Applied Mathematics and Computation 371 (2020) 124957

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Computing the Moore-Penrose inverse using its error bounds☆

Predrag S. Stanimirović[a,*], Falguni Roy[b,c], Dharmendra K. Gupta[b], Shwetabh Srivastava[d]

[a] Faculty of Science and Mathematics, University of NiŠ, ViŠegradska 33, NiŠ 18000, Serbia
[b] Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur 721302, India
[c] Department of Mathematical and Computational Sciences, National Institute of Technology Karnataka, Surathkal 575025, India
[d] Department of Mathematics, C.M.P. Degree College, University of Allahabad, India

ARTICLE INFO

Article history:
Received 6 November 2016
Revised 25 November 2019
Accepted 1 December 2019

MSC:
15A09
65F20

Keywords:
Moore-Penrose inverse
Interval method
Newton iteration
Interval iterative method

ABSTRACT

A new iterative scheme for the computation of the Moore-Penrose generalized inverse of an arbitrary rectangular or singular complex matrix is proposed. The method uses appropriate error bounds and is applicable without restrictions on the rank of the matrix. But, it requires that the rank of the matrix is known in advance or computed beforehand. The method computes a sequence of monotonic inclusion interval matrices which contain the Moore-Penrose generalized inverse and converge to it. Successive interval matrices are constructed by using previous approximations generated from the hyperpower iterative method of an arbitrary order and appropriate error bounds of the Moore-Penrose inverse. A convergence theorem of the introduced method is established. Numerical examples involving randomly generated matrices are presented to demonstrate the efficacy of the proposed approach. The main property of our method is that the successive interval matrices are not defined using principles of interval arithmetic, but using accurately defined error bounds of the Moore-Penrose inverse.

1. Introduction

The problem of finding generalized inverses of rectangular or singular square real or complex matrices is an important and challenging area in matrix theory, numerical analysis and in applied sciences. There is a substantial growth in the study of generalized inverses over the past few decades. For example, pseudoinversion of a singular square matrix is needed in the case of the minimization of indefinite quadratic forms [5] which involve positive semi-definite matrices. Many real world applications of control theory, numerical analysis, statistics, image processing, prediction theory, etc., often require computation of these generalized inverses. For further details, one may refer to [3,10,19]. Generalized inverses also emerge from the necessity of finding a solution of a singular system of linear equations. Such type of singular linear systems arise for example in the discrete modeling of Neumann problems for elliptic partial differential equations as well as in certain problems from statistics, such as the steady state of Markov chains and computerized topography. However, generalized inverses are not easily computable, especially for matrices of large dimensions which usually arise in practical applications.

☆ Predrag Stanimirović gratefully acknowledge support from the Research Project 174013 of the Serbian Ministry of Science.

* Corresponding author.
E-mail addresses: pecko@pmf.ni.ac.rs, peckois@ptt.rs (P.S. Stanimirović), royfalguni@nitk.edu.in (F. Roy), dkg@maths.iitkgp.ernet.in (D.K. Gupta), shwetabhiit@gmail.com (S. Srivastava).

https://doi.org/10.1016/j.amc.2019.124957
0096-3003/

The most important generalized inverse is the Moore-Penrose generalized inverse (or pseudoinverse), denoted by $A^\dagger$ for a matrix A. The Moore-Penrose generalized inverse $A^\dagger$ of $A \in \mathbb{C}^{m\times n}$ is defined as the unique matrix $X \in \mathbb{C}^{n\times m}$ such that the following four Penrose equations are satisfied:

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

Both direct and iterative methods for computing $A^\dagger$ have been proposed [3,7–9,21]. The most popular direct methods are based on the singular value decomposition (SVD) [3], QR factorization [8], conjugate Gram-Schmidt process [21], LDL^* factorization [13,18], Gaussian elimination [20], etc. Usually these methods give very accurate results but take a large amount of computational resources and time, especially in the case of large matrices. For this reason, iterative methods to compute $A^\dagger$ for large-scale problems have been proposed. The convergence and error bounds of these iterative schemes have been analyzed. The famous Schultz iterative method [3] for computing $A^\dagger$ is given by the iterative rule

$$X_{k+1} = X_k + X_k(I - AX_k), \quad k = 0, 1, \ldots \tag{1}$$

The initial approximation X_0 must be chosen in such a way that the eigenvalues of $I - AX_0$ have a magnitude less than 1 in order to ensure the convergence of (1). This condition is satisfied in the case $X_0 = \alpha A^*$, where $0 < \alpha < \frac{2}{\lambda_{\max}(AA^*)}$ and $\lambda_{\max}(AA^*)$ denotes the maximum eigenvalue of AA^*. Under these conditions, Ben-Israel and Charnes [4] proved that the sequence given by

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

also converges to $A^\dagger$. Li et al. [9] developed the following family of iterative methods for computing the approximate generalized inverse of a square matrix and inner inverse of a non-square matrix:

$$X_{k+1} = X_k\left(pI - \frac{p(p-1)}{2}AX_k + \ldots + (-1)^{p-1}(AX_k)^{p-1}\right), \quad k = 0, 1, 2, \ldots \tag{2}$$

The iterations (2) in the case $p = 2$ reduce to (1). Chen and Wang [6] extended this result to $A^\dagger$. Weiguo et al. [23] verified the convergence of (2) to $A^\dagger$ using matrix calculus and showed, by comparing the computing time and efficiency index, that the third order iterations are as good as the second order iterations. An iterative method for computing $A^\dagger$ which is based on the Penrose equations (ii) and (iii) was discussed in [11]. Convergence properties of the method as well as their first-order and second-order error terms were also considered. A higher order iterative method (or hyperpower iterative method) for computing $A^\dagger$ using only the Penrose equation (ii) was proposed in [16].

In general, all iterative methods require initial conditions which are rigorous and sometimes cannot be fulfilled easily. However, the vast majority of these methods do not take care of computational errors, especially round-off errors of computation. The authors of [11,12] demonstrated that, under certain condition, a scaled Newton iterative method can be unstable due to the accumulation of the roundoff errors. Moreover, numerical examples presented in [8] indicate that the scaled Newton iterations are usually unstable, especially for ill-conditioned matrices. Thus, the validation of the computed approximation to $A^\dagger$ can not be ascertained.

Another efficient approach to compute the pseudoinverse is based on the concepts of interval analysis [1,2]. This approach not only leads to the computation of $A^\dagger$ but also rigorously estimates the involved computational errors simultaneously. The computation in interval analysis is carried out using intervals of real numbers in place of real numbers. A few research works considered the computation of generalized inverses using interval analysis. An interval iterative method for estimating error bounds of the Moore-Penrose generalized inverse of arbitrary rectangular or singular square real or complex matrices using the concepts of interval analysis was developed by Zhang et al. [24]. The drawback of that method is its applicability to only matrices of full row (column) ranks.

The aim of this paper is to propose a new iterative scheme for deriving error bounds of the Moore-Penrose generalized inverse of an arbitrary rectangular or singular real or complex matrix of arbitrary rank. Zhang et al. [24] calculated error bounds for computing the pseudoinverse of full rank (column) matrices, arising from the concepts of interval analysis. Therefore, the interval iterative method from [24] is not applicable to rank deficient matrices. In the present paper, we use an alternative approach, based on a sequence of inclusion interval matrices containing the Moore-Penrose inverse. The proposed iterative scheme requires that the rank of the input matrix A is known or computed. Starting from the initial interval matrix constructed appropriately by using the initial approximation to the Moore-Penrose inverse in conjunction with the hyperpower iterative method for generating successive approximations, a sequence of monotonic inclusion interval matrices is generated. Each interval matrix included in the sequence contains the Moore-Penrose inverse $A^\dagger$. A convergence theorem is established. Theoretical results show that the sequence of monotonic inclusion interval matrices converges to $A^\dagger$. Numerical examples involving randomly generated matrices are solved to demonstrate the efficacy of our approach. In comparison with the results obtained using the method from [24], it is found that our approach gives better accuracy.

The paper is organized as follows. Section 1 is introductory. Some basic notations, definitions and concepts of generalized inverses and interval analysis used in the paper are described briefly in Section 2. The proposed iterative approach is investigated and discussed in Section 3 along with the results establishing its convergence analysis. A computational algorithm is also developed in the same section. Numerical examples are worked out in Section 4 to demonstrate the efficacy of the proposed approach. Finally, conclusions are included in Section 5.

