J. Appl. Math. & Informatics Vol. **32**(2014), No. 1 - 2, pp. 171 - 184 http://dx.doi.org/10.14317/jami.2014.171

# HIGHER ORDER ITERATIONS FOR MOORE-PENROSE INVERSES $^\dagger$

SHWETABH SRIVASTAVA\* AND D.K. GUPTA

ABSTRACT. A higher order iterative method to compute the Moore-Penrose inverses of arbitrary matrices using only the Penrose equation (ii) is developed by extending the iterative method described in [1]. Convergence properties as well as the error estimates of the method are studied. The efficacy of the method is demonstrated by working out four numerical examples, two involving a full rank matrix and an ill-conditioned Hilbert matrix, whereas, the other two involving randomly generated full rank and rank deficient matrices. The performance measures are the number of iterations and CPU time in seconds used by the method. It is observed that the number of iterations always decreases as expected and the CPU time first decreases gradually and then increases with the increase of the order of the method for all examples considered.

AMS Mathematics Subject Classification : 65H05, 65F10. *Key words and phrases* : Moore-Penrose inverse, Random matrices, Convergence analysis, Residual, Null space, Matrix rank.

#### 1. Introduction

The theory of generalized inverses has seen a substantial growth over the past few decades. Many applications of statistics, prediction theory, control analysis and numerical analysis often require computation of generalized inverses. The Moore-Penrose inverse is one of the most important generalized inverses of arbitrary singular square or rectangular (real or complex) matrix. It has been extensively studied by many researchers [4, 2, 6, 9, 5, 3] and many methods are proposed in the literature. Accordingly, it is important both practically and theoretically to find good higher order algorithms for computing a Moore-Penrose inverse of a given arbitrary matrix. Let  $\mathbb{C}^{m \times n}$  and  $\mathbb{C}_r^{m \times n}$  denote the set of

Received March 10, 2013. Revised April 22, 2013. Accepted April 26, 2013. Corresponding author.<sup>†</sup>This work was supported by the research grant of the Indian Institute of Technology Kharagpur, India.

 $<sup>\</sup>bigodot$  2014 Korean SIGCAM and KSCAM.

all complex  $(m \times n)$  matrices and all complex  $(m \times n)$  matrices with rank r, respectively. For  $A \in \mathbb{C}^{m \times n}$ , it is denoted by  $A^{\dagger}$  and defined as

$$AA^{\dagger} = P_{R(A)}$$
$$A^{\dagger}A = P_{R(A^{\dagger})}$$

where, R(A) and  $P_{R(A)}$  denote the range space of A and orthogonal projection on R(A) respectively. The unique matrix  $A^{\dagger}$  satisfies the following four equations

(i) 
$$AXA = A$$
, (ii)  $XAX = X$ , (iii)  $(AX)^* = AX$ , (iv)  $(XA)^* = XA$  (1)

Both direct and iterative methods (cf. [11, 10, 2, 3, 9]) can be used to compute  $A^{\dagger}$ . One of the most commonly used direct methods is the Singular Value Decomposition (SVD) method. For  $A \in \mathbb{C}^{m \times n}$ , the SVD method is a factorization of the form

 $A = U\Sigma V^*$ 

where, U is an  $m \times m$  complex unitary matrix,  $\Sigma$  is an  $m \times n$  rectangular diagonal matrix with nonnegative real numbers on the diagonal and V is an  $n \times n$  complex unitary matrix. The  $A^{\dagger}$  can then be written as

 $A^{\dagger} = V \Sigma^{\dagger} U^*$ 

where,  $\Sigma^{\dagger}$  is the generalized inverse of  $\Sigma$  obtained by replacing every nonzero diagonal entry in  $\Sigma$  by its reciprocal and then transposing the resulting matrix. This method is very accurate but time-intensive since it requires a large amount of computational resources, especially in the case of large matrices. The most frequently used iterative method for approximating  $A^{-1}$  is the method studied by Householder [14] by analyzing successive improvements of a matrix X to solve AX = M, A nonsingular, using the equation

$$X_{k+1} = X_k + C_k (M - AX_k), \ k \ge 0$$
<sup>(2)</sup>

One version of the above has the matrices  $C_k$  project on the *i*th row, where *i* cycles through all the rows of A. By taking M = I and  $C_k = V_k$ , the famous Newton's method originated in [21] is

$$V_{k+1} = V_k (2I - AV_k), \ k \ge 0$$
(3)

