Journal of Computational and Applied Mathematics 267 (2014) 61–71

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Two improvements of the iterative method for computing Moore–Penrose inverse based on Penrose equations

Marko D. Petković*, Predrag S. Stanimirović

University of Niš, Faculty of Science and Mathematics, Višegradska 33, 18000 Niš, Serbia

ARTICLE INFO

Article history:
Received 9 August 2013
Received in revised form 21 January 2014

MSC:
15A09

Keywords:
Moore–Penrose inverse
Iterative methods
Convergence
Penrose equations

ABSTRACT

Two improvements of the iterative method for computing the Moore–Penrose inverse, introduced in Petković and Stanimirović (2011) are introduced. The first improvement defines new choice for the initial approximation and ensures better stability of the method. The second one defines its extension to the set of outer inverses with prescribed range and null space. Numerical examples for both of these improvements are presented.

1. Introduction

In the paper [1], we discussed the following iterative method for computing the Moore–Penrose inverse $X = A^\dagger$ of the matrix $A \in \mathbb{C}^{m\times n}$, based on the Penrose equations (2) and (4):

$$X_{k+1} = (1+\beta)X_k - \beta X_k A X_k, \qquad X_0 = \beta A^*. \tag{1.1}$$

Recall that the Moore–Penrose inverse $A^\dagger$ of the matrix A is the unique solution of the following matrix system of Penrose equations:

$$(1)\ AXA = A, \qquad (2)\ XAX = X, \qquad (3)\ (AX)^* = AX, \qquad (4)\ (XA)^* = XA.$$

Denote by $\lambda_1(M), \lambda_2(M), \ldots, \lambda_n(M)$ all eigenvalues of the matrix $M \in \mathbb{C}^{n\times n}$ and assume that $\lambda_i(M) \neq 0$ for $i = 1, 2, \ldots, r$ where $r = \operatorname{rank}(M)$. The method (1.1) is convergent for arbitrary $\beta \in (0, 1]$ such that

$$\max_{1\le i\le r} |1 - \beta\lambda_i(A^*A)| < 1, \tag{1.2}$$

where $r = \operatorname{rank}(A^*A) = \operatorname{rank}(A)$. It is shown that the method (1.1) has linear convergence for every $\beta \neq 1$ and quadratic for $\beta = 1$. It is also demonstrated that, under certain condition, the method can be unstable due to the accumulation of the roundoff errors (see Section 4 of the paper [1]).

Moreover, numerical examples indicating that the method (1.1) is usually unstable, especially for ill-conditioned matrices, are given in a recently published paper [2]. Testings were done on the matrices from the Matrix Computation Toolbox [3] and from Matrix Market [4].

* Corresponding author.
E-mail addresses: dexterofnis@gmail.com (M.D. Petković), pecko@pmf.ni.ac.rs (P.S. Stanimirović).

http://dx.doi.org/10.1016/j.cam.2014.01.034
0377-0427/© 2014 Elsevier B.V. All rights reserved.

The content of the paper is organized as follows. The second section shows an improvement of the method (1.1) in the choice of the initial approximation. It also provides a generalization of (1.1) to the set of outer inverses with prescribed range and null space. Numerical results are presented in the third section. Better numerical behavior of the modified method is demonstrated by means of calculating generalized inverses on the randomly generated matrices and ones from [3] and [4].

2. Improvement and generalization to outer inverses

Method (1.1) uses the same coefficient β both for the starting value and for the rule which defines computation of the next iterative step. Condition (1.2) may force the value β to be very small (for example 10^{-6}), which implies very slow convergence of the matrix sequence $(X_k)_{k\in\mathbb{N}_0}$ defined by (1.1). Our first improvement overcomes that drawback by observing that there is no need for using the same value β both for the initial value X_0 and the iterative step in (1.1). In such way, we give the following improvement:

$$X_{k+1} = (1+\beta)X_k - \beta X_k A X_k, \qquad X_0 = \alpha A^*, \quad \beta \in (0, 1], \tag{2.1}$$

where α, β are (non necessarily equal) real constants. It is not difficult to see that all properties of the method (1.1) from [1] remain valid also for the more general method (2.1).

The outer inverse of $A \in \mathbb{C}_r^{m\times n}$ with prescribed range T and null space S, denoted by $A^{(2)}_{T,S}$, satisfies the matrix equation $XAX = X$ and two additional properties: $\mathcal{R}(X) = T$ and $\mathcal{N}(X) = S$. The most important generalized inverses are particular appearances of outer inverses with prescribed range and null space. The Moore–Penrose inverse $A^\dagger$ and the weighted Moore–Penrose inverse $A^\dagger_{M,N}$, the Drazin inverse A^D and the group inverse $A^\#$ can be derived by means of appropriate choices of subspaces T and S:

$$\begin{aligned} &A^\dagger = A^{(2)}_{\mathcal{R}(A^*),\mathcal{N}(A^*)}, \qquad A^\dagger_{M,N} = A^{(2)}_{\mathcal{R}(A^\sharp),\mathcal{N}(A^\sharp)}, \\ &A^D = A^{(2)}_{\mathcal{R}(A^k),\mathcal{N}(A^k)}, \quad k \ge \operatorname{ind}(A), \qquad A^\# = A^{(2)}_{\mathcal{R}(A),\mathcal{N}(A)}, \quad \operatorname{ind}(A) = 1, \end{aligned} \tag{2.2}$$

where $A^\sharp = MAN^{-1}$ and $\operatorname{ind}(A)$ denotes the index of a square matrix A (see for example [5]).

Our motivation for further generalization can be described as follows. The choice of the initial iteration $X_0 = \alpha A^*$ in (2.1) directs the convergence $\lim_{k\to\infty} X_k = A^\dagger = A^{(2)}_{\mathcal{R}(A^*),\mathcal{N}(A^*)}$. According to (2.2), it is reasonable to expect that the different choices of X_0 will imply different convergence of the iterative process (2.1). Therefore, our further improvement is based on the usage of an appropriate matrix G in the definition of X_0.

Assume that $A \in \mathbb{C}^{m\times n}$ is a given matrix of rank r. We define the following generalization of the iterative process (2.1):

$$X_{k+1} = (1+\beta)X_k - \beta X_k A X_k, \qquad X_0 = \alpha G, \quad \beta \in (0, 1], \tag{2.3}$$

where α, β are real constants and $G \in \mathbb{C}^{n\times m}$ is a chosen matrix of rank $0 < s \le r$. A full-rank factorization of G in the form

$$G = UV, \quad U \in \mathbb{C}_s^{n\times s},\ V \in \mathbb{C}_s^{s\times m} \tag{2.4}$$

is considered.