2. Preliminaries

In this section, notations, definitions and concepts of generalized inverses used in the paper are described in brief. We shall follow the notational conventions of [3]. Let $\mathbb{C}^{m\times n}$, A^*, $\mathcal{R}(A)$, $\mathcal{N}(A)$ denote, respectively, the set of all $m\times n$ complex matrices, the conjugate transpose, range space, null space of $A\in\mathbb{C}^{m\times n}$.

2.1. Moore-Penrose generalized inverse of complex matrices

Let $A\in\mathbb{C}^{m\times n}$ and $A^\dagger$ be the Moore-Penrose generalized inverse of A. Clearly, $A^\dagger$ satisfies $AA^\dagger=P_{\mathcal{R}(A),\mathcal{N}(A^*)}$ and $A^\dagger A=P_{\mathcal{R}(A^*),\mathcal{N}(A)}$, where $P_{L,M}$ denotes the orthogonal projector on the subspace L along the subspace M. The orthogonal projector $P_{\mathcal{R}(A),\mathcal{N}(A^*)}$ is a transformation that projects each $x\in\mathbb{C}^n$ into its orthogonal projection on the subspace $\mathcal{R}(A)$. A matrix C is called Hermitian (resp. idempotent) if it satisfies $C^*=C$ (resp. $C^2=C$). Other types of commonly used generalized inverses are the Drazin inverse A^d [17], the group inverse A_g [22], the Bott-Duffin inverse $A^{(-1)}_{(L)}$ [3], the weighted Moore-Penrose inverse [15], etc. All of these inverses are special cases of the outer inverse $A^{(2)}_{T,S}$ of $A\in\mathbb{C}^{m\times n}$ having the prescribed range T and null space S, where T and S are subspaces of $\mathbb{C}^n$ and $\mathbb{C}^m$ respectively. The Euclidean vector norm or the spectral norm is denoted by the symbol $\|.\|$, while the Frobenius matrix norm is denoted by $\|.\|_F$ and defined as $\|A\|_F^2=\text{trace}(A^*A)$. The spectral radius $\rho(A)$ of a square matrix $A\in\mathbb{C}^{n\times n}$ is the maximal value among the n moduli of the eigenvalues of A: $\rho(A)=\max\{|\lambda|:\lambda\in\lambda(A)\}$, where $\lambda(A)$ is the spectrum of A. For an iterative method generating the sequence of matrices $\{X_k\}$ to approximate the generalized inverse, its convergence rate is determined in terms of the corresponding sequence of residuals $R_k=P_{\mathcal{R}(A)}-AX_k$. This sequence converges to the zero matrix 0 as $X_k\to A^\dagger$. It is said to be of pth order if there exists a positive constant M such that $\|R_{k+1}\|\le M\|R_k\|^p,\ k=0,1,\dots,$ for some $p>1$.

Lemma 2.1 ([3], Ch. 7, Thm 4). *Let $0\neq A\in\mathbb{C}^{m\times n}$, and let the initial approximation X_0 and its residual $R_0=P_{R(A)}-AX_0$ satisfy*

$$X_0\in\mathcal{R}(A^*,A^*)=\{X:X=A^*BA^*,\ B\in\mathbb{C}^{m\times n}\},\quad \rho(R_0)=\rho(P_{\mathcal{R}(A)}-AX_0)<1. \tag{3}$$

Then the sequence

$$X_{k+1}=X_k\sum_{i=0}^{p-1}(I-AX_k)^i=X_k\sum_{i=0}^{p-1}T_k^i \tag{4}$$

converges to $A^\dagger$ as $k\to\infty$, for any integer $p\ge 2$, and the corresponding sequence of residuals satisfies

$$\|R_{k+1}\|\le\|R_k\|^p,\ k=0,1,\dots \tag{5}$$

The iterative method (4) is known as the hyperpower iterative extension of order p of the Schultz method. Later in this paper, this method is termed as Schultz type iterative scheme (SIS). Next lemma is crucial for developing the proposed iterative scheme (PIS). Lemma 2.2 gives a representation of $A^\dagger$ in terms of the convergent series of the residual $R_0=P_{R(A)}-AX_0$, whereas Lemma 2.1 gives an iterative formula for computing $A^\dagger$ using $T_k=I-AX_k$.

Lemma 2.2. *Let $0\neq A\in\mathbb{C}^{m\times n}$ and X_0,R_0 be same as defined in Lemma 2.1. Then*

$$A^\dagger=X_0(I-R_0)^{-1}=X_0\sum_{i=0}^{\infty}R_0^i. \tag{6}$$

Proof. The following holds for X_0 satisfying (3):

$$A^\dagger(I-R_0)=A^\dagger(I-P_{\mathcal{R}(A)}+AX_0)=A^\dagger-A^\dagger AA^\dagger+A^\dagger AX_0=X_0.$$

According to the assumption, the matrix R_0 satisfies $\rho(R_0)<1$. This implies that 1 can not be an eigenvalue of R_0 and further 0 is not an eigenvalue of $I-R_0$. Thus $I-R_0$ is invertible and

$$A^\dagger=X_0(I-R_0)^{-1}=X_0\sum_{i=0}^{\infty}R_0^i, \tag{7}$$

which completes the proof. □

2.2. Basic elements of interval analysis

An interval $[x]$ is defined as a set of real numbers given by

$$[x]=[\underline{x},\overline{x}]=\{x:\underline{x}\le x\le\overline{x},\quad \underline{x},\overline{x},x\in\mathbb{R}\}.$$

The midpoint, width and the radius of $[x]$ are defined, respectively, as

$$m([x])=\frac{1}{2}(\overline{x}+\underline{x}),\quad w([x])=(\overline{x}-\underline{x}),\quad r([x])=\frac{1}{2}w([x])=\frac{1}{2}(\overline{x}-\underline{x}).$$

Thus,

$$[x] = [m([x]) - r([x]), m([x]) + r([x])] = m([x]) + [-r([x]), r([x])].$$

The absolute value of $[x]$ is defined as $|[x]| = \max\limits_{y\in[x]} |y| = \max\{|\underline{y}|, |\overline{y}|\}$ and the width of $[x]$ can be rewritten as $w([x]) = \max\limits_{y,z\in[x]} |y - z| = |[x] - [x]|$. The set of all real interval numbers are denoted by the symbol $\mathbb{IR}$. Since any real number a can be seen as an interval number $[a] = [a, a]$, it follows that $\mathbb{R} \subset \mathbb{IR}$. For further details, one can refer to [1,2].

Definition 2.1. For $[x],\ [y] \in \mathbb{IR}$ the relation $[x] \cap [y] = \{c : c \in [x],\ c \in [y]\}$ denotes the set theoretic intersection of two intervals. The result of this operation is in $\mathbb{IR}$ if and only if the set theoretic intersection is nonempty. Then it follows that $[x] \cap [y] = [\max\{\underline{x}, \underline{y}\},\ \min\{\overline{x}, \overline{y}\}]$.

Definition 2.2. A sequence of intervals $\{[x]^{(k)} = [\underline{x}^{(k)}, \overline{x}^{(k)}]\}$ converges to an interval $[x]$ iff the sequence of bounds of the individual members of the sequence converges to the corresponding bounds of $[x] = [\underline{x}, \overline{x}]$. Thus $\lim\limits_{k\to\infty} [x]^{(k)} = [x] \iff \left(\lim\limits_{k\to\infty} \underline{x}^{(k)} = \underline{x} \text{ and } \lim\limits_{k\to\infty} \overline{x}^{(k)} = \overline{x}\right)$.

These concepts can be extended componentwise to vectors and matrices.

Definition 2.3 [2]**.** An interval matrix $[A]$ of dimension $m \times n$ is defined as

$$[A] = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} : \underline{A} \le A \le \overline{A}\},$$

where, $\underline{A} \le \overline{A}$ and $\underline{A},\ \overline{A} \in \mathbb{R}^{m\times n}$ and the partial order is componentwise.