As usual, I denotes the identity matrix of an appropriate order. If L is the desired limit matrix and  $V_k$  is the k-th estimate of L, then the convergence properties of examined iterative method can be studied with the aid of error matrix  $E_k = V_k - L$ . If an iterative method is expressible as a simple matrix formula,  $E_{k+1}$  is sum of several terms

- zero order term consisting of a matrix which does not depend on  $E_k$ ,
- one or more first order matrix terms in which  $E_k$  or its conjugate transpose  $E_k^*$  appears only once,
- higher-order terms in which  $E_k$  or  $E_k^*$  appears at least twice

All suitable algorithms have a zero-order term equal to 0. Hence, the firstorder term determine the terminal convergence properties. The eigenvalues of  $A \in \mathbb{C}_r^{n \times n}$  are given by

$$\lambda_1(A) \ge \dots \ge \lambda_r(A) > \lambda_{r+1}(A) = \dots + \lambda_n(A) = 0.$$
(4)

It is further established that the eigenvalues of  $I - AV_0$  must have magnitude less than 1 to ensure the convergence of (3). Since the residuals  $R_k = I - AV_k$  in (3) satisfies  $||R_{k+1}|| \leq ||A|| ||R_k||^2$ , Newton's method is a second order iterative method [7]. Ben-Israel [11] used (3) and the starting value  $X_0 = \alpha A^*$ , where  $\alpha$ satisfies

$$0 < \alpha < \frac{2}{\lambda_1(AA^*)}.\tag{5}$$

The iterative scheme (3) is generalized by the iteration  $U_{k+1} = U_k(2P_R(A) - AU_k)$ , which converges to  $A^{\dagger}$ . Ben-Israel and Charnes [13] proved that the sequence

$$Y_k = \alpha \sum_{i=0}^k A^* (I - \alpha A A^*)^i, \ k = 0, 1, \dots$$
(6)

converges to  $A^{\dagger}$  under the assumption (5). Li et al.[6] established a family of iterative methods to compute the approximate inverse of a square matrix and inner inverse of a non-square matrix given by

$$V_{q+1} = V_q(kI - \frac{k(k-1)}{2}AV_q + \dots + (-1)^{k-1}(AV_q)^{k-1}), \ k = 2, 3, \dots$$
(7)

Chen and Wang [4] showed that this can be extended to compute  $A^{\dagger}$  with higher orders. Katskis et.al.[8] developed a much faster computational method to calculate the  $A^{\dagger}$  of singular square matrices and of rectangular matrices A. This method has significantly better accuracy than the already proposed methods and works for full and sparse matrices. Weiguo et al.[5] improves on the generalized properties of a family of iterative methods to compute the approximate inverses of square matrices and fixed inner inverses of rectangular matrices proposed in [6]. They have also established that this fixed inverse is the Moore-Penrose inverse of the considered matrix. Vasilios et.al.[3] also presented a very fast and reliable method to compute Moore-Penrose inverse. By using a general framework where analytic functions of scalers are first developed and then matrices substituted for the scalers, Katsaggelos and Efstratiadis [12] produced a convergence faster than quadratic, for restricted initial estimates.

In this paper, a higher order iterative method to compute the Moore-Penrose inverses of arbitrary matrices using only the Penrose equation (ii) is developed by extending the iterative method described in [1]. Convergence properties as well as the error estimates of the method are studied. The efficacy of the method is demonstrated by working out four numerical examples, two involving a full rank simple and an ill-conditioned Hilbert matrix, whereas, the other two involving full rank and rank deficient randomly generated matrices. The performance measures are the number of iterations and CPU time in seconds used by the method. It is observed that the number of iterations always decreases as expected and the CPU time first decreases gradually and then increases with the increase of the order of the method for all examples considered.