Remark 2.1. In the case $\beta = 1$ the iterative process (2.3) produces well known generalization of the Schultz iterative method, intended for computation of outer inverses [6,7].

2.1. Properties of the method

Lemma 2.1. *Let $A \in \mathbb{C}_r^{m\times n}$ be given matrix and $G \in \mathbb{C}_s^{n\times m}$, $0 < s \le r$ is selected but fixed matrix. Let $(X_k)_{k\in\mathbb{N}_0}$ be the sequence defined by (2.3). If (2.4) is the full rank factorization of G such that the matrix VAU is invertible, then the outer inverse $X := A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V)} = A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)}$ exists and satisfies the following statements:*

$$XAX_k = X_k, \qquad X_k AX = X_k, \quad k \ge 0. \tag{2.5}$$

Proof. Using the main result from [8] we conclude the existence of the outer inverse

$$X := A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V)} = U(VAU)^{-1}V.$$

Further, $X_0 = \alpha G$ implies

$$\mathcal{R}(X_0) = \mathcal{R}(G) = \mathcal{R}(U), \qquad \mathcal{N}(X_0) = \mathcal{N}(G) = \mathcal{N}(V).$$

Now by using

$$\begin{aligned} X_{k+1} &= X_k\left((1+\beta)I_m - \beta AX_k\right) &\quad (2.6)\\ &= \left((1+\beta)I_n - \beta X_k A\right)X_k, &\quad (2.7) \end{aligned}$$

we have

$$\mathcal{R}(X_{k+1}) \subseteq \mathcal{R}(X_k), \qquad \mathcal{N}(X_{k+1}) \supseteq \mathcal{N}(X_k), \quad k \geq 0.$$

Therefore,

$$\mathcal{R}(X_k) \subseteq \mathcal{R}(G) = \mathcal{R}(U) \tag{2.8}$$

$$\mathcal{N}(X_k) \supseteq \mathcal{N}(G) = \mathcal{N}(V). \tag{2.9}$$

As usual, $P_{L,M}$ denotes the projector on L along M. Now, the statements in (2.5) follow from (2.8), the following known facts (cf. [5])

$$AX = P_{A\mathcal{R}(U),\mathcal{N}(V)}, \qquad XA = P_{\mathcal{R}(U),\left(A^*\mathcal{N}(V)^\perp\right)^\perp}$$

and well-known results: $P_{L,M}X_k = X_k$, if and only if $\mathcal{R}(X_k) \subseteq L$ and $X_kP_{L,M} = X_k$, if and only if $\mathcal{N}(X_k) \supseteq M$. □

Remark 2.2. Note that (2.6) and (2.7) are equivalent variants of the method (2.3). Iterative process (2.6) reduces the number of floating point operations in each iterative step in the case $m \leq n$. In the opposite case, $n < m$, the iterations (2.7) are more appropriate.

In the case $G = A^*$, the following analogous statement of Lemma 3.1 from [1] is satisfied.

Lemma 2.2 *([1, Lemma 3.1]). The sequence $(X_k)_{k\in\mathbb{N}_0}$ defined by (2.1) satisfies*

$$X_kA = (X_kA)^*, \qquad XAX_k = X_k, \qquad X_kAX = X_k, \quad k \geq 0. \tag{2.10}$$

2.2. Proof of the convergence

In the following theorem we prove the convergence of the iterative method (2.3) to $A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)}$. Particularly, we also consider the convergence of the method (2.1). For the sake of simplicity we use the notation $X := A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)}$. The residual (resp. error) matrices are denoted by $F_k := AX - AX_k$ (resp. $E_k := X - X_k$), for each $k \geq 0$.

Theorem 2.1. *Under the assumption of Lemma 2.1 the iterative method (2.3) converges to $X := A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)}$ if the condition*

$$\|F_0\| = \|AX - AX_0\| < 1, \quad 0 < \beta \leq 1 \tag{2.11}$$

is satisfied. The method has a linear convergence in the case $\beta \neq 1$, while for $\beta = 1$ its convergence is quadratic. The first-order and the second-order terms, corresponding to the error estimation of (2.3) are equal to:

$$error_1 = (1-\beta)E_k, \qquad error_2 = -\beta E_kAE_k. \tag{2.12}$$

Proof. Note that $F_kAX = (AX - AX_k)AX = AX - AX_k = F_k$ and also $AXF_k = F_k$. By some algebraic transformations we get

$$\begin{aligned} F_{k+1} &= AX - AX_{k+1} = AX - (1+\beta)AX_k + \beta(AX_k)^2 \\ &= AX - (1+\beta)(AX - F_k) + \beta(AX - F_k)^2 \end{aligned}$$

which further implies

$$F_{k+1} = (1-\beta)F_k + \beta F_k^2. \tag{2.13}$$

By applying an arbitrary matrix norm $\|\cdot\|$ on the both sides of the last identity, and taking into account that $\beta \in (0, 1]$, we attain

$$\|F_{k+1}\| \leq \|F_k\| \left(1 - \beta + \beta\|F_k\|\right). \tag{2.14}$$

Using the principle of mathematical induction (and the assumption $\|F_0\| < 1$), one can verify the following two inequalities for each $k \geq 0$:

$$\|F_{k+1}\| \leq \|F_k\| < 1.$$

Sequence $\|F_k\|$ is non-negative and decreasing, hence it is convergent. Its limit value t clearly satisfies $0 \leq t < 1$. By applying a limit on the both sides of (2.14), we obtain $\beta t(t-1) \geq 0$, which implies $t = 0$. Hence, $\|F_k\| \to 0$ when $k \to +\infty$.

On the other hand, based on the fact $XAX_k = X_k$ from (2.5), the error matrix E_k satisfies

$$E_k = X - X_k = XAX - XAX_k = XF_k, \quad k \geq 0$$

and hence

$$\|E_k\| \le \|X\| \, \|F_k\| \to 0, \quad k \to +\infty \tag{2.15}$$

which proves the convergence $X_k \to X$ when $k \to +\infty$.