The *midpoint* $m([A])$ and the *width* $w([A])$ of an interval matrix $[A]$ are defined as

$$m([A]) = \frac{1}{2}(\overline{A} + \underline{A}), \quad w([A]) = \overline{A} - \underline{A}.$$

The interval matrix $[A]$ can also be written as

$$[A] = [\underline{A}, \overline{A}] = m([A]) + \frac{1}{2}[-w([A]), w([A])]. \tag{8}$$

The set of all $m \times n$ real interval matrices is denoted as $\mathbb{IR}^{m\times n}$.

Definition 2.4. The set theoretic intersection $[A] \cap [B] = \{C : C \in [A], C \in [B]\}$ denotes the intersection of $[A] = ([a_{ij}]) \in \mathbb{IR}^{m\times n}$ and $[B] = (b_{ij}) \in \mathbb{IR}^{m\times n}$. The intersection of two interval matrices $[A]$ and $[B]$ is in $\mathbb{IR}^{m\times n}$ if and only if the set theoretic intersection is nonempty. In this case $[A] \cap [B] = \left([a_{ij}] \cap [b_{ij}]\right)$, where $[a_{ij}] \cap [b_{ij}]$, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, is formed according to Definition 2.1.

Definition 2.5. A sequence $[A]^{(k)} = \left([a_{ij}]^{(k)}\right)$ of $m \times n$ interval matrices converges to $[A]$ if and only if the following equivalence is satisfied:

$$\lim_{k\to\infty} [A]^{(k)} = [A] \iff \lim_{k\to\infty} [a_{ij}]^{(k)} = [a_{ij}], \quad i = 1, 2, \ldots, m,\ \ j = 1, 2, \ldots, n.$$

Definition 2.6. An interval iterative method generating the sequence of matrices $\{[X]^{(k)}\}$ is said to be of pth order of convergence, $p > 1$, if there exists a positive constant M such that

$$\|w([X]^{(k+1)})\| \le M\|w([X]^{(k)})\|^p,\ \ k = 0, 1, \ldots$$

Lemma 2.3. *Every sequence of $m \times n$ interval matrices $[A]^{(k)} = \left([a_{ij}]^{(k)}\right)$ which satisfies*

$$[A]^{(0)} \supseteq [A]^{(1)} \supseteq [A]^{(2)} \supseteq \ldots$$

converges to an interval matrix $[A] = ([a_{ij}])$ whose entries are defined by

$$[a_{ij}] = \bigcap_{k=0}^{\infty} [a_{ij}]^{(k)}, \quad i = 1, 2, \ldots, m,\ \ j = 1, 2, \ldots, n.$$

Now, we restate the interval iterative method, originated in [24], for computing $A^{\dagger}$ of a full row (or full column) rank matrix.

Lemma 2.4 [24]**.** *Let $0 \ne A \in \mathbb{R}^{m\times n}$ $(m \le n)$ and the rank of A be m. Let the initial approximation $[X]^{(0)}$ and its residual $R_0 = I - A\,m([X]^{(0)})$ satisfy*

$$m([X]^{(0)}) \in R(A^{\mathrm{T}}, A^{\mathrm{T}}) = \{X : X = A^{\mathrm{T}}BA^{\mathrm{T}}, B \in \mathbb{R}^{m\times n}\} \tag{9}$$

and

$$\rho(I - A\,m([X]^{(0)})) < 1, \tag{10}$$

where $m([X]^{(0)})$ denotes the midpoint matrix of $[X]^{(0)}$. Then the sequence

$$[X]^{(k+1)} = m([X]^{(k)}) \sum_{i=0}^{p-2} \left(I - A\, m([X]^{(k)})\right)^i + [X]^{(k)} \left(I - A\, m([X]^{(k)})\right)^{p-1}, \quad k = 0, 1, \ldots \tag{11}$$

converges to $A^\dagger$ as $k \to \infty$. It is assumed that $[X]^{(k)} \in \mathbb{IR}^{n\times m}$, $(k = 0, 1, \ldots)$ and $p > 1$ is a natural number. Also, $w([X]^{(k)})$ satisfies the following inequality

$$\left\| w([X]^{(k+1)}) \right\| \le M \left\| w([X]^{(k)}) \right\|^p, \quad M \ge 0. \tag{12}$$

3. Proposed iterative method for computing $A^\dagger$

As was mentioned, the convergence of the hyperpower iterations (4) is significantly affected by roundoff errors caused due to the use of finite precision arithmetic. Motivated by this fact, our aim is to find appropriate error bounds for the Moore-Penrose inverse in each step of the iteration (4), and then generate a sequence of inclusion interval matrices containing $A^\dagger$ and converging to it in the sense proposed in Lemma 2.3.

Based on Lemma 2.2, the error of the initial approximation X_0 can be approximated in terms of the spectral norm $\|A^\dagger - X_0\|$ as

$$\begin{aligned} \|A^\dagger - X_0\| &= \left\| X_0 \sum_{i=0}^{\infty} R_0^i - X_0 \right\| \\ &= \|X_0 R_0 \sum_{i=0}^{\infty} R_0^i\| \\ &= \|X_0 R_0 (I - R_0)^{-1}\| \\ &\le \frac{\|X_0\|\,\|R_0\|}{1 - \|R_0\|}, \end{aligned} \tag{13}$$

since $\|(I - R_0)^{-1}\| \le \frac{1}{1-\|R_0\|}$ for $\rho(R_0) < 1$. As an interval matrix represents a compact set in $\mathbb{C}^{n\times m}$, we can bound any ball with respect to the Frobenius norm by an interval matrix. Thus we require to express the bound (13), given by means of the spectral radius norm, in terms of the Frobenius norm. The relation between the Frobenius norm and the spectral radius norm of an arbitrary matrix T is given by the inequality $\|T\|_F \le \sqrt{s}\,\|T\|$, where s is the rank of T. If r denotes the rank of A and $l = \min\{m, n\}$ then

$$\begin{aligned} \|A^\dagger - X_0\|_F &\le \sqrt{r}\,\|A^\dagger - X_0\| \\ &\le \sqrt{l}\,\|A^\dagger - X_0\| \qquad (\text{since } r \le l = \min\{m, n\}) \\ &\le \sqrt{l}\,\frac{\|X_0\|\,\|R_0\|}{1 - \|R_0\|}. \end{aligned} \tag{14}$$

Now, take $\gamma_0 = \sqrt{l}\,\frac{\|X_0\|\,\|R_0\|}{1-\|R_0\|}$ and denote by $B(X_0, \gamma_0)$ the ball in $\mathbb{C}^{n\times m}$ with the center X_0 and of radius γ_0. Then, according to (14), it follows that $A^\dagger \in B(X_0, \gamma_0)$. We now construct the initial interval matrix $[X]^{(0)} \in \mathbb{IC}^{n\times m}$ defined by

$$[X]^{(0)} = X_0 + [-\gamma_0 E, \gamma_0 E],$$

where E is the $n \times m$ matrix whose all entries are one. Then $A^\dagger \in B(X_0, \gamma_0) \subset [X]^{(0)}$. To get the next inclusion interval matrix $[X]^{(1)}$, it is sufficient to first compute X_1 according to (4) and then construct $[X]^{(1)}$ in a similar manner as it was done for $[X]^{(0)}$. More precisely $[X]^{(1)}$ is given by

$$[X]^{(1)} = X_1 + [-\gamma_1 E, \gamma_1 E],$$

where γ_1 and $B(X_1, \gamma_1)$ are also computed in the similar manner as it is done for X_0. This process then continues and gives a sequence of inclusion interval matrices containing $A^\dagger$. A formalization of the above investigation is stated in Theorem 3.1.

Remark 3.1. One of the crucial task is to find the quantity $\|R\|_0$. From basic theory of generalized inverses [3], we have $\|R_0\| = \max_i\{|1 - \alpha\lambda_i(AA^*)|\}$, where, $\lambda_i(AA^*)$ denotes the non-zero eigenvalues of AA^*.

