some experiments are presented to demonstrate the capability of the suggested method. The computer algebra system MATHEMATICA 8, [16] and [17], has been used in this section. For numerical comparisons, we have used the methods (1), (2), (5), and (16). We will also use double precision in our calculations. The computer specifications are Microsoft Windows XP Intel(R), Pentium(R) 4, CPU 3.20 GHz, and 4 GB of RAM.

Experiment 1. In this test, 10 sparse random complex matrices of the dimension 2500 are considered as follows:

```
n = 2500; number = 10;
Table[A[j] = SparseArray[{Band[{-100, 700}]
->RandomReal[], {i_., i_} :> 3.3,
Band[{500, -150}, {n - 20, n - 25}]
->{RandomReal[], 10. + I},
Band[{600, 250}, {n - 100, n - 400}]
->{RandomReal[10],
RandomReal[10]}], {n, n}, 0.], {j, number}];
```

In this test, the stopping criterion is $\|I - V_n A\|_1 \leq 10^{-6}$, and the maximum number of iterations allowed is set to 100. Note that in this test the initial choice has been constructed by $V_0 = A^T / (\|A\|_1 \|A\|_\infty)$. We also plot the condition number of the 10 test matrices in Figure 1.

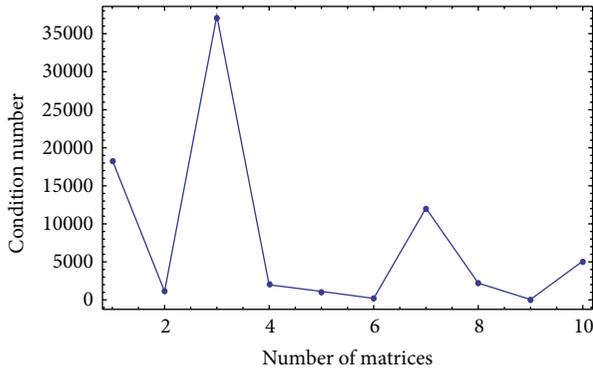


FIGURE 1: Comparison of condition numbers for the 10 test sparse matrices in Experiment 1.

The results of comparisons for the test problem have been presented in Figures 2-3. We have distinguished the curves by various symbols like circle, triangle, and so forth, alongside different colors. The attained results reverify the robustness of the proposed iterative method (16) by a clear reduction in the number of iterations and the elapsed time. Note that, in figures of this paper, Schulz, KSM, MM, and proposed method (PM) stand for (1), (2), (5), and (16), respectively.

In general, iterative Schulz-type methods are very useful for large sparse matrices having a sparse inverse or when only an approximate inverse is needed.

After a few iterations of such methods, the computed approximate inverse may be dense, and thus the whole procedure might be slow. To remedy this, a threshold can be imposed to the implemented algorithm. Hopefully, this can be done by the command Chop[exp, tol]. In Experiment 1, we have set the tol to 10^{-10} . This technique of implementation would be also fruitful for preconditioning.

We also point out that the construction of (16) is based on applying the nonlinear equation solver (similar idea to [3])

$$\begin{aligned}
 y_n &= v_n - f'(v_n)^{-1} f(v_n), \\
 z_n &= v_n - (2^{-1} f(v_n)^{-1}) (f(v_n)^{-1} + f(y_n)^{-1}), \\
 w_n &= z_n - (f[z_n, y_n])^{-1} f(z_n), \\
 v_{n+1} &= w_n - f(w_n)^{-1} f(w_n), \quad n = 0, 1, 2, \dots,
 \end{aligned}
 \tag{28}$$

on the matrix equation $F(V) = V^{-1} - A = 0$, where, for example, $f[z_n, y_n] = (z_n - y_n)^{-1} (f(z_n) - f(y_n))$ is the two-point divided difference, which gives us the aimed tenth-order method (16).