Putting $E_k = XF_k$ in (2.13) and using both $XAX_k = X_k$ and $X_kAX = X_k$ (Lemma 2.1), we get

$$E_{k+1} = (1-\beta)E_k - \beta E_k A E_k$$

implying that $error_1 = (1-\beta)E_k$ and $error_2 = -\beta E_k A E_k$. It also proves the result related to the convergence order of the method (2.3). □

It is well-known that if $M \in \mathbb{C}^{n\times n}$ and $\epsilon > 0$ are given, there exists at least one matrix norm $\|\cdot\|$ such that $\|M\| \le \rho(M) + \epsilon$. Here $\rho(M)$ is the spectral radius of the matrix M. It is also known that $\rho(M) \le \|M\|$ for arbitrary matrix norm $\|\cdot\|$ and matrix $M \in \mathbb{C}^{n\times n}$. These two facts prove the equivalency of conditions $\rho(AX - AX_0) < 1$ and $\|AX - AX_0\| < 1$ for some matrix norm $\|\cdot\|$.

Corollary 2.1. *Let $A \in \mathbb{C}_r^{m\times n}$ be given matrix and $G \in \mathbb{C}_s^{n\times m}$, $0 < s \le r$ is selected but fixed matrix satisfying* $\operatorname{rank}(AG) = \operatorname{rank}(GA) = \operatorname{rank}(G)$. *Then the method* (2.3) *is convergent if* $0 < \beta \le 1$ *and*

$$\max_{1\le i\le s} |1 - \alpha\lambda_i(AG)| < 1, \tag{2.16}$$

where $s = \operatorname{rank}(AG)$.

Proof. According to Theorem 2.1, the method (2.3) is convergent if $\rho(F_0) < 1$. Assume that $\lambda_i = \lambda_i(AG)$, $i = 1, 2, \dots, s = \operatorname{rank}(AG)$ are non-zero eigenvalues of AG corresponding to eigenvectors v_i. Since $AGv_i = \lambda_i v_i$, it holds that $v_i \in A\mathcal{R}(G)$ and $AXv_i = v_i$. Hence $(AX - AX_0)v_i = (1-\alpha\lambda_i)v_i$. Vice versa is also true, i.e. if μ_i is an eigenvalue of $AX - AX_0$ then $(1-\mu_i)/\alpha$ is an eigenvalue of AX_0 ($\alpha \ne 0$). If $v_{s+1}, v_{s+2}, \dots, v_m$ are eigenvectors of AG corresponding to the eigenvalue 0, then $v_i \in \mathcal{N}(G)$ and also $(AX - AX_0)v_i = 0$ for every $i = s+1, s+2, \dots, m$. Hence, the set of eigenvalues of $AX - AX_0$ is equal to

$$\{1-\alpha\lambda_1, 1-\alpha\lambda_2, \dots, 1-\alpha\lambda_s, 0\}$$

and $\rho(F_0) = \max_{1\le i\le s} |1-\alpha\lambda_i(AG)|$. The result now follows from Theorem 2.1. □

The following corollary immediately follows from (2.2) and Theorem 2.1.

Corollary 2.2. *If the condition* (2.16) *is satisfied then the iterates X_k defined by* (2.3) *converge to*

$$\lim_{k\to\infty} X_k = \begin{cases} A^{\dagger}, & X_0 = \alpha A^*; \\ A^{\dagger}_{M,N}, & X_0 = \alpha A^{\sharp}; \\ A^{\#}, & X_0 = \alpha A; \\ A^{D}, & X_0 = \alpha A^{l}, \ l \ge \operatorname{ind}(A); \\ A^{(2)}_{\mathcal{R}(X_0),\mathcal{N}(X_0)}, & \textit{in general} \end{cases} \tag{2.17}$$

where $A^{\sharp} = N^{-1}A^*M$ denotes weighted conjugate-transpose matrix of A.

Remark 2.3. We can also prove identities $E_k = R_kX$, where $R_k = XA - X_kA$, and $R_{k+1} = (1-\beta)R_k - \beta R_k^2$ for every $k \ge 0$. The proof is similar to the proof of Theorem 2.1. In that sense, if $\|R_0\| < 1$ then also $\|R_k\| \to 0$ and hence $\|E_k\| \to 0$, i.e. $X_k \to X$ when $k \to +\infty$. The condition $\|R_0\| < 1$ is satisfied for some matrix norm in the case $\rho(R_0) < 1$, which is equivalent to

$$\max_{1\le i\le s} |1 - \alpha\lambda_i(GA)| < 1, \tag{2.18}$$

where $s = \operatorname{rank}(GA)$. Hence, we need to verify that either (2.16) or (2.18) is satisfied, to prove the convergence of the method (2.3). If $G = A^*$ then $s = r$ and $\lambda_i(AA^*) = \lambda_i(A^*A)$ for all $i = 1, 2, \dots, r$ ($r = \operatorname{rank}(A^*A) = \operatorname{rank}(AA^*)$). Hence, conditions (2.16) and (2.18) are equivalent in this special case.

Condition $\max_{1\le i\le r} |1-\alpha\lambda_i(A^*A)| < 1$ is equivalent to $0 < \alpha < 2/\lambda_{\max}(A^*A)$, which can be proven by simple algebraic transformations. It is clear that $\alpha = 2/\operatorname{Tr}(A^*A)$ (as suggested in [9]) satisfies the previous condition and hence the method (2.1) is convergent for this choice.

To simplify notation, denote $\lambda_i = \lambda_i(AG)$. In some cases, it is easier to check the condition equivalent to (2.16), shown in the following lemma. The proof of the equivalency is given in the paper [10]. The same theorem also holds for $\lambda_i = \lambda_i(GA)$.

Lemma 2.3 *([10, Theorem 2.4]). Condition* $\max_{1\le i\le s} |1-\alpha\lambda_i| < 1$ *is satisfied if and only if* $\operatorname{Re}\lambda_i$ *has the same sign for all $i = 1, 2, \dots, s$ and one of the following conditions holds:*

1. $\operatorname{Re}\lambda_i > 0$ *and* $0 < \alpha < \frac{2\operatorname{Re}\lambda_i}{|\lambda_i|^2}$
2. $\operatorname{Re}\lambda_i < 0$ *and* $\frac{2\operatorname{Re}\lambda_i}{|\lambda_i|^2} < \alpha < 0$.

Hence, the assumption that all real parts $\operatorname{Re}\lambda_i, i = 1, 2, \ldots, s$ have the same sign is the sufficient condition for existence of the value of α which ensures convergence of the method (1.1). Moreover, in the case of convergence, the following theorem implies that the order of convergence of the method (2.3) is exactly 1, in the case $\beta \neq 1$.