Theorem 3.1. *Let $A \in \mathbb{C}_r^{m\times n}$ be a nonzero matrix of rank r, $p > 1$ be a natural number, and let E be the $n \times m$ matrix whose entries are all equal to one. Let the initial approximation X_0 and its residual $R_0 = P_{\mathcal{R}(A)} - AX_0$ (or $R_0 = P_{\mathcal{R}(A^*)} - X_0A$) satisfy*

$$X_0 \in \mathcal{R}(A^*, A^*) = \{X : X = A^*BA^*, B \in \mathbb{C}^{m\times n}\}$$

and

$$\rho(R_0) < 1 \ (\text{i.e., } \ \rho(P_{\mathcal{R}(A)} - AX_0) < 1 \text{ or } \rho(P_{\mathcal{R}(A^*)} - X_0A) < 1).$$

Then the sequence of interval matrices $\{[X]^{(k)}\}$ which are defined starting from

$$[X]^{(0)} = X_0 + [-\gamma_0 E, \gamma_0 E], \quad \gamma_0 = \sqrt{l}\,\frac{\|X_0\|\,\|R_0\|}{1-\|R_0\|}, \quad l = \min\{m, n\}$$

and updated by the iterative rules

$$\begin{aligned} X_{k+1} &= X_k \sum_{i=0}^{p-1} (I - AX_k)^i = \left[\sum_{i=0}^{p-1} (I - AX_k)^i\right] X_k \\ \gamma_{k+1} &= \sqrt{l}\,\frac{\|X_k\|\,\|R_0\|^{p^{k+1}}}{1-\|R_0\|^{p^k}} \\ [X]^{(k+1)} &= X_{k+1} + [-\gamma_{k+1}E, \gamma_{k+1}E], \quad k = 0, 1, \ldots \end{aligned} \tag{15}$$

contains $A^\dagger$ and converges to it. Also, the following relation holds for the iteration (15):

$$\|w([X]^{(k+1)})\| \le M \|w([X]^{(k)})\|^p, \quad 0 \le M < \infty. \tag{16}$$

Proof. We first prove that each interval matrix from the iterations (15) contains $A^\dagger$. According to the hyperpower iterative rule (4), it is clear that each X_k for $k = 0, 1, \ldots,$ satisfies $X_k \in \mathcal{R}(A^*, A^*)$ and $\rho(R_k) < 1$, where $R_k = P_{\mathcal{R}(A)} - AX_k$. This result in conjunction with Lemma 2.2 gives

$$A^\dagger = X_k (I - R_k)^{-1} = X_k \sum_{i=0}^{\infty} R_k^i. \tag{17}$$

Further, using $X_k(I - AX_k) = X_k T_k = X_k R_k$, where $T_k = I - AX_k$, one can conclude

$$\begin{aligned} \|A^\dagger - X_{k+1}\| &= \left\| X_k \sum_{i=0}^{\infty} R_k^i - X_k \sum_{i=0}^{p-1} (I - AX_k)^i \right\| \\ &= \left\| X_k \sum_{i=0}^{\infty} R_k^i - X_k \sum_{i=0}^{p-1} R_k^i \right\| \\ &= \left\| X_k \sum_{i=p}^{\infty} R_k^i \right\| \\ &= \left\| X_k R_k^p \sum_{i=0}^{\infty} R_k^i \right\| \\ &= \left\| X_k R_k^p (I - R_k)^{-1} \right\| \\ &\le \frac{\|X_k\|\,\|R_k^p\|}{1-\|R_k\|}. \end{aligned} \tag{18}$$

On the other hand, the iterative method of type (4) satisfies $R_k = R_{k-1}^p = \ldots = R_0^{p^k}$, which implies $\|R_k\| \le \|R_0\|^{p^k}$ and $\|R_k^p\| \le \|R_0\|^{p^{k+1}}$. Hence, from (18), it follows that

$$\|A^\dagger - X_{k+1}\| \le \frac{\|X_k\|\,\|R_0\|^{p^{k+1}}}{1-\|R_0\|^{p^k}}. \tag{19}$$

A transformation of (19) into the Frobenius matrix norm gives

$$\|A^\dagger - X_{k+1}\|_F \le \sqrt{l}\,\frac{\|X_k\|\,\|R_0\|^{p^{k+1}}}{1-\|R_0\|^{p^k}} = \gamma_{k+1}. \tag{20}$$

Then $\{\gamma_{k+1}\}$ is a sequence of positive real numbers. Clearly, as a consequence of (20), if $B(X_{k+1}, \gamma_{k+1})$ is the ball in $\mathbb{C}^{n\times m}$ with center at X_{k+1} and of radius γ_{k+1}, then $A^\dagger \in B(X_{k+1}, \gamma_{k+1})$. Now, the interval matrix at the $(k+1)$st iterate is given as

$$[X]^{(k+1)} = X_{k+1} + [-\gamma_{k+1}E, \gamma_{k+1}E].$$

This implies $A^\dagger \in B(X_{k+1}, \gamma_{k+1}) \subset [X]^{(k+1)}$, for $k = 0, 1, \ldots$ Thus, each interval matrix of the sequence $\{[X]^{(k)}\}$ contains $A^\dagger$.

It now remains to show that $[X]^{(k)} \to A^\dagger$ as $k \to \infty$. For this purpose, it is necessary to show that the width $w([X]^{(k)})$ of the interval matrix $[X]^{(k)}$ satisfies $w([X]^{(k)}) \to 0$ as $k \to \infty$. According to the definition of $[X]^{(k)}$, the width $w([X]^{(k)})$ satisfies $w([X]^{(k)}) = 2\gamma_k E$. On the other hand, since $\|R_0\| = \rho(R_0) < 1$, it follows that $\|R_0\|^{p^k} \to 0$ and further $\gamma_k \to 0$ as $k \to \infty$. This gives $w([X]^{(k)}) \to 0$ as $k \to \infty$. Now to prove (16), let us consider the limit

$$\lim_{k\to\infty}\frac{\left\|w([X]^{(k+1)})\right\|}{\left\|w([X]^{(k)})\right\|^p}=\lim_{k\to\infty}\frac{2\sqrt{l}\,\frac{\|X_k\|\|R_0\|^{p^{k+1}}}{1-\|R_0\|^{p^k}}\|E\|}{\left(2\sqrt{l}\,\frac{\|X_{k-1}\|\|R_0\|^{p^k}}{1-\|R_0\|^{p^{k-1}}}\|E\|\right)^p}$$

$$=\frac{1}{(2\sqrt{l}\,\|E\|)^{p-1}}\lim_{k\to\infty}\frac{\|X_k\|}{\|X_{k-1}\|^p}\frac{\left(1-\|R_0\|^{p^{k-1}}\right)^p}{1-\|R_0\|^{p^k}}. \tag{21}$$

Since $X_k, X_{k-1}\to A^\dagger$ as $k\to\infty$ and $\|R_0\|^{p^k}, \|R_0\|^{p^{k-1}}\to 0$, from (21) the following holds:

$$\lim_{k\to\infty}\frac{\left\|w([X]^{(k+1)})\right\|}{\left\|w([X]^{(k)})\right\|^p}=\frac{1}{\left(2\sqrt{l}\,\|E\|\|A^\dagger\|\right)^{p-1}}=M<\infty. \tag{22}$$

Thus, $\lim_{k\to\infty}\frac{\|w([X]^{(k+1)})\|}{\|w([X]^{(k)})\|^p}$ is bounded i.e. $\frac{\|w([X]^{(k+1)})\|}{\|w([X]^{(k)})\|^p}\le M$ for large k and $M>0$. Hence, (16) is verified. Similarly, the result for the case $R_0=P_{\mathcal{R}(A^*)}-X_0A$ can be verified. □

Since the sequence of interval matrices from (15) always contains $A^\dagger$, according to Theorem 3.1 the improved sequence of new iterates can be formed by taking the intersection of the new iterate with the previous iterate. Thus, the $(k+1)$th iterate computed using the iterative rule (15) is equal to $[Y]^{(k+1)}=X_{k+1}+[-\gamma_{k+1}E,\gamma_{k+1}E]$. Then the new $(k+1)$th iterate is defined as $[X]^{(k+1)}=[Y]^{(k+1)}\cap[X]^{(k)}$.

This gives a nested sequence of interval matrices

$$[X]^{(0)}\supseteq[X]^{(1)}\supseteq[X]^{(2)}\supseteq\ldots\supseteq[X]^{(k)}\supseteq\ldots$$

each containing $A^\dagger$ and by Lemma 2.3 converging to it as $k\to\infty$.