The paper is organized as follows. Section 1 is the introduction. In Section 2, the higher order iterative method for computing the Moore-Penrose generalized inverse of an arbitrary rectangular matrix is described. Some Lemmas and the convergence analysis of the method are also established. In Section 3, the efficacy of the method is demonstrated by working out four numerical examples, two involving a full rank simple and an ill-conditioned Hilbert matrix, whereas, the other two involving full rank and rank deficient randomly generated matrices. Finally, conclusions are included in Section 4.

#### 2. Higher order iteration with convergence Analysis

In this section, higher order iterative methods and their convergence analysis to compute the Moore-Penrose inverse  $A^{\dagger}$  of an arbitrary complex matrix  $A \in \mathbb{C}^{m \times n}$  is described. Let  $A^t$ ,  $A^*$ ,  $\mu(A)$ ,  $\nu(A)$  and rank(A), represent the transpose, the conjugate transpose, the range space, the null space and the rank of the matrix  $A \in \mathbb{C}^{m \times n}$ , respectively.

**Definition 2.1** ([19]). Let  $A \in \mathbb{C}^{m \times n}$ , with column vectors  $v_1, v_2, \ldots, v_n$ , the set

$$\mu(A) = \{c_1v_1 + c_2v_2 + \ldots + c_nv_n | c_1, c_2, \ldots, c_n \text{ are scalers}\}$$

are called the Range (the Image, or the Column Space) of A. The set

$$v(A) = \{x \in C^n | Ax = 0\}$$

is called the Null Space (or the Kernel) of A and for any  $A \in \mathbb{C}^{m \times n}, B \in \mathbb{C}^{s \times t}$ , we call

$$\mu(A,B) = \{ X \in \mathbb{C}^{m \times t} | X = AYB, \forall Y \in \mathbb{C}^{n \times s} \},\$$

 $\nu(A, B) = \{ Y \in \mathbb{C}^{n \times s} | AYB = 0 \}$ 

the Range of (A, B) and the Null Space of (A, B) respectively.

We shall now describe the higher order iterative method extending the method described in [1] for computing the Moore-Penrose generalized inverses. Petković et al. [1] proposed a quadratically convergent iterative method for computing  $A^{\dagger}$  based on Penrose equations (*ii*) and (*iv*) as follows. Let

$$X^* = (XAX)^* = X^*(XA)^* = X^*XA$$

Hence, for arbitrary  $\beta \in \mathbb{R}$ , we get

$$X^{*} = X^{*} - \beta (X^{*}XA - X^{*}) = X^{*}(I - \beta XA) + \beta X^{*}$$

or equivalently,

$$X = (I - \beta X A)^* X + \beta X$$

This leads to the following iterative method

$$X_{k+1} = (1+\beta)X_k - \beta X_k A X_k, \ k = 0, 1, \dots$$
(8)

where,  $X_0 = \beta A^*$  for an appropriate real number  $\beta$ . For  $\beta < 1$ , the method (8) has a linear convergence while for  $\beta = 1$  its convergence is quadratic. The first-order and the second-order error terms of (8) are

$$error_1 = (1 - \beta)E_k, \ error_2 = -\beta E_k A E_k \tag{9}$$

where,  $E_k = X_k - A^{\dagger}$  is the error matrix. Now, using only the Penrose equations (*ii*) given by X = XAX and for arbitrary  $\beta \in \mathbb{R}$ , we get

$$X = X + \beta(2X - 3XAX + XAXAX)$$

or, equivalently

$$X = X + \beta X (2I - 3AX + (AX)^2)$$

This leads to the following third order extension of method (8)

$$X_{k+1} = X_k + \beta X_k (2I - 3AX_k + (AX_k)^2), \ k = 0, 1, \dots$$
(10)

for  $X_0 = \beta A^*$ . Continuing in a similar manner, this can further be extended to the *p*th order for  $p \ge 2$ , given by

$$X_{k+1} = X_k + \beta X_k [(I - AX_k) + (I - AX_k)^2 + \ldots + (I - AX_k)^{p-1}], \ k = 0, 1, 2, \ldots$$
(11)  
for  $X_0 = \beta A^*$  where  $\beta$  is an appropriate real number. The following Lemmas

for  $X_0 = \beta A^*$ , where  $\beta$  is an appropriate real number. The following Lemmas will be used for establishing the convergence of these iterative methods.