Theorem 2.2. *Suppose that method (2.3) is convergent. Then*

$$\lim_{k\to+\infty}\frac{t_{k+1}}{t_k} = \lim_{k\to+\infty}\frac{s_{k+1}}{s_k} = \lim_{k\to+\infty}\frac{d_{k+1}}{d_k} = 1-\beta \tag{2.19}$$

where $t_k = \|E_kA\|$, $s_k = \|E_k\|$ and $d_k = \|E_{k+1} - E_k\|$.

Proof. The proof is identical to one shown in [1, Theorem 3.2]. □

In the particular case $G = A^*$, the result of Theorem 2.1 reduces to the statement [1, Theorem 3.1] and the result of Theorem 2.2 reduces to [1, Theorem 3.4].

The following lemma shows that $\mathcal{R}(X_k)$ and $\mathcal{N}(X_k)$ do not depend on k and are equal to $\mathcal{R}(G)$ and $\mathcal{N}(G)$ respectively.

Lemma 2.4. *Suppose that the method (2.3) is convergent. Then the sequence X_k satisfies $\mathcal{R}(X_k) = \mathcal{R}(G)$ and $\mathcal{N}(X_k) = \mathcal{N}(G)$ for each $k \geq 0$.*

Proof. Recall that $G = UV$ is a full-rank factorization of G. Statements $\mathcal{R}(X_k) \subseteq \mathcal{R}(X_0) = \mathcal{R}(G)$ and $\mathcal{N}(X_k) \supseteq \mathcal{N}(X_0) = \mathcal{N}(G)$ are proved in Lemma 2.1. To prove the opposite inclusions we consider $\mathcal{N} = \bigcup_{k\in\mathbb{N}_0}\mathcal{N}(X_k)$. Let $y \in \mathcal{N}$ be arbitrary vector and let $y \in \mathcal{N}(X_{k_0})$ for some $k_0 \in \mathbb{N}_0$. Since $y \in \mathcal{N}(X_k)$ for all $k \geq k_0$ we have $X_ky = 0$ and using Theorem 2.1 we have

$$A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V)}y = \lim_{k\to+\infty} X_ky = 0.$$

Last implies $y \in \mathcal{N}(A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V)}) = \mathcal{N}(G)$ and $\mathcal{N} \subseteq \mathcal{N}(G)$. Furthermore holds

$$\mathcal{N}(G) \subseteq \mathcal{N}(X_k) \subseteq \mathcal{N} \subseteq \mathcal{N}(G),$$

and hence we conclude $\mathcal{N}(X_k) = \mathcal{N}(G)$. Now the relation

$$\dim\mathcal{R}(X_k) = m - \dim\mathcal{N}(X_k) = m - \dim\mathcal{N}(G) = \dim\mathcal{R}(G)$$

and $\mathcal{R}(X_k) \subseteq \mathcal{R}(G)$ directly imply $\mathcal{R}(X_k) = \mathcal{R}(G)$. Therefore

$$X = A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V)} = U(VAU)^{-1}V,$$

which completes the proof. □

2.3. *Influence of the roundoff errors*

As it was the case when $G = A^*$ and $X = A^\dagger$ (i.e. with the sequences (1.1) and (2.1)), the sequence X_k defined by (2.3) is also influent on the roundoff errors. In other words, Theorem 4.1 in [1] is valid in this general case. However, we have to make a stronger assumption.

Theorem 2.3. *Consider the iterative method (2.3) and assume that the roundoff error in pth iteration initiates $\mathcal{N}(\tilde{X}_pA\tilde{X}_p) \supset \mathcal{N}(\tilde{X}_p)$. Then the resulting method*

$$\tilde{X}_{k+1} = (1+\beta)\tilde{X}_k - \beta\tilde{X}_kA\tilde{X}_k, \quad k \geq p, \tag{2.20}$$

diverges and $\|\tilde{X}_k\| \geq c \cdot (1+\beta)^{k-p}$, where $c > 0$ is a constant which depends only on $\tilde{X}_p$. The matrix norm $\|\cdot\|$ is induced by some corresponding vector norm.

Proof. Let us consider $y \in \mathcal{N}(\tilde{X}_pA\tilde{X}_p) \setminus \mathcal{N}(\tilde{X}_p)$. We will prove the identities $\tilde{X}_ky = (1+\beta)^{k-p}\tilde{X}_py$ and $X_kAX_ky = 0$ by mathematical induction. Since it holds for $k = p$ by assumption, we can assume that it holds for some $k \geq p$ and prove that it is valid for $k+1$. Therefore

$$\begin{aligned}\tilde{X}_{k+1}A\tilde{X}_{k+1}y &= \tilde{X}_{k+1}A((1+\beta)\tilde{X}_ky - \beta\tilde{X}_kA\tilde{X}_ky)\\ &= (1+\beta)\tilde{X}_{k+1}A\tilde{X}_ky\\ &= (1+\beta)((1+\beta)\tilde{X}_k - \beta\tilde{X}_kA\tilde{X}_k)A\tilde{X}_ky\\ &= (1+\beta)^2\tilde{X}_kA\tilde{X}_ky - (1+\beta)\beta\tilde{X}_kA\tilde{X}_kA\tilde{X}_ky = 0\end{aligned}$$

and

$$\tilde{X}_{k+1}y = (1+\beta)\tilde{X}_k y - \beta\tilde{X}_k A\tilde{X}_k y = (1+\beta)^{k-p+1}X_p y.$$

This completes the proof by mathematical induction.

It now implies

$$\|\tilde{X}_k\| \geq \frac{\|\tilde{X}_k y\|}{\|y\|} = (1+\beta)^{k-p}\frac{\|\tilde{X}_p y\|}{\|y\|} = c\cdot(1+\beta)^{k-p}$$

for $c := \|\tilde{X}_p y\|/\|y\|$. Previous inequality also shows that $\|\tilde{X}_k\| \to +\infty$ when $k \to +\infty$ and hence $\tilde{X}_k$ diverges. □

Note that $\operatorname{rank}(\tilde{X}_p) > \operatorname{rank}(A)$ implies $\mathcal{N}(\tilde{X}_p A\tilde{X}_p) \supset \mathcal{N}(\tilde{X}_p)$ but $\operatorname{rank}(\tilde{X}_p) > \operatorname{rank}(X_p) = \operatorname{rank}(G)$ (Lemma 2.4) will not necessarily cause $\mathcal{N}(\tilde{X}_p A\tilde{X}_p) \supset \mathcal{N}(\tilde{X}_p)$, since $\operatorname{rank}(G) \leq \operatorname{rank}(A)$. Hence, simple increase of the rank of X_p due to the roundoff errors will not always cause divergence of the iterative method (2.1). On the other hand, it will certainly change its limit value and produce the error in the computation of $X = A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)}$.