Algorithm 1 describes the computational steps of the method. It is worth mentioning that the formula for computing $\|R_0\|$ (in line 5 of the Algorithm 1) requires to compute the real value $\max_i\{|1-\alpha\lambda_i(AA^*)|\}$, where, $\lambda_i(AA^*)$ denotes the non-zero eigenvalues of AA^*, sorted as

$$\lambda_{\max}(AA^*)=\lambda_1(AA^*)\ge\lambda_2(AA^*)\ge\ldots\ge\lambda_r(AA^*)>0, 0=\lambda_{r+1}(AA^*)=\ldots=\lambda_m(AA^*). \tag{23}$$

This step requires the knowledge of the matrix rank r.

Algorithm 1

1: Inputs: Given matrix A and the maximal number of iterations, denoted by *maxit*;
2: Compute:
3: $0<\alpha<\frac{2}{\lambda_{\max}(AA^*)}$;
4: Set $X_0=\alpha A^*$;
5: $\|R_0\|=\max_i\{|1-\alpha\lambda_i(AA^*)|\}$, where, $\lambda_i(AA^*)$ denotes the non-zero eigenvalues of AA^*;
6: $\gamma_0=\sqrt{l}\,\frac{\|X_0\|\|R_0\|^2}{1-\|R_0\|}$;
7: $[X]^{(0)}=X_0+[-\gamma_0E,\gamma_0E]$;
8: **for** $k=0,1,\ldots,$ maxit **do**
9: {
10: $X_{k+1}=X_k\sum_{i=0}^{p-1}(I-AX_k)^i$;
11: $\gamma_{k+1}=\sqrt{l}\,\frac{\|X_k\|\|R_0\|^{p^{k+1}}}{1-\|R_0\|^{p^k}}$;
12: $[Y]^{(k+1)}=X_{k+1}+\left[-\gamma_{k+1}E,\gamma_{k+1}E\right]$;
13: $[X]^{(k+1)}=[Y]^{(k+1)}\cap[X]^{(k)}$;
14: **if** $\gamma_{k+1}\ge\gamma_k$ **then**
15: **break**
16: }

Algorithm 2 describes how the minimum nonzero eigenvalue of AA^* can be computed up to a tolerance ϵ.

Remark 3.2. There is an essential difference between the iterations (11) from [24] and the iterations (15) defined in our method. Namely, iterations (11) are defined using the midpoint matrix $m([X]^{(k)})$ of $[X]^{(k)}$ and principles of the interval analysis. On the other hand, the iterations (15) are based on the estimation of the rough matrix $[X]^{(k+1)}$ which contains the

Algorithm 2

Inputs: Given matrix AA^* and the tolerance ϵ;
Compute:
$\lambda_{\max}(AA^*)$;
Set $\mu = \lambda_{\max}(AA^*) + 1$;
$B = AA^* + \mu I$;
$j = 1$;
$\Lambda_j = 0$;
while $\Lambda_j(1) < \epsilon$ **do**
{
$\Lambda_j = [\lambda_{m-j+1}(B)\, \lambda_{m-j+2}(B)\, \ldots\, \lambda_m(B)]^T$;
$\Lambda_j = \Lambda_j - \mu$;
$j = j + 1$;
}
Nonzero $\lambda_{\min}(AA^*) = \Lambda_j(1)$;

Moore-Penrose inverse. Moreover, $[X]^{(k+1)}$ is generated using the error bounds of the hyperpower iterative method of arbitrary order with respect to the Moore-Penrose inverse.

Remark 3.3. The Algorithm 1 can be modified by finding an α such that

$$\|R_0\| = \max_{\alpha}\{|1-\alpha\lambda_1(AA^*)|\},\ 0 < \alpha < \frac{2}{\lambda_{\max}(AA^*)}.$$

Such an α will avoid the computation of the minimum nonzero eigenvalue of AA^*, thus avoiding the use of Algorithm 2. However, the following arguments justify that such an α will not provide any benefits. The relation

$$\|R_0\| = \max_{i}\{|1-\alpha\lambda_i(AA^*)| : i = 1, 2, \ldots, r\}$$

always holds for such an α, where (23) are the eigenvalues of AA^*. In order to do so, first we have to find the ranges of α in which $|1-\alpha\lambda_1(AA^*)|$ dominates $|1-\alpha\lambda_i(AA^*)|$ for $i = 1, 2, \ldots, r$. For this purpose, let us divide the range of α, i.e., $0 < \alpha < \frac{2}{\lambda_{\max}(AA^*)}$ in two parts.

Case 1: $0 < \alpha < \frac{1}{\lambda_{\max}(AA^*)}$. Take α as $\alpha = \frac{\theta}{\lambda_{\max}(AA^*)}$, where $0 < \theta < 1$. This gives

$$|1-\alpha\lambda_i(AA^*)| = |1 - \frac{\theta}{\lambda_{\max}(AA^*)}\lambda_i(AA^*)|,\ i = 1, 2, \ldots, r.$$

Now $0 < \theta < 1$ and $0 < \frac{\lambda_i(AA^*)}{\lambda_{\max}(AA^*)} \le 1$ for each $i = 1, 2, \ldots, r$ implies $0 < 1-\alpha\lambda_i(AA^*) < 1$. Therefore $|1-\alpha\lambda_i(AA^*)| = 1-\alpha\lambda_i(AA^*)$. Also due to the ordering of the eigenvalues $\lambda_i(AA^*)$, it follows that $1-\alpha\lambda_1(AA^*) \le 1-\alpha\lambda_2(AA^*) \le \ldots \le 1-\alpha\lambda_r(AA^*)$. Thus, in this range we can not find such α.

Case 2: $\frac{1}{\lambda_{\max}(AA^*)} < \alpha < \frac{2}{\lambda_{\max}(AA^*)}$. This means that α can be expressed as $\alpha = \frac{1+\theta}{\lambda_{\max}(AA^*)}$, where $0 < \theta < 1$. Now, $\alpha\lambda_1(AA^*) = \frac{1+\theta}{\lambda_{\max}(AA^*)}\lambda_1(AA^*) = 1+\theta > 1$. This case can also be divided into two subcases.

First subcase is if $\alpha\lambda_r(AA^*) > 1$. Then $\alpha\lambda_i(AA^*) > 1$, $i = 1, 2, \ldots, r$, which further implies $1-\alpha\lambda_i(AA^*) < 0$, $i = 1, \ldots, r$. Now $|1-\alpha\lambda_1(AA^*)|$ dominates the other terms $|1-\alpha\lambda_i(AA^*)|$ if

$$\begin{aligned}
&|1-\alpha\lambda_1(AA^*)| \ge |1-\alpha\lambda_i(AA^*)|\\
&\Rightarrow \left|1 - \frac{1+\theta}{\lambda_{\max}(AA^*)}\lambda_1(AA^*)\right| \ge \left|1 - \frac{1+\theta}{\lambda_{\max}(AA^*)}\lambda_i(AA^*)\right|\\
&\Rightarrow |-\theta| \ge \left|1 - (1+\theta)\frac{\lambda_i(AA^*)}{\lambda_1(AA^*)}\right|\\
&\Rightarrow \theta \ge -\left(1 - (1+\theta)\frac{\lambda_i(AA^*)}{\lambda_1(AA^*)}\right)\\
&\Rightarrow 1+\theta \ge (1+\theta)\frac{\lambda_i(AA^*)}{\lambda_1(AA^*)},\ i = 1, 2, \ldots, r.
\end{aligned}$$

As $0 < \frac{\lambda_i(AA^*)}{\lambda_1(AA^*)} \le 1$ holds for each $i = 1, 2, \ldots, r$, the last relation holds for all $0 < \theta < 1$.