**Lemma 2.2.** For all  $k \ge 0$ , the sequence  $\{X_k\}$  generated by (10) and (11) satisfies

(1) 
$$X_k A = (X_k A)^*$$
 (2)  $X_k A A^{\dagger} = X_k$  (3)  $A^{\dagger} A X_k = X_k$ 

*Proof.* This Lemma can be proved by induction. Clearly, (1) holds for k = 0 as  $X_0A = \beta A^*A = (X_0A)^*$ . Assume that it holds for some k. To show that it also holds for k + 1, we consider

$$(X_{k+1}A)^* = (X_kA)^* + \beta \{ 2(X_kA)^* - 3((X_kA)^*)^2 + ((X_kA)^*)^3 \}$$
  
=  $X_kA + \beta (2X_kA - 3(X_kA)^2 + (X_kA)^3)$   
=  $X_kA + \beta (2X_k - 3X_kAX_k + X_k(AX_k)^2)A$   
=  $X_{k+1}A$ 

Hence it holds for all  $k \ge 0$ . Likewise for *p*th order method (11), we get

$$(X_{k+1}A)^* = \{X_kA + \beta X_k[(I - AX_k) + \dots (I - AX_k)^{p-1}]A\}^*$$
  
=  $(X_kA)^* + \beta[(X_k(I - AX_k)A)^* + \dots + (X_k(I - AX_k)^{p-1}A)^*]$   
=  $X_kA + \beta[X_k(I - AX_k)A + \dots + X_k(I - AX_k)^{p-1}A]$   
=  $X_kA + \beta X_k[(I - AX_k) + \dots (I - AX_k)^{p-1}]A$   
=  $X_{k+1}A$ 

Clearly, (2) trivially holds for k = 0. Let it also holds for some k. To show that it also holds for k + 1, we get

$$X_{k+1}AA^{\dagger} = X_kAA^{\dagger} + \beta X_k(2I - 3AX_k + (AX_k)^2)AA^{\dagger}$$
$$= X_k + \beta X_k(2I - 3AX_k + (AX_k)^2)$$
$$= X_{k+1}$$

Likewise for pth order method (11), we get

$$X_{k+1}AA^{\dagger} = X_kAA^{\dagger} + \beta X_k[(I - AX_k) + \dots + (I - AX_k)^{p-1}]AA^{\dagger}$$
  
=  $X_k + \beta X_k[(I - AX_k) + \dots + (I - AX_k)^{p-1}]$   
=  $X_{k+1}$ 

Proceeding in a similar manner, (3) can easily be proved for (10) and (11). Hence, the Lemma 2.1 is proved for all  $k \ge 0$ .

**Theorem 2.3.** Let  $0 \neq A \in \mathbb{C}^{m \times n}$ ,  $X = A^{\dagger}$ , the initial approximation  $X_0 = \beta A^*$ ,  $\beta \in (0, 1]$  and its residual  $R_0 = (X_0 - X)A$  satisfy  $||R_0|| < 1$ . The sequence  $\{X_k\}$  generated by (10) starting with  $X_0 = \beta A^*$  converges to the Moore-Penrose inverse  $A^{\dagger}$ . It has linear convergence for  $\beta \neq 1$  and third order convergence for  $\beta = 1$ . Its first, second and the third order error terms are given by

$$error_1 = (1 - \beta)E_k, \ error_2 = 0 \ and \ error_3 = \beta E_k (AE_k)^2$$
 (12)

where,  $E_k = X_k - A^{\dagger}$  denotes the error matrix.

*Proof.* Using Lemma 2.2 and substituting for  $X_{k+1}$ , we get

$$||X_{k+1} - X|| = ||X_{k+1}AX - XAX|| \le ||X_{k+1}A - XA|| ||X||$$

Using Lemma 2.2 and (10), we get

$$X_{k+1}A - XA = (X_k + \beta X_k (2I - 3AX_k + (AX_k)^2)A - XA$$
  
=  $X_k A - XA + \beta (2X_k A - 3(X_k A)^2 + (X_k A)^3)$   
=  $X_k A - XA + \beta ((X_k A - XA)^3 - (X_k A - XA))$ 