In the particular case $G = A^*$ we can prove the same statement as [1, Theorem 4.1].

Theorem 2.4 (*[1, Theorem 4.1]*). *Consider the iterative method (2.1) and assume that the roundoff error in p-th iteration initiates* $\operatorname{rank}(\tilde{X}_p) > \operatorname{rank}(X_p) = \operatorname{rank}(A)$. *Then, resulting method*

$$\tilde{X}_{k+1} = (1+\beta)\tilde{X}_k - \beta\tilde{X}_k A\tilde{X}_k, \quad k \geq p, \tag{2.21}$$

diverges and $\|\tilde{X}_k\| \geq c\cdot(1+\alpha)^{k-p}$, *where* $c > 0$ *is the constant which depends only on* $\tilde{X}_p$.

3. Numerical examples

In this section we test the method (2.3) (including its special case (2.1)) on a series of numerical examples. These examples include a set of randomly generated matrices and matrices from Matrix Computation Toolbox [3] and Matrix Market [4].

3.1. Numerical examples of small dimensions for outer inverses

Example 3.1. Let us consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 4 & 6 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 4 \\ 4 & 5 & 6 & 7 & 6 \\ 6 & 6 & 7 & 7 & 8 \end{bmatrix} \tag{3.1}$$

of rank $r = \operatorname{rank}(A) = 4$. We choose the matrix $G = UV$ of rank $s = 2 < r$, where U and V are full rank matrices

$$U = \begin{bmatrix} 0 & 0 \\ 2 & 1 \\ 3 & 2 \\ 5 & 3 \\ 1 & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{3.2}$$

The choices $\alpha = 0.002$, $\beta = 0.99$ in the iterative process (2.3) produces the following outer inverse of A with the precision 10^{-15} after 21 iterative steps:

$$A^{(2)}_{\mathcal{R}(G),\mathcal{N}(G)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.12069 & 0.109195 & -0.12069 & 0.109195 & -0.12069 & 0.109195 \\ 0.344828 & -0.264368 & 0.344828 & -0.264368 & 0.344828 & -0.264368 \\ 0.224138 & -0.155172 & 0.224138 & -0.155172 & 0.224138 & -0.155172 \\ -0.586207 & 0.482759 & -0.586207 & 0.482759 & -0.586207 & 0.482759 \end{bmatrix}.$$

The choice $\alpha = 0.07$ produces the same result (in 6 decimal digits) after 18 iterations.

Example 3.2. In this example we compute the Drazin inverse of the matrix

$$A=\begin{bmatrix}
2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\
-1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & -2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 2 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & -2\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 2
\end{bmatrix}$$

with $\mathrm{ind}(A)=3$. Using $R=A^3$, $\alpha=0.05$ and $\beta=0.9$ we get the following approximation of the Drazin inverse with the precision 10^{-9} after 22 iterations:

$$A^d=\begin{bmatrix}
0.25 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1.25 & 1.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-1.6641 & -0.9922 & 0.25 & -0.25 & 0 & 0 & 0 & 0 & -0.0625 & -0.0625 & 0 & 0.1563\\
-1.1953 & -0.6797 & -0.25 & 0.25 & 0 & 0 & 0 & 0 & -0.0625 & 0.1875 & 0.6875 & 1.3438\\
-2.7637 & -1.0449 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.4844 & 2.5781 & 3.3203 & 6.6406\\
-2.7637 & -1.0449 & -1.875 & -1.25 & -1.25 & 1.25 & 1.25 & 1.25 & 1.484 & 2.5781 & 4.5703 & 8.5156\\
14.109 & 6.3008 & 6.625 & 3.375 & 5 & -3 & -5 & -5 & -4.1875 & -8.5 & -10.5078 & -22.4609\\
-19.324 & -8.5078 & -9.75 & -5.25 & -7.5 & 4.5 & 7.5 & 7.5 & 6.375 & 12.5625 & 15.9766 & 33.7891\\
-0.625 & -0.3125 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & -0.25 & -0.875 & -1.625\\
-1.25 & -0.9375 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.25 & -0.875 & -1.625\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.25 & 1.25\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.25
\end{bmatrix}.$$

3.2. Numerical examples for the Moore–Penrose inverse

Theorem 2.4 implies that the last computed iteration is not necessarily the best one. Hence, we have to define a criterion to be used for selecting the best iteration. In contract to [1], we used the maximal residual norm criterion, i.e. we select the iteration which minimizes the following value:

$$\mathrm{res}(X)=\max\{\|AXA-A\|,\ \|XAX-X\|,\ \|AX-(AX)^*\|,\ \|XA-(XA)^*\|\}.$$

This criterion produces slightly more reliable approximations of $A^\dagger$ than the one used in [1] (minimum of $d_k=\|X_{k+1}-X_k\|$).

We used $\beta=0.9$ and $\alpha=2/\mathrm{Tr}(A^*A)$, as suggested in [9]. It can be proven straightforward that condition (2.16) (for $G=A^*$) is satisfied for this choice of α.

Numerical results on randomly generated test matrices are shown in the Table 1. Each matrix is generated by the formula

$$\mathrm{randrankn}(n,r)=\mathrm{randn}(n,r)\cdot\mathrm{randn}(r,n),$$

where $\mathrm{randn}(m,n)$ is matrix containing pseudorandom values drawn from the standard normal distribution. Each matrix $\mathrm{randrankn}(n,r)$ has (almost surely) rank equal to r, since $\mathrm{randn}(m,n)$ are (almost surely) full-rank matrices. It can be seen from the norms of the residual matrices that method (2.1) computes the Moore–Penrose inverse with acceptable accuracy.

Numerical results on test matrices from Matrix Computation Toolbox in `Matlab` [3] are given in Table 2. We can also see that method (2.1) computes Moore–Penrose inverse with satisfactory accuracy for the most of the tested matrices.

In a few cases, we were unable to generate 200×200 test matrix, due to the extremely large elements contained in such a matrix. For those matrices, we used 10×10 representative. Results are given in Table 3.