Second subcase is the situation

$$\alpha\lambda_1(AA^*) \ge \alpha\lambda_2(AA^*) \ge \ldots \ge \alpha\lambda_q(AA^*) > 1 > \alpha\lambda_{q+1}(AA^*) \ge \ldots \ge \alpha\lambda_r(AA^*)$$

for some $1 < q < r$. Then by the first subcase

$$|1-\alpha\lambda_1(AA^*)| \ge |1-\alpha\lambda_i(AA^*)|,\ i = 1, 2, \ldots, q.$$

Now $0 < 1-\alpha\lambda_{q+1}(AA^*) \le \ldots \le 1-\alpha\lambda_r(AA^*) < 1$. Thus, $|1-\alpha\lambda_1(AA^*)|$ dominates $1-\alpha\lambda_r(AA^*)$ if

$$\begin{aligned}
&|1-\alpha\lambda_1(AA^*)| \ge 1-\alpha\lambda_r(AA^*)\\
&\Rightarrow \theta \ge 1-(1+\theta)\frac{\lambda_r(AA^*)}{\lambda_1(AA^*)}\\
&\Rightarrow \theta \ge \frac{\lambda_1(AA^*)-\lambda_r(AA^*)}{\lambda_1(AA^*)+\lambda_r(AA^*)}.
\end{aligned}$$

Therefore $|1-\alpha\lambda_1(AA^*)|$ dominates the other terms if

$$\alpha \ge \frac{1+\frac{\lambda_1(AA^*)-\lambda_r(AA^*)}{\lambda_1(AA^*)+\lambda_r(AA^*)}}{\lambda_1(AA^*)} = \frac{2}{\lambda_1(AA^*)+\lambda_r(AA^*)}.$$

This implies that for this case also $\lambda_r(AA^*)$ will be required. This value can be computed easily if the rank r of A is known beforehand since the rth eigenvalue $\lambda_r(AA^*)$ gives the minimal nonzero eigenvalue. If $r = \mathrm{rank}(A)$ is not known, then Algorithm 2 can be used to compute the nonzero minimal eigenvalue. It is important to mention that the quantity Λ_j in Line 10 of Algorithm 2 can be calculated by the MATLAB command `eigs`.

4. Numerical examples

Firstly, the efficacy of the proposed iterative scheme (PIS) (15) of order $p = 2$ is demonstrated on a real 5×6 matrix of rank 3 in Example 4.1. Pseudoinverses of randomly generated matrices of full rank are computed with PIS (15) using $p = 2$ in Example 4.2. The results obtained are compared with those obtained from the Schultz type iterative scheme (SIS) (4) and the interval iterative method (IIM) from [24]. All the computations are done in INTLAB toolbox of MATLAB R2012a [14]. The error in kth step is taken as $error_k = \max\{\|AY_kA-A\|, \|Y_kAY_k-Y_k\|, \|(AY_k)^*-AY_k\|, \|(Y_kA)^*-Y_kA\|\}$, where, Y_k approximates $A^\dagger$. Examples 4.3 and 4.4, compare the number of iterations needed to achieve convergence of Algorithm 1 in the cases $p = 2$ and $p = 3$. The stopping criterion used in numerical examples is $error_{k+1} \ge error_k$. The number of iterations and computational times required by PIS of different orders on random rank deficient matrices are studied in Example 4.5.

Example 4.1. Let us consider the matrix

$$A = \begin{pmatrix} 1.4566 & -3.2396 & -0.1088 & -0.2435 & 2.1872 & -0.4288\\ -0.2590 & -1.4605 & 2.4233 & -1.1504 & -1.4567 & -0.3618\\ 1.3187 & -3.5390 & 1.4690 & -2.4546 & 0.3003 & -1.0545\\ 1.2127 & 0.4126 & -1.4065 & -3.5733 & -0.3133 & -1.1694\\ 0.7516 & -4.6450 & 2.2216 & 0.8493 & 1.5396 & -0.0856 \end{pmatrix}$$

of $\mathrm{rank}(A) = 3$. The Moore-Penrose generalized inverse computed by the hyperpower iterative scheme (4) is

$$X = \begin{pmatrix} 0.0627 & -0.0544 & 0.0215 & 0.0565 & 0.0015\\ -0.0572 & -0.0136 & -0.0489 & 0.0163 & -0.0787\\ -0.0910 & 0.1549 & 0.0358 & -0.0753 & 0.0445\\ 0.0138 & -0.0520 & -0.0841 & -0.1391 & 0.0532\\ 0.1199 & -0.1394 & -0.0261 & 0.0050 & 0.0335\\ -0.0076 & -0.0078 & -0.0293 & -0.0468 & 0.0112 \end{pmatrix}.$$

Since A is not of full row rank, the interval iterative scheme (IIM) (11) is not applicable for this example.

The values of γ_k obtained by PIS (15) for $p = 2$ during 9 iterations are

$$\gamma_1 = 1.7202, \gamma_2 = 0.9177, \gamma_3 = 0.4684, \gamma_4 = 0.1693, \gamma_5 = 0.0263, \gamma_6 = 8.0488e-04,$$
$$\gamma_7 = 9.6050e-07, \gamma_8 = 1.4602e-12, \gamma_9 = 3.3828e-24.$$

These values provide a tight enclosure of the Moore-Penrose inverse of A. After the 9th iteration the approximate value of $A^\dagger$ is given by

$$m([X]^{(9)}) = \begin{pmatrix} 0.06272303613524 & -0.05443619766846 & 0.02153848196884 & 0.05650324033441 & 0.00150833517662\\ -0.05719393744124 & -0.01360766680803 & -0.04887265005667 & 0.01628369092325 & -0.07866575317945\\ -0.09095884617772 & 0.15493331516307 & 0.03578164105033 & -0.07525631928451 & 0.04454562010883\\ 0.01376235289933 & -0.05195759627286 & -0.08409904330419 & -0.13910945200531 & 0.05317147185141\\ 0.11994292152441 & -0.13939097908681 & -0.02611067564595 & 0.00495860705800 & 0.03350573965452\\ -0.00760882189236 & -0.00783766851728 & -0.02934674229509 & -0.04681868359121 & 0.01118626933438 \end{pmatrix}$$

and $w([X]^{(9)}) = \gamma_9 E$, where E is 6×5 matrix with entries are all equal to one.

Example 4.2. A random 500×600 full row rank matrix is used in this example. The left graphs in Fig. 1 give the comparison of the SIS (4) and lower bounds of PIS (15) in the case $p = 2$, for $no.of iterations(k)$ versus $\log_{10}(error_k)$. Similarly,

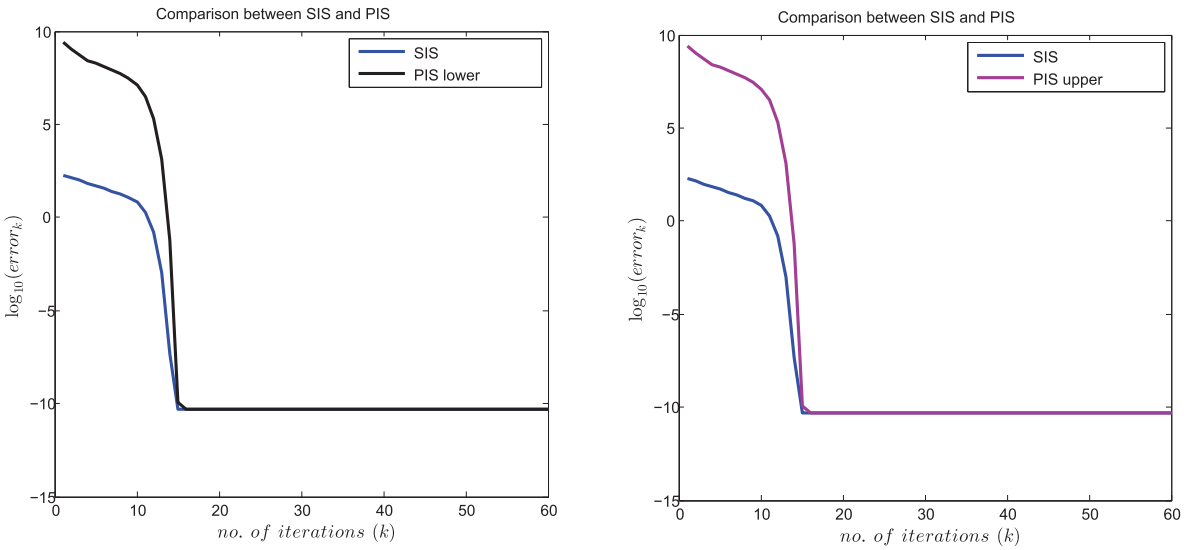

Fig. 1. *No. of iterations* (k) versus $\log_{10}(error_k)$ plot.