Remark 7. As discussed before, an important application of such solvers is to provide robust preconditioners for solving linear systems of equations. To illustrate further, we have chosen the following example carefully, in which, in a practical problem, the GMRES solver fails in solving the resulting system of a discretization, when the maximum number of steps allowed to the GMRES is 2000.

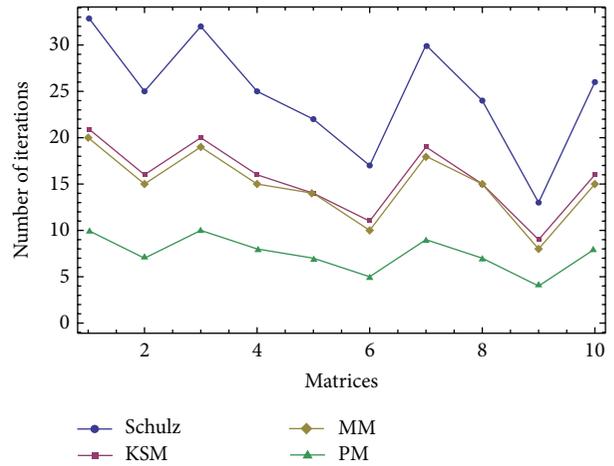


FIGURE 2: Comparison of the number of iterations for the 10 test sparse matrices in Experiment 1.

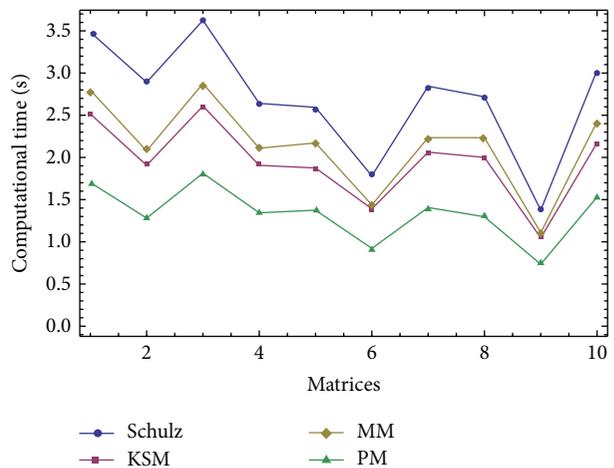


FIGURE 3: Comparison of the computational times for the 10 test sparse matrices in Experiment 1.

Experiment 2. We consider solving a boundary value problem (BVP) using finite difference discretization. Let us solve the BVP

$$\frac{\partial^2 u}{\partial x^2} + f(x)u = g(x), \quad u(0) = 0, \quad u'(1) = 0, \tag{29}$$

where $f(x) = 1 + 100 \exp(-(321(x - 1/2))^2)$, $g(x) = \sin(\pi x)$, and the number of discretization in our range $[0, 1]$ is $n = 1000$.

The implementation to find the solution with the tolerance 10^{-6} would be (in the MATHEMATICA environment) as comes next:

```

n = 1000; h = 1./n; xgrid = h Range[1, n];
f[x_] := 1 + 100 Exp[-(321(x - 1/2)) ^ 2];
g[x_] := Sin[Pi x];
id = SparseArray[{i_, i_} -> 1., {n, n}, 0.];
d2 = SparseArray[{{i_, i_} -> -2., {n, n - 1} ->
2., {i_, j_}; Abs[i - j] == 1 -> 1.}, {n, n}, 0.]
    
```

TABLE 1: Comparison of the computational time in solving the linear system $Ax = b$ in Experiment 2.

| Methods | GMRES | (1)-PGMRES | (2)-PGMRES | (5)-PGMRES | (16)-PGMRES |
|----------------|-------|------------|------------|------------|-------------|
| The whole time | Fail | 9.21 | 7.31 | 4.76 | 3.15 |

Now the coefficient matrix and the right hand side vector for finding the solution of the BVP (29) in the sparse form can be deduced by

```
A = SparseArray[d2/h ^ 2 + id f[xgrid]]
b = g[xgrid];
```

Just like the other problems, this is the most time-consuming step, that is, to solve the sparse system $Ax = b$. Herein, we choose the Krylov subspace method GMRES to solve the system, without preconditioning and with V_5 produced from the iterative method (1), V_4 produced from the iterative methods (2) and (5), and V_2 produced from the iterative methods (16) to solve the left preconditioned system $V_iAx = V_ib$.