Finally, we compared our results with the method `qrginv` from [2], obtained by the QR factorization, as well as with SVD (Singular Value Decomposition) method, implemented in `Matlab` built-in function `pinv`. Testings are done on the same matrices as in [2].

As it can be seen from Table 4, the maximal residual norm of the equations (1)–(4) of both `qrginv` and our iterative method (2.1) are comparable.

Finally, we provide testing results on sparse matrices taken from Matrix Market [4]. These matrices come from real world problems and mostly correspond to the discretization of a certain partial differential equations. Iterative methods are generally suitable for application on large sparse matrices. Residual norms and numbers of iterative steps (denoted by It.) are given in Table 5.

It can be seen from Table 5 that the Moore–Penrose inverse is computed with satisfactory accuracy for all test matrices. It can be also seen that obtained results are comparable with those obtained by a method `qrginv` from [2].

Table 1
Testing results on random generated test matrices $A = \text{randrankn}(n, r) \in \mathbb{C}^{n\times n}$ having $r = \text{rank}A$ equal to $5n/6$, $n/2$ and $n/10$ respectively.

n	rank(A)	It.	cond(A)	$\|AXA - A\|$	$\|XAX - X\|$	$\|AX - (AX)^*\|$	$\|XA - (XA)^*\|$
50	42	25	7.96e+17	3.54e−12	6.92e−12	2.26e−14	9.27e−15
100	83	26	9.34e+16	3.13e−12	8.46e−12	2.45e−14	1.20e−14
150	125	26	6.06e+17	5.26e−12	6.36e−12	3.57e−14	1.36e−14
200	167	27	8.03e+17	2.39e−12	1.07e−11	4.84e−14	1.85e−14
250	208	26	3.39e+17	7.71e−12	5.21e−12	4.24e−14	1.82e−14
300	250	27	8.04e+17	2.58e−12	9.01e−12	5.34e−14	1.60e−14
350	292	27	2.94e+20	8.75e−12	5.63e−12	6.37e−14	1.76e−14
400	333	27	5.22e+17	5.22e−12	5.35e−12	7.25e−14	1.90e−14
450	375	27	2.45e+18	6.10e−12	4.75e−12	7.10e−14	1.95e−14
500	417	27	2.04e+18	5.69e−12	4.79e−12	6.69e−14	2.05e−14
50	25	21	8.86e+17	9.03e−13	8.59e−13	2.57e−15	2.29e−15
100	50	22	3.59e+17	2.41e−13	1.14e−12	3.12e−15	3.39e−15
150	75	22	8.90e+17	3.74e−13	8.70e−13	3.85e−15	3.73e−15
200	100	23	8.51e+17	3.61e−13	1.10e−12	4.80e−15	4.95e−15
250	125	22	7.48e+17	7.33e−13	5.95e−13	5.02e−15	5.21e−15
300	150	23	8.83e+17	4.48e−13	9.41e−13	3.80e−15	4.06e−15
350	175	23	2.63e+18	5.51e−13	8.74e−13	3.89e−15	4.36e−15
400	200	23	1.17e+18	6.74e−13	8.07e−13	4.24e−15	4.63e−15
450	225	23	9.88e+17	7.84e−13	7.43e−13	4.58e−15	4.93e−15
500	250	23	2.74e+18	9.69e−13	7.06e−13	4.84e−15	5.19e−15
50	5	18	6.85e+18	3.76e−14	9.66e−14	5.06e−16	6.33e−16
100	10	19	1.90e+18	3.32e−14	1.26e−13	7.53e−16	7.59e−16
150	15	19	2.07e+18	5.53e−14	1.05e−13	7.68e−16	9.51e−16
200	20	19	4.46e+18	7.63e−14	9.29e−14	8.78e−16	1.02e−15
250	25	19	2.64e+18	1.01e−13	7.04e−14	9.79e−16	1.16e−15
300	30	20	1.58e+19	9.09e−14	8.64e−14	6.07e−16	7.45e−16
350	35	20	1.56e+20	1.01e−13	8.35e−14	6.38e−16	8.12e−16
400	40	20	6.39e+18	1.31e−13	7.11e−14	6.91e−16	8.36e−16
450	45	20	3.06e+19	1.46e−13	6.58e−14	7.59e−16	9.22e−16
500	50	21	5.62e+19	1.71e−13	1.22e−13	7.57e−16	9.84e−16

3.3. *Numerical examples for Drazin inverse*

In this section we show some testing results of the method (2.3) in the case $G = A^l$, $l \geq \text{ind}(A)$. In such case, the method converges to Drazin inverse $X = A^D$ of matrix A. Note that, according to Lemma 2.3, all non-zero eigenvalues of matrix A^{l+1} have to have the same sign of its real part. It is obviously not satisfied for an arbitrary matrix A.

We use randomly generated test matrices with specified rank and with property that each eigenvalue is real and has single multiplicity. Such matrices can be randomly generated by the rule $A = PDP^*$, where P is random orthogonal matrix and $D = \text{diag}(\nu_1, \nu_2, \ldots, \nu_r, 0, \ldots, 0)$. Eigenvalues $\nu_1, \nu_2, \ldots, \nu_r$ are arbitrary (randomly generated) positive real numbers. Non-zero eigenvalues of $GA = A^{l+1}$ are equal to $\lambda_i = \nu_i^{l+1}$ for $i = 1, 2, \ldots, r$, which means that they are real and positive. Hence, the condition of Lemma 2.3 now reduces to $0 < \alpha < 2/\lambda_i$, $i = 1, 2, \ldots, r$. This is obviously satisfied for any $0 < \alpha < 2/Tr(A^{l+1})$, which is used in our testing.

Testing results are shown in Table 6. Here we also return the iteration minimizing the maximal residual norm

$$\text{res}_d(X) = \max\{\|A^l XA - A^l\|, \|XAX - X\|, \|AX - XA\|\}.$$

As it can be seen from the table, iterative method (2.3) works well for all randomly generated matrices.

4. Conclusion

We observed a drawback in the iterative method for computing the Moore–Penrose inverse, based on Penrose equations (2) and (4), which is introduced in [1]. Namely, this method uses the same coefficient β both to define the starting value and the computation of the next iteration. In this way, the parameter β is limited by the convergence condition (1.2), which frequently implies very slow convergence. Much better performance of the method have been obtained by overcoming this limitation. We also define a generalization of the method. More precisely, we show that an appropriate choice of the initial iteration is the fact which defines convergence to desired class of outer inverses. Numerical examples are presented.