Thus, the sequence of residual matrices defined by  $R_k = X_k A - X A$  satisfies the following recurrence relation

$$R_{k+1} = (1 - \beta)R_k + \beta R_k^3$$
(13)

Let  $s_k = ||R_k||$ . Now, for the convergence of the sequence  $\{X_k\}$ , we require that  $s_k \to 0$  as  $k \to \infty$ . This can be shown by mathematical induction. It trivially holds for k = 0, since,  $s_0 = ||R_0|| = ||X_0A - XA|| < 1$ . Let it holds for some k, i.e.,  $s_k < 1$ . To show that it also holds for k + 1, we take norm on (13) and get

$$s_{k+1} \le \beta s_k^3 + (1-\beta)s_k < \beta s_k + (1-\beta)s_k = s_k < 1$$
(14)

Thus,  $s_k$  is a monotonically decreasing bounded sequence converging to s as  $k \to \infty$  and  $0 \le s < 1$ . From (14), we get

$$s \le \beta s^3 + (1 - \beta)s$$

This gives either s = 0 or  $s \ge 1$ . Thus, s = 0. This completes the proof that  $s_k \to 0$  when  $k \to \infty$ . Thus,  $X_k \to X$  as  $k \to \infty$ . Substituting  $X_k = X + E_k$  in (10) and rearranging the terms, we get  $E_{k+1}$  given by

$$E_{k+1} = (1+2\beta)E_k - 2\beta E_kAX - 3\beta XAE_k + 2\beta XAE_kAX - 3\beta E_kAE_k + 2\beta E_kAE_kAX + \beta X(AE_k)^2 + \beta E_k(AE_k)^2$$

This leads to

$$error_1 = (1 - \beta)E_k$$
$$error_2 = 0$$

and

$$error_3 = \beta E_k (AE_k)^2$$

One can easily see that the method is linear convergent for  $\beta \neq 1$  and of third order convergent for  $\beta = 1$  as  $error_1$  and  $error_2$  are equal to 0.

**Theorem 2.4.** Let  $0 \neq A \in \mathbb{C}^{m \times n}$ ,  $X = A^{\dagger}$ , the initial approximation  $X_0 = \beta A^*$ ,  $\beta \in (0,1]$  and its residual  $R_0 = (X_0 - X)A$  satisfy  $||R_0|| < 1$ . The sequence  $\{X_k\}$  generated by (11) starting with  $X_0 = \beta A^*$  converges to the Moore-Penrose inverse  $A^{\dagger}$ . It has linear convergence for  $\beta \neq 1$  and pth order convergence for  $\beta = 1$ , where,  $p \geq 2$  is a positive integer. All error terms of the method vanishes except the first and the pth order error terms given by

$$error_1 = (1 - \beta)E_k, \ error_p = (-1)^{p-1}\beta E_k (AE_k)^{p-1}$$
 (15)

*Proof.* Using Lemma 2.2 and substituting for  $X_{k+1}$ , we get

$$||X_{k+1} - X|| = ||X_{k+1}AX - XAX|| \le ||X_{k+1}A - XA|| ||X||$$

Again using Lemma 2.2 and (11), this gives

$$X_{k+1}A - XA = (X_k + \beta X_k [(I - AX_k) + (I - AX_k)^2 + \dots + (I - AX_k)^{p-1}])A - XA.$$

Thus, the sequence of residual matrices defined by  $R_k = X_k A - X A$  satisfies the following recurrence relation

$$R_{k+1} = (1-\beta)R_k + (-1)^{p-1}\beta R_k^p \tag{16}$$

Proceeding similar to theorem 2.3, it can be proved that  $X_k \to X$  as  $k \to \infty$ . Substituting  $X_k = X + E_k$  in (11) and rearranging the terms, we get  $E_{k+1}$  given by

$$E_{k+1} = (1+\beta)E_k - \beta XAE_k - \beta E_kAX + (-1)^{p-1}\beta E_k(AE_k)^{p-1}$$

On simplification, we get all error terms equal to zero except  $error_1$  and  $error_p$  given by

$$error_1 = (1 - \beta)E_k$$
$$error_p = (-1)^{p-1}\beta E_k (AE_k)^{p-1}$$

One can easily see that the method is linearly convergent for  $\beta \neq 1$  and of *p*th order convergent for  $\beta = 1$  as all other error terms vanishes. Hence, the theorem is proved.