In this way, the system will be well behaved and we expect to find the solution of the BVP (29) in less computational time than the nonpreconditioned GMRES. The results of consuming time for this purpose are given in Table 1. We should remark that in this test $V_0 = \text{diag}(1/a_{11}, 1/a_{22}, \dots, 1/a_{mm})$ has been chosen as the initial approximation based on [18], where a_{ii} is the i th diagonal entry of A .

As is obvious from Table 1, the nonpreconditioned GMRES fails to converge even after 2000 iteration steps, while the required accuracy could be attained using preconditioning. The best time which beats all the other ways comes from the new scheme (16). The left preconditioned system resulting from (16), showed as (16)-PGMRES, is reliable in solving linear ill-conditioned systems.

Note that the time reported for PGMRES is the whole time of constructing the initial matrix, obtaining the preconditioner V_i and solving the left preconditioned system $V_iAx = V_ib$, by GMRES.

Experiment 3. In order to compare the preconditioners obtained from new method with the famous preconditioners of the literature resulting from incomplete LU factorizations, [1], we pay heed to solving the linear sparse systems $Ax = b$, of the dimension 841 using BiCGSTAB. The matrix A has been chosen from MatrixMarket database as $A = \text{ExampleData}["\text{Matrix}", "YOUNG1C"]$, while the right hand side vector is $b = (1, 1, \dots, 1)^T$. The solution would be $(-0.0177027 - 0.00693171I, \dots, -0.0228083 - 0.00589176I)^T$. Figure 4 denotes the plot of matrix A .

The left preconditioned system using V_3 of Schulz, V_2 of KSM, and V_1 of the proposed method PM along with the well-known preconditioned techniques ILU0, ILUT, and ILUTP has been tested, while the initial vector has been chosen for all the cases automatically by the command of `LinearSolve[]` in MATHEMATICA 8. The results of consuming time comparisons for different tolerances (residual norms) have been listed in Figure 5.

Note that, after a few iterations, the computed preconditioner of Schulz-type methods may be dense. Like before, we

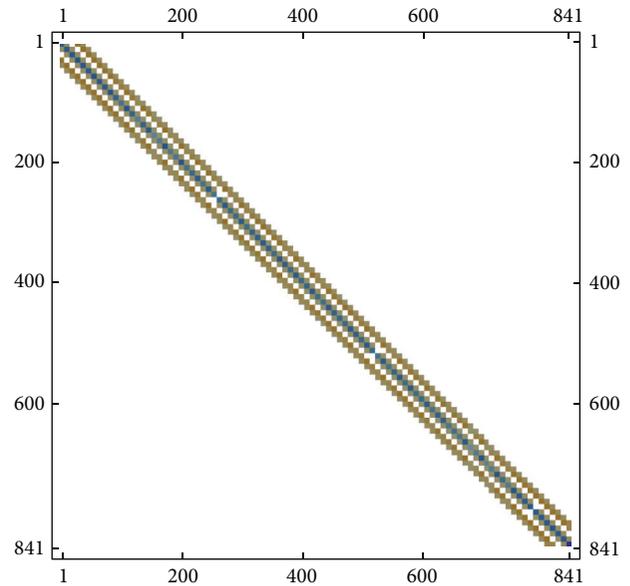


FIGURE 4: The plot of matrix A in Experiment 3.