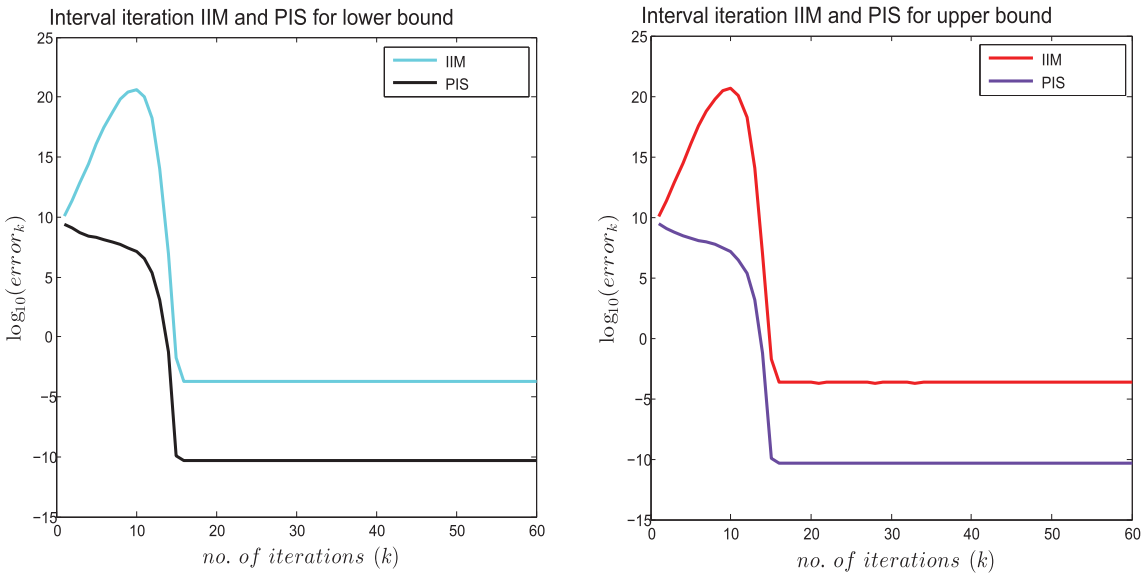

Fig. 2. *No. of iterations* (k) versus $\log_{10}(error_k)$ plot.

the right graphs in Fig. 1 illustrate the SIS (4) and upper bounds of PIS (15) in the case $p = 2$, for *no. of iterations* (k) versus $\log_{10}(error_k)$. The blue color in both left and right graphs is used to denote the error curve for SIS, while black and magenta color denotes the error curve for the lower and upper bound of PIS, respectively. From Fig. 1 it is observable that the proposed PIS method generates lower values $\log_{10}(error_k)$ even for their own lower bounds.

Further, the left (resp. right) graphs in Fig. 2 compare the IIM (11) and lower (resp. upper) bounds of PIS (15) for $p = 2$ with respect to *no. of iterations* (k) versus $\log_{10}(error_k)$. Here, sky blue and red color denote the error curves for IIM. One can easily see from Fig. 2 that our PIS method gives higher accuracy (lower error) for both lower bound and upper bounds with respect to IIS. Moreover, it is observed that errors generated by the IIM method goes up in the initial stage.

The method SIS is the classical iterative method for finding Moore-Penrose inverse. Therefore, it does not provide any error bound for the approximated solution. The IIM provides an interval method. It computes an interval matrix with very small radius which encloses the exact solution. The drawback of IIM is that it takes more time than SIS. The proposed iterative scheme (PIS) overcomes this problem by avoiding interval computations but providing an interval matrix with very small radius enclosing the exact solution. It is also clear from Fig. 2 that it gives better error estimates than IIM.

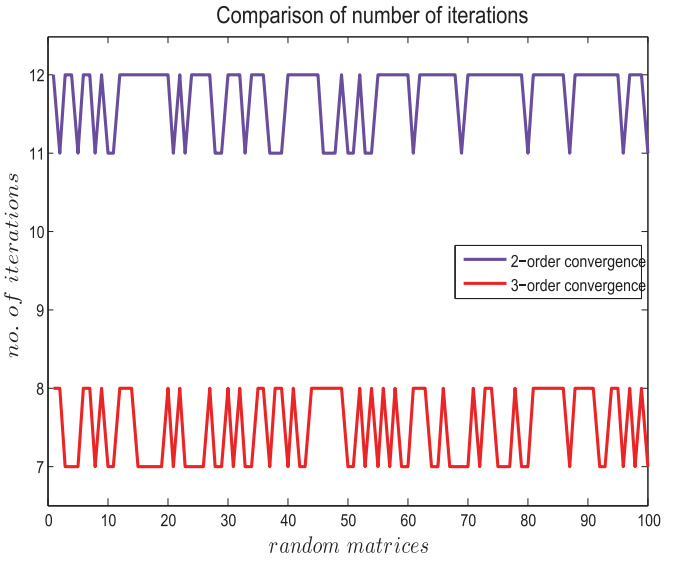

Fig. 3. Random full row rank matrices versus No. of iterations plot.

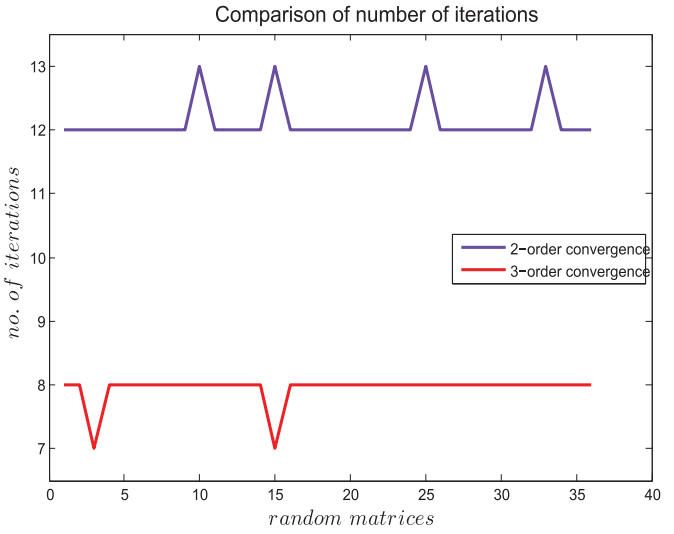

Fig. 4. Random rank deficient matrices versus No. of iterations plot.

Example 4.3. Let $A = randn(100, 200)$. Hundred randomly generated 100×200 matrices are tested by PIS (15) for $p = 2$ and $p = 3$ and comparisons of the number of iterations are plotted in Fig. 3. Magenta and red color in Fig. 3 denote the 2-order and 3-order of PIS, respectively. It is clear from the figure that the 3-order PIS takes a smaller number of iterations than the 2-order PIS.

A similar experiment for rank deficient matrices is done in the next example.

Example 4.4. Let $A = A_1A_2$ where, $A_1 = randn(100, 50)$ and $A_2 = randn(50, 200)$. Then A becomes a 100×200 matrix of $\text{rank}(A) = 50$. Thirty five matrices generated in such a way in MATLAB are tested by PIS (15) in the cases $p = 2$ and $p = 3$ and the numbers of iterations are plotted in Fig. 4. The same color scheme is followed as in the previous example. This example also justifies that 3-order PIS is faster than 2-order PIS.

Example 4.5 shows the difference in the number of iterations and time of PIS for matrices of larger dimensions.