We must note here that the convergence of the above methods (10) and (11) require the condition  $\|(\beta A^* - X)A\| < 1$ . To verify this, we need an equivalent condition which does not contain the Moore-Penrose inverse X or  $A^{\dagger}$ . We follow exactly the same way as done in [1] by using the following Lemma.

**Lemma 2.5** ([1]). Let the eigenvalues of matrix  $A^*A$  satisfies (4). Condition  $\rho((\beta A^* - X)A) < 1$  is satisfied under the assumption

$$\max_{1 \le i \le r} |1 - \beta \lambda_i(A^*A)| < 1 \tag{17}$$

To establish this Lemma, we need the following Lemmas.

**Lemma 2.6** ([15]). Let  $M \in \mathbb{C}^{n \times n}$  and  $\epsilon > 0$  be given. There is at least one matrix norm  $\| \cdot \|$  such that

$$\rho(M) \le \parallel M \parallel \le \rho(M) + \epsilon$$

where  $\rho(M) = \max\{|\lambda_1(M)|, \ldots, |\lambda_n(M)|\}$  denotes the spectral radius of M. Lemma 2.7 ([18]). If  $P \in \mathbb{C}^{n \times n}$  and  $S \in \mathbb{C}^{n \times n}$  are such that  $P = P^2$  and PS = SP then

$$\rho(PS) \le \rho(S)$$

**Remark 2.1.** For any  $A \in \mathbb{C}^{m \times n}$ , the sequence generated by our higher order iterative methods starting with  $X_0 = \beta A^*$  are convergent for any  $\beta$  satisfying  $0 < \beta < 2/\sigma^2(A)$ , where  $\sigma(A) = ||A||_2$ .

### 3. Numerical examples

In this section, four numerical examples are worked out to demonstrate the efficacy of our *p*th order method for various values of *p*. The mean CPU time in second and the number of iterations are measured as performance parameters. For full rank matrices the number of iterations used are compared by plotting figures with *x*-axis representing the values of *p* and *y*-axis representing the number of iterations. For randomly generated matrices, we have tested 50 times matrices and the mean CPU time in second are measured and tabulated for values of  $2 \le p \le 7$  and the number of iterations are compared by plotting figures with *x*-axis representing the times of choice of random matrices and *y*-axis representing the number of iterations only for p = 2 and p = 3. The value of  $\epsilon = 10^{-7}$  and the maximum number of iterations equal to 100 are taken as the termination criteria. All the examples are run on an Intel core 2 Duo processor running at 2.80GHz and using MATLAB 7.4 (R2009b).

**Example 3.1.** Consider the matrix A of order  $(5 \times 4)$  given by

	0.2794	0.1676	0.0645	0.2326
	0.0065	0.2365	0.2274	0.1261
A =	0.2271	0.1430	0.1009	0.2867
	0.1265	0.1015	0.1806	0.2846
	0.2773	0.0632	0.0503	0.1979

The choice  $\beta=0.60$  satisfies the convergence criteria given by

 $\max_{1 \le i \le 4} | 1 - \beta \lambda_i(A^*A) | = 0.9988 < 1$ 

since the eigenvalues of the matrix  $A^*A$  are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.0020, 0.0146, 0.0832, 0.6152)$$

The iterative method (11) generates a sequence of iterates  $\{X_k\}$  converging to the Moore-Penrose generalized inverse  $A^{\dagger}$  given by

	(-0.2165)	1.4802	-4.9702	-1.3732	8.4865	
<u>4</u> †	5.0277	1.8673	4.1653	-4.6975	-6.3778	
A' =	-5.3215	4.5524	-8.4278	3.4688	10.5748	
	0.8566	-4.0180	6.9330	3.0649	-7.8449	

The comparison of number of iterations are plotted in Figure 1. It can be observed from Figure 1 that the iterative method (11) converges to the Moore-Penrose generalized inverse  $A^{\dagger}$  in 36 iterations for p = 2 and as expected as the order p increases, it reduces to 25 for p = 8.