Our original intention was to overcome a deficiency in the choice of the initial approximation in the iterative method from [1]. But, we observed additional property of the method: appropriate choice of the initial approximation ensures convergence to a particular outer inverse or a particular class of outer inverses, as it is shown in Corollary 2.2.

Table 2
Testing results on 200 × 200 matrices from Matrix Computation Toolbox in `Matlab` [3]. Parameter values are $\beta = 0.9$ and $\alpha = 2/\text{Tr}(A^*A)$.

	It.	Name	cond(A)	$\|AXA - A\|$	$\|XAX - X\|$	$\|AX - (AX)^*\|$	$\|XA - (XA)^*\|$
1	4	cauchy	1.187e+20	1.665e−01	4.798e−01	3.878e−16	5.186e−16
2	42	chebspec	3.697e+15	4.734e−10	3.099e−09	1.449e−11	2.304e−14
3	16	chebvand	1.494e+18	3.479e−01	3.830e−01	2.762e−15	2.407e−15
4	50	chow	–	1.222e−13	1.036e−15	3.613e−14	1.149e−15
5	30	circul	2.010e+02	1.014e−10	4.706e−15	4.881e−13	1.826e−14
6	31	clement	1.505e+21	1.533e−13	1.125e−17	1.240e−15	2.995e−15
7	36	condex	1.010e+02	9.534e−13	2.071e−13	1.949e−13	1.192e−14
8	99	cycol	1.516e+33	1.959e−14	3.323e−17	3.382e−15	6.318e−16
9	28	dramadah	1.065e+17	3.043e−13	6.154e−13	1.622e−14	7.617e−15
10	88	fiedler	2.779e+04	8.784e−09	3.229e−12	9.124e−09	2.543e−12
11	15	forsythe	6.711e+07	8.743e−07	1.197e−06	0.000e+00	0.000e+00
12	49	frank	1.558e+18	1.470e−11	2.790e−12	4.192e−13	1.366e−14
13	32	gearmat	3.074e+16	2.078e−13	3.606e−10	1.415e−14	5.281e−15
14	80	grcar	3.618e+00	1.268e−15	3.137e−16	6.345e−16	6.151e−16
15	97	invhess	7.529e+03	2.132e−11	3.338e−16	1.702e−13	3.572e−14
18	57	jordbloc	2.553e+02	2.220e−16	1.417e−14	8.865e−33	8.865e−33
19	67	kahan	7.533e+24	3.394e−14	2.088e−09	8.968e−10	8.989e−14
20	30	kms	8.995e+00	2.281e−15	7.545e−16	7.455e−16	5.553e−16
22	46	lehmer	4.194e+04	8.709e−11	4.889e−10	5.466e−09	2.715e−12
23	126	lesp	1.362e+02	2.672e−13	8.575e−17	4.023e−16	3.901e−16
24	8	lotkin	2.508e+21	3.265e−01	2.686e−01	7.993e−16	3.375e−15
25	100	minij	6.517e+04	1.477e−08	7.166e−12	2.356e−08	9.545e−12
26	42	moler	2.471e+16	1.460e−09	7.566e−13	4.380e−10	1.270e−12
27	25	orthog	1.000e+00	1.070e−15	1.116e−15	1.241e−15	1.239e−15
28	150	parter	3.591e+00	4.437e−15	6.242e−16	1.410e−15	1.315e−15
29	55	pei	2.010e+02	6.230e−13	1.026e−14	2.725e−13	4.684e−13
30	7	prolate	3.982e+17	4.367e−01	3.392e−01	4.253e−16	3.120e−16
31	53	randcolu	5.802e+01	7.009e−15	6.672e−14	5.265e−14	7.736e−15
32	91	randcorr	9.926e+01	1.165e−14	1.658e−13	1.511e−13	1.219e−14
33	67	rando	1.998e+03	5.645e−13	9.901e−14	5.526e−13	6.081e−14
34	3	randsvd	6.711e+07	3.804e−01	3.928e−01	7.163e−17	7.336e−17
35	96	redheff	4.589e+02	1.429e−14	1.557e−14	1.498e−14	1.271e−14
36	89	riemann	5.728e+02	1.226e−12	1.677e−14	1.310e−12	1.129e−14
37	72	ris	3.591e+00	1.729e−15	1.262e−15	1.315e−15	1.434e−15
38	53	smoke	1.273e+02	1.596e−14	3.635e−14	3.317e−14	1.259e−14
39	55	toeppd	5.243e+02	5.038e−12	1.972e−14	1.686e−12	4.072e−14
40	38	triw	1.379e+19	1.055e−13	2.596e−14	1.812e−14	2.926e−14
41	3	hilb	7.133e+19	3.149e−01	2.994e−01	2.678e−16	3.482e−16
43	42	magic	2.125e+19	5.611e−09	2.802e−12	4.669e−14	2.608e−14
45	45	rand	5.257e+03	1.403e−12	7.268e−13	2.306e−12	1.435e−13
46	46	randn	4.991e+03	4.829e−12	4.691e−11	9.356e−11	3.530e−13
47	103	augment	1.091e+04	7.105e−12	3.701e−12	2.200e−11	6.059e−13
48	31	gfpp	8.981e+01	7.738e−14	6.069e−16	1.657e−14	7.032e−15
49	42	magic	2.125e+19	5.611e−09	2.802e−12	4.669e−14	2.608e−14
50	64	makejcf	5.413e+06	4.214e−07	7.467e−10	5.313e−07	1.165e−10
51	63	rschur	1.179e+03	3.422e−13	1.838e−16	5.554e−16	5.554e−16
52	11	vand	1.860e+20	3.419e−01	4.232e−01	9.684e−16	1.586e−15

Table 3
Testing results on several 10 × 10 matrices from Matrix Computation Toolbox in `Matlab` [3]. Parameter values are $\beta = 0.9$ and $\alpha = 2/\text{Tr}(A^*A)$.

	It.	Name	cond(A)	$\|AXA - A\|$	$\|XAX - X\|$	$\|AX - (AX)^*\|$	$\|XA - (XA)^*\|$
16	97	invol	3.772e+06	2.211e−08	5.008e−09	1.848e−08	7.587e−11
17	96	ipjfact	1.398e+08	5.433e−08	1.846e−12	6.758e−08	5.420e−11
21	176	krylov	3.367e+01	6.029e−15	5.047e−16	1.665e−15	3.165e−15
42	101	invhilb	4.766e+05	2.590e−07	4.025e−12	8.923e−08	1.291e−11
44	108	pascal	8.518e+03	6.110e−12	2.603e−12	3.918e−12	4.948e−13

Acknowledgment

The authors gratefully acknowledge the support from the research project 174013 of the Serbian Ministry of Science.