Example 4.5. Here, fifty randomly generated matrices of different dimensions are considered and the variations in the number of iterations and computational times with the matrix dimensions is observed. Thirty random matrices are chosen for every dimensions. Matrix dimensions are constructed as $(\texttt{round}(3 + 0.008(j-1)^3), \texttt{round}(5 + 0.016(j-1)^3))$ for $j = 1, \ldots, 50$, where '`round`' is the `MATLAB` command used for rounding the floating point numbers to integers. The

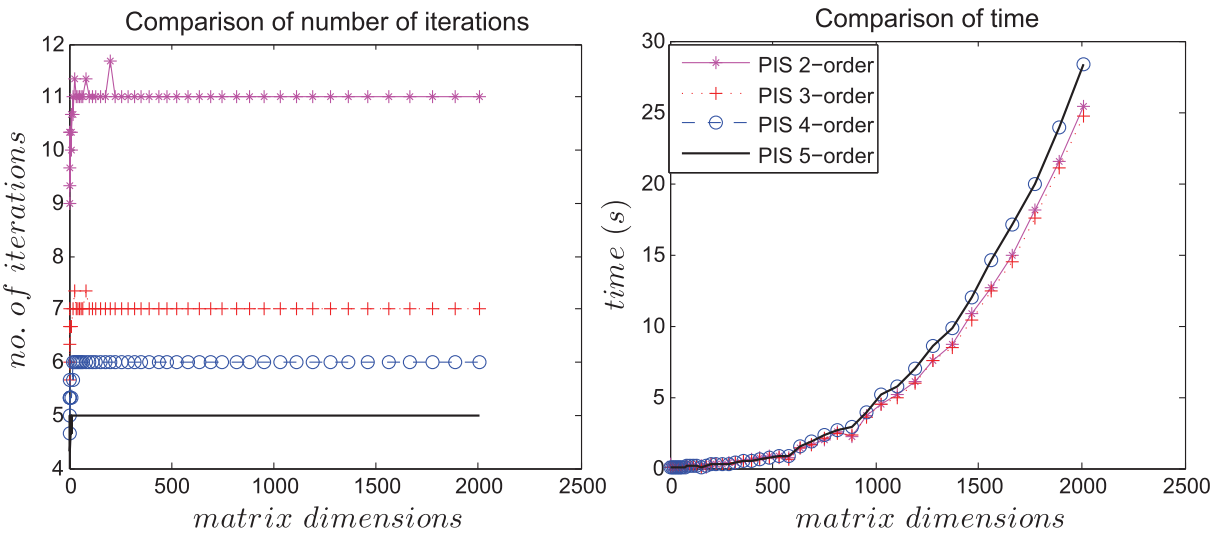

Fig. 5. Matrix dimensions versus No. of iterations/time plot.

Table 1
Comparison of number of iterations and time.

Matrix dimensions		PIS 2-order	PIS 3-order	PIS 4-order	PIS 5-order
114×226	No. of iterations	10	6	5	**4**
	Time (sec)	0.2831	0.2835	**0.2675**	0.2877
219×437	No. of iterations	12	8	**5**	**5**
	Time (sec)	0.7244	**0.6760**	0.7427	0.7339
376×751	No. of iterations	11	7	6	**5**
	Time (sec)	2.1343	**1.9940**	2.4470	2.3628
596×1190	No. of iterations	11	7	6	**5**
	Time (sec)	6.1456	**6.0516**	6.9064	7.2020
888×1774	No. of iterations	11	7	6	**5**
	Time (sec)	16.8652	**16.3591**	18.7999	19.9153

termination condition for different order PIS in kth step is $\gamma_k < 10^{-12}$. Two plots are given in Fig. 5 which compares the averages of the number of iterations and computational times with matrix dimensions for PIS (15) with $p = 2, 3, 4$ and 5. The color for the curves for 2, 3, 4, 5 -order PIS are respectively magenta, red, blue and black. It is clear from Fig. 5 (left) that for larger dimensional matrices, the number of iterations is inversely proportional to the order of PIS. On the other hand, Fig. 5 (right) shows that the number of iterations is inversely proportional to the order of PIS.

Table 1 provides the averages of a number of iterations and computational times measured in different order PIS for five sample matrices chosen from the above fifty set of random matrices. The best results in the table are marked in bold. According to results arranged in Table 1, higher order methods require smaller number of iterative steps. On the order hand, higher order PIS require greater CPU time. In general, the best order with respect to the CPU time is $p = 3$.

5. Conclusions

A new iterative scheme is proposed for computing the Moore-Penrose generalized inverse and its error bounds of a rectangular or singular real or complex matrix of arbitrary rank. The method requires that the rank of the matrix is known or computed beforehand. A convergence theorem is established. Contrary to the iterative method defined in [24], which holds only for full-column rank matrix, the proposed method is also applicable for a rank deficient matrix. Moreover, numerical examples involving randomly generated matrices are solved to demonstrate the efficacy of our approach. In comparison with the results obtained by Zhang et al. [24], it is found that our approach gives better accuracy.

References

[1] G. Alefeld, J. Herzberger, Introduction to Interval Computation, Academic Press, New York, London, 2012.
[2] G. Alefeld, G. Mayer, Interval analysis: theory and applications, J. Comput. Appl. Math. 121 (1–2) (2000) 421–464.
[3] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, Vol. 15., Springer Science & Business Media, 2003.
[4] A. Ben-Israel, A. Charnes, Contributions to the theory of generalized inverses, J. Soc. Ind. Appl. Math. 11 (1963) 667–699.
[5] A. Bojanczyk, N.J. Higham, H. Patel, Solving the indefinite least squares problem by hyperbolic QR factorization, SIAM J. Matrix Anal. Appl. 24 (2003) 914–931.

[6] H. Chen, Y. Wang, A family of higher-order convergent iterative methods for computing the Moore-Penrose inverse, Appl. Math. Comput. 218 (2011) 4012–4016.
[7] V.N. Katsikis, D. Pappas, Fast computing of the Moore-Penrose inverse matrix, Electron. J. Linear Algebra 17 (2008) 637–650.
[8] V.N. Katsikis, D. Pappas, A. Petralias, An improved method for the computation of the Moore-Penrose inverse matrix, Appl. Math. Comput. 217 (2011) 9828–9834.
[9] W. Li, 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) 3433–3442.
[10] I.V. Oseledets, E.E. Tyrtyshnikov, Approximate inversion of matrices in the process of solving a hypersingular integral equation, Zhurnal Vychislitelnoi Matematiki Matematicheskoi Fiziki 45 (2005) 315–326.
[11] M.D. Petković, P.S. Stanimirović, Iterative method for computing the Moore-Penrose inverse based on Penrose equations, J. Comput. Appl. Math. 235 (2011) 1604–1613.
[12] M.D. Petković, P.S. Stanimirović, Two improvements of the iterative method for computing Moore-Penrose inverse based on Penrose equations, J. Comput. Appl. Math. 267 (2014) 61–71.
[13] M.D. Petković, P.S. Stanimirović, Generalized matrix inversion is not harder than matrix multiplication, J. Comput. Appl. Math. 230 (2009) 270–282.
[14] S.M. Rump, INTLAB-interval laboratory, in: Developments in Reliable Computing, Springer, Netherlands, 1999, pp. 77–104.
[15] X. Sheng, Computation of weighted Moore-Penrose inverse through Gauss-Jordan elimination on bordered matrices, Appl. Math. Comput. 323 (2018) 64–74.
[16] S. Srivastava, D.K. Gupta, P.S. Stanimirovic, S. Singh, F. Roy, A hyperpower iterative method for computing the generalized Drazin inverse of Banach algebra element, Sadhana 42 (2017) 625–630.
[17] S. Srivastava, D.K. Gupta, Higher order iterations for Moore-Penrose inverses, J. Appl. Math. Inform. 32 (2014) 171–184.
[18] I.P. Stanimirović, M.B. Tasić, Computation of generalized inverses by using the LDL^* decomposition, Appl. Math. Lett. 25 (2012) 526–531.
[19] P.S. Stanimirović, S. Chountasis, D. Pappas, I. Stojanović, Removal of blur in images based on least squares solutions, Math. Methods Appl. Sci. 36 (2013) 2280–2296.
[20] P.S. Stanimirović, M.D. Petković, Gauss-Jordan elimination method for computing outer inverses, Appl. Math. Comput. 219 (2013) 4667–4679.
[21] F. Toutounian, A. Ataei, A new method for computing Moore-Penrose inverse matrices, J. Comput. Appl. Math. 228 (2009) 412–417.
[22] Y. Wei, On the perturbation of the group inverse and oblique projection, Appl. Math. Comput. 98 (1999) 29–42.
[23] L. Weiguo, L. Juan, Q. Tiantian, A family of iterative methods for computing Moore-Penrose inverse of a matrix, Linear Algebra Appl. 438 (2013) 47–56.
[24] X. Zhang, J. Cai, Y. Wei, Interval iterative methods for computing Moore-Penrose inverse, Appl. Math. Comput. 183 (2006) 522–532.