FIGURE 1. Number of iterations versus the value of p for example 3.1

**Example 3.2.** Consider a  $(30 \times 30)$  matrix A whose elements are generated randomly from [-0.2, 0.2]. Taking the termination criteria as

 $\max\{\|AX_kA - A\|_F, \|X_kAX_k - X_k\|_F, \|(AX_k)^* - AX_k\|_F, \|(X_kA)^* - X_kA\|_F\} \le \epsilon$ 

where,  $\|.\|_F$  stands for the Frobenius-norm of a matrix, Table 1 shows the comparison of mean CPU time in seconds for for  $2 \le p \le 7$ . One can easily see that with the increase of p, the CPU time decreases initially and then starts increases after p > 4. The comparison of number of iterations are plotted in Figure 2 for p = 2 and p = 3. Similar results are also observed for higher order random matrices, for example,  $(40 \times 40)$  matrix A whose elements are also randomly generated from [-0.2, 0.2].

TABLE 1. Comparison of CPU time

р	p =2	p =3	p =4	p =5	p =6	p = 7
CPU time	2.1387e - 5	1.6948e - 5	1.6245e - 5	1.6554e - 5	1.7205e - 5	1.8946e - 5



FIGURE 2. Comparison of number of iterations for example 3.2

**Example 3.3.** Consider a  $(100 \times 50)$  rank deficient matrix A whose elements are generated randomly from [-0.2, 0.2]. Taking the termination criteria as  $\max\{\|AX_kA - A\|_F, \|X_kAX_k - X_k\|_F, \|(AX_k)^* - AX_k\|_F, \|(X_kA)^* - X_kA\|_F\} \le \epsilon$ 

where,  $\|.\|_F$  stands for the Frobenius-norm of a matrix, Table 2 shows the comparison of mean CPU time in seconds for for  $2 \le p \le 7$ . One can easily see that with the increase of p, the CPU time decreases initially and then starts increases after  $p \ge 4$ . The comparison of number of iterations are plotted in Figure 3 for p = 2 and p = 3.



FIGURE 3. Comparison of number of iterations for example 3.3

TABLE 2. Comparison of CPU time

р	p = 2	p =3	p = 4	p = 5	p =6	p = 7
CPU time	2.5287e - 5	1.7793e - 5	1.8346e - 5	1.9451e - 5	2.0135e - 5	2.1137e - 5

**Example 3.4.** Consider the ill-conditioned Hilbert matrix A of order  $(5 \times 5)$  given by

	/ 1.0000	0.5000	0.3333	0.2500	0.2000	
	0.5000	0.3333	0.2500	0.2000	0.1667	
A =	0.3333	0.2500	0.2000	0.1667	0.1429	
	0.2500	0.2000	0.1667	0.1429	0.1250	
	0.2000	0.1667	0.1429	0.1250	0.1111	)
The choice $\beta = 0.80$	) satisfies	the conv	ergence o	riteria g	iven by	,
	$\max_{1 \le i \le 5}$	$_{5} \mid 1 - \beta \lambda$	$\Lambda_i(A^*A) \mid$	= 0.9999	< 1	

since the eigenvalues of the matrix  $A^*A$  are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (2.4556, 0.0435, 0.0001, 9.357e - 8, 1.081e - 11)$$

The iterative method (11) generates a sequence of iterates  $\{X_k\}$  converging to the Moore-Penrose generalized inverse  $A^{\dagger}$  given by

	/ 20	-300	1050	-1400	630
	-300	4800	-18900	26880	-12600
$A^{\dagger} =$	1050	-18900	79380	-117600	56700
	-1400	26880	-117600	179200	-88200
	630	-12600	56700	-88200	44100 /

The comparison of number of iterations are plotted in Figure 4. It can be observed from Figure 4 that the iterative method (11) converges to the Moore-Penrose generalized inverse  $A^{\dagger}$  in 51 iterations for p = 2 and as expected as the order p increases, it reduces to 19 for p = 10.