Table 4
Comparison between our method, method proposed in [2] (`qrginv`) and built-in `Matlab` function for generalized inversion (`pinv`). Testings are done on 200×200 matrices from Matrix Computation Toolbox [3].

Name	Method	$\|AXA - A\|$	$\|XAX - X\|$	$\|AX - (AX)^*\|$	$\|XA - (XA)^*\|$
chow	pinv	4.741e−13	1.896e−13	2.312e−13	2.261e−13
	qrginv	5.012e−13	1.078e−13	5.269e−13	1.691e−13
	(2.1)	1.222e−13	1.036e−15	3.613e−14	1.149e−15
cycol	pinv	2.965e−13	5.690e−16	1.386e−14	1.251e−14
	qrginv	4.405e−14	7.144e−17	1.507e−15	1.224e−15
	(2.1)	1.959e−14	3.323e−17	3.382e−15	6.318e−16
gearmat	pinv	1.899e−14	3.033e−13	8.062e−14	7.561e−14
	qrginv	2.870e−15	2.627e−13	7.416e−14	1.923e−14
	(2.1)	2.078e−13	3.606e−10	1.415e−14	5.281e−15
kahan	pinv	1.432e−13	4.155e−09	3.795e−09	9.070e−14
	qrginv	2.128e−05	1.822e−09	6.923e−01	6.964e−15
	(2.1)	3.394e−14	2.088e−09	8.968e−10	8.989e−14
lotkin	pinv	8.752e−05	1.034e+08	1.004e−03	1.149e−03
	qrginv	8.347e−06	4.897e−08	4.629e−02	3.546e−11
	(2.1)	3.265e−01	2.686e−01	7.993e−16	3.375e−15
prolate	pinv	1.748e−04	7.398e+09	9.568e−03	8.093e−03
	qrginv	1.383e−06	6.140e−07	4.771e−02	6.588e−11
	(2.1)	4.367e−01	3.392e−01	4.253e−16	3.120e−16
hilb	pinv	7.779e−05	3.184e+09	3.266e−03	5.145e−03
	qrginv	7.878e−06	8.815e−09	1.005e−01	6.941e−12
	(2.1)	3.149e−01	2.994e−01	2.678e−16	3.482e−16
magic	pinv	1.211e−08	2.491e−19	9.701e−14	4.236e−14
	qrginv	1.003e−08	3.947e−19	1.821e−13	4.566e−14
	(2.1)	5.611e−09	2.802e−12	4.669e−14	2.608e−14
vand	pinv	3.337e−04	1.708e+08	2.206e−03	1.852e−03
	qrginv	1.273e−05	2.618e−07	5.330e−01	5.553e−11
	(2.1)	3.419e−01	4.232e−01	9.684e−16	1.586e−15

Table 5
Testing results on large scale sparse matrices from Matrix Market [4].

Name	It.	$\|AXA - A\|$	$\|XAX - X\|$	$\|AX - (AX)^*\|$	$\|XA - (XA)^*\|$
watt1850	38	1.67e−13	4.50e−13	1.21e−12	2.35e−13
watt1033	40	9.65e−14	1.82e−13	1.39e−12	1.32e−13
gr_30_30	46	6.04e−13	2.02e−14	9.79e−14	1.59e−13
illc1850	45	1.01e−12	1.43e−11	1.08e−11	6.72e−12
illc1330	53	7.53e−12	1.44e−10	1.03e−10	1.21e−10
watt_1	13	2.01e−04	4.41e−04	4.63e−22	7.84e−26
watt_2	14	8.38e−05	5.09e−04	1.37e−15	6.64e−15

Table 6
Testing results for computation of Drazin inverse A^D of randomly generated matrices.

n	rank(A)	It.	cond(A)	$\|A^l XA - A^l\|$	$\|XAX - X\|$	$\|AX - XA\|$
50	25	19	9.64e+16	8.29e−13	1.21e−12	1.20e−15
100	50	20	7.95e+17	9.46e−13	1.62e−12	1.89e−15
150	75	21	1.59e+17	5.19e−13	2.43e−12	2.27e−15
200	100	21	1.16e+18	1.51e−12	1.95e−12	2.56e−15
250	125	21	9.88e+17	3.07e−12	3.06e−12	2.67e−15
300	150	22	8.49e+17	6.44e−13	2.96e−12	1.87e−15
350	175	22	1.10e+19	1.17e−12	2.65e−12	2.26e−15
400	200	22	6.24e+18	1.89e−12	2.54e−12	2.21e−15
450	225	22	4.94e+18	2.94e−12	2.93e−12	2.31e−15
500	250	22	5.85e+17	3.93e−12	3.90e−12	2.62e−15

References

[1] M.D. Petković, P.S. Stanimirović, Iterative method for computing Moore–Penrose inverse based on Penrose equations, J. Comput. Appl. Math. 235 (2011) 1604–1613.

[2] 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.

[3] N.J. Higham, The matrix computation toolbox, http://www.ma.man.ac.uk/~higham/mctoolbox.
[4] Matrix Market, National Institute of Standards and Technology, Gaithersburg, MD. Available online from http://arxiv.org///math.nist.gov/MatrixMarket.
[5] G. Wang, Y. Wei, S. Qiao, Generalized Inverses: Theory and Computations, Science Press, Beijing/New York, 2004.
[6] D.S. Djordjević, P.S. Stanimirović, Y. Wei, The representation and approximation of outer generalized inverses, Acta Math. Hungar. 104 (2004) 1–26.
[7] Y. Wei, H. Wu, The representation and approximation for the generalized inverse $A_{T,S}^{(2)}$, Appl. Math. Comput. 135 (2003) 263–276.
[8] X. Sheng, G. Chen, Full-rank representation of generalized inverse $A_{T,S}^{(2)}$ and its applications, Comput. Math. Appl. 54 (2007) 1422–1430.
[9] L. Chen, E.V. Krishnamurthy, I. Macleod, Generalized matrix inversion and rank compuation by successive matrix powering, Parallel Comput. 20 (1994) 297–311.
[10] P.S. Stanimirović, D.S. Cvetković-Ilić, Successive matrix squaring algorithm for computing outer inverses, Appl. Math. Comput. 203 (2008) 19–29.