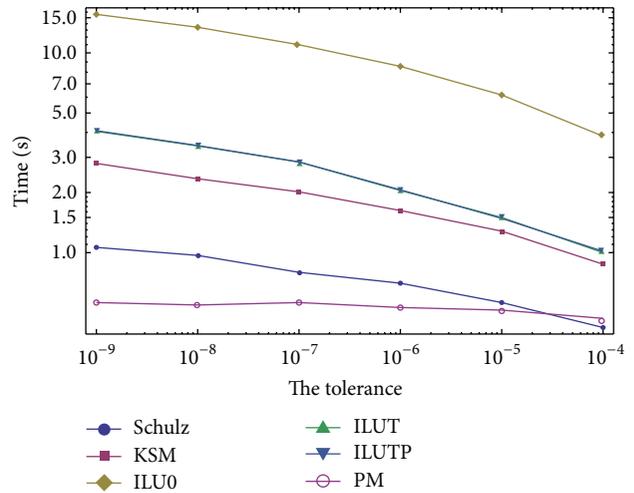


FIGURE 5: Comparison of the computational times for various preconditioners in Experiment 3.

must choose a strategy to control the sparsity of the preconditioner. This here can be done by setting `Chop[Vi, 10-6]`, in our obtained approximate inverses.

5. Conclusions

In this work, we have developed an iterative method in inverse-finding of complex matrices belonging to the well-known class of Schulz-type methods.

We have proposed a simpler form of a recently published method (3) in the form (5) and proved analytically that the scheme possesses third-order convergence.

We have shown that the suggested method (16) reaches tenth order of convergence. We moreover have discussed that (16) can be considered for the left and right preconditioned systems under a certain condition. The efficacy of the new scheme was illustrated numerically using the computer programming package MATHEMATICA.

One may note that the approximate inverse obtained per step of Algorithm (5) or (16) can also easily be taken into account as a preconditioner to reduce the ill-conditioning of a system and let the users apply iterative methods such as GMRES or BiCGSTAB in solving large scale sparse linear systems of algebraic equations efficiently.

Conflict of Interests

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

References

- [1] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd edition, 2003.
- [2] F. Soleimani, F. Soleymani, A. Cordero, and J. R. Torregrosa, "On the extension of Householder's method for weighted Moore-Penrose inverse," *Applied Mathematics and Computation*, vol. 231, pp. 407–413, 2014.
- [3] G. Schulz, "Iterative Berechnung der Reziproken matrix," *Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, pp. 57–59, 1933.
- [4] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*, Berlin, Germany, Springer, 2nd edition, 2003.
- [5] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *Journal of Computational Physics*, vol. 182, no. 2, pp. 418–477, 2002.
- [6] 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.
- [7] V. Y. Pan, M. Van Barel, X. Wang, and G. Codevico, "Iterative inversion of structured matrices," *Theoretical Computer Science*, vol. 315, no. 2-3, pp. 581–592, 2004.
- [8] 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.
- [9] E. V. Krishnamurthy and S. K. Sen, *Numerical Algorithms: Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, India, 2007.
- [10] 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.
- [11] 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.
- [12] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, New York, NY, USA, 1966.
- [13] L. Weiguo, L. Juan, and Q. Tiantian, "A family of iterative methods for computing Moore-Penrose inverse of a matrix," *Linear Algebra and Its Applications*, vol. 438, no. 1, pp. 47–56, 2013.
- [14] F. Soleymani, "A fast convergent iterative solver for approximate inverse of matrices," *Numerical Linear Algebra with Applications*, vol. 21, pp. 439–452, 2014.
- [15] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, NY, USA, 1964.
- [16] <http://reference.wolfram.com/language/tutorial/LinearAlgebra-InMathematicaOverview>.
- [17] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.
- [18] L. Grosz, "Preconditioning by incomplete block elimination," *Numerical Linear Algebra with Applications*, vol. 7, no. 7-8, pp. 527–541, 2000.



Hindawi

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