FIGURE 4. Number of iterations versus the value of p for example 3.4

## 4. Conclusions

A quadratically convergent iterative method proposed by [1] is extended to a family of higher order iterative methods to compute the Moore-Penrose generalized inverses of arbitrary singular or rectangular (real or complex) matrices. This extension is carried out by using only the Penrose equation (ii) in place of (ii) and (iv) as used in [1]. Convergence properties as well as the error estimations are studied. The efficacy of the method is demonstrated by working out four numerical examples involving full rank and rank deficient randomly generated matrices and ill-conditioned Hilbert matrix. The performance in terms of computational time and the number of iterations are evaluated with respect to the order of the iterative methods. It is observed that the number of iterations

182

always decreases as expected and the CPU time first decreases gradually and then increases with respect to the order of the iterative methods for all examples.

#### References

- Marko D. Petković and Predrag S.Stanimirović, Iterative method for computing Moore-Penrose inverse based on Penrose equations, Journal of Computational and Applied Mathematics 235 (2011), 1604-1613.
- W. Guo and T. Huang, Method of elementary transformation to compute Moore-Penrose inverse, Appl. Math. Comput. 216 (2010), 1614-1617.
- Vasilios N. Katsikis, Dimitrios Pappas and Athanassios Petralias, An improved method for the computation of the Moore-Penrose inverse matrix, Appl. Math. Comput. 217 (2011), 9828-9834.
- Haibin Chen and Yiju Wang, A Family of higher-order convergent iterative methods for computing the Moore-Penrose inverse, Appl. Math. Comput. 218 (2011), 4012-4016.
- Li Weiguo, Li Juan and Qiao Tiantian, A family of iterative methods for computing Moore-Penrose inverse of a matrix, Linear Algebra and its Applications 438 (2013), 47-56.
- Weiguo Li and Zhi 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), 3433-3442
- 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), 1109-1130.
- V.N. Katsikis, D. Pappas and T. Huang, Fast computing of the Moore-Penrose inverse matrix, Electronic Journal of Linear Algebra 17 (2008), 637-650.
- F. Toutounian and A. Ataei, A new method for computing Moore-Penrose inverse matrices, Journal of Computational and applied Mathematics 228 (2009), 412-417.
- P. Courrieu, Fast Computation of Moore-Penrose Inverse matrices, Neural Information Processing-Letters and Reviews 8 (2005), 25-29.
- 11. A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer-Verlag, NewYork, 2003.
- A.K.Katsaggelos and S.N. Efstratiadis, A class of iterative signal restoration algorithms, IEE Trans. Acoust. Speech Signal Process, 38 (1990), 778-793.
- A. Ben-Israel and A.Charnes, Contribution to the theory of generalized inverses, J.Soc.Indust.Appl.Math. 11 (1963), 667-669.
- A.S.Householder, The Theory of Matrices in Numerical Analysis, Dover Publications, Newyork, 1964.
- R.A. Horn and C.R. Johnson, *Matrix Analysis* Cambridge University Press, Cambridge, UK, 1986.
- F.J. Hall and C.D. Meryer Jr., Generalized inverses of the fundamental bordered matrix used in linear estimation, Sankhya Ser. A 37 (1975), 428-438.
- B. Nacevska, Iterative methods for computing generalized inverses and splittings of operators, Appl. Math. Comput., 208 (2009), 186-188.
- P.S.Stanimirovic and D.S. Cvetkovic-illic, Successive matrix squaring algorithm for computing outer inverses, Appl. Math. Comput., 203 (2008), 19-29.
- W. Sun and Y. Yuan, Optimization Theory and Methods- Nonlinear Programming, Springer, New York, 2006.
- 20. Stephen L. Campbell, and Carl D. Meyer, *Generalized Inverses of Linear Transformations*, SIAM, 2009.
- G.Schultz, iterative Berechnmg der reziproken matrix, Angew. Math. Mech., 13 (1933), 57-59.

**Shwetabh Srivastava** is a Ph.D. research scholar in the Department of Mathematics, Indian Institute of Technology Kharagpur, India. His research interests include Theory of Generalized inverses and their applications.

Department of Mathematics, IIT Kharagpur, India. e-mail: shwetabh20059@gmail.com

**D.K. Gupta** is a senior professor in the Department of Mathematics, Indian Institute of Technology Kharagpur, India. His research interests are Numerical Analysis and Computer Science.

Department of Mathematics, IIT Kharagpur, India. e-mail: dkg@maths.iitkgp.ernet.in

184