Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 596049, 6 pages
http://dx.doi.org/10.1155/2014/596049

Research Article

Execute Elementary Row and Column Operations on the Partitioned Matrix to Compute M-P Inverse $A^{\dagger}$

Xingping Sheng

School of Mathematics and Computational Science, Fuyang Normal College, Fuyang, Anhui, China

Correspondence should be addressed to Xingping Sheng; spy_sxp@sina.com

Received 3 November 2013; Revised 23 January 2014; Accepted 23 January 2014; Published 10 March 2014

Academic Editor: Sergei V. Pereverzyev

We first study the complexity of the algorithm presented in Guo and Huang (2010). After that, a new explicit formula for computational of the Moore-Penrose inverse $A^{\dagger}$ of a singular or rectangular matrix A. This new approach is based on a modified Gauss-Jordan elimination process. The complexity of the new method is analyzed and presented and is found to be less computationally demanding than the one presented in Guo and Huang (2010). In the end, an illustrative example is demonstrated to explain the corresponding improvements of the algorithm.

1. Introduction

Throughout this paper we use the following notation. Let C^n and $C_r^{m\times n}$ be the n dimensional complex space and the set of $m\times n$ complex matrices with rank r. For a matrix $A\in C^{m\times n}$, $R(A)$ and $N(A)$ are the range and null space of A; $r(A)$ and A^* denote the rank and the the conjugate transpose of A, while $A^{\dagger}$ and $\|A\|$ denote the M-P inverse and Frobenius norm, respectively.

In 1920, Moore [1] defined a new inverse of a matrix by projection matrices. Moore's definition of the generalized inverse of an $m\times n$ matrix A is equivalent to the existence of an $n\times m$ matrix G such that

$$AG = P_{R(A)}, \qquad GA = P_{R(G)}, \tag{1}$$

where $P_{R(A)}$ is the orthogonal projector on $R(A)$. Unaware of Moore's work, In 1955 Penrose [2] showed that there exists a unique matrix X satisfying the four conditions

$$\begin{aligned} AXA &= A,\\ XAX &= X,\\ (AX)^* &= AX,\\ (XA)^* &= XA, \end{aligned} \tag{2}$$

where $*$ denotes conjugate transpose. These conditions are equivalent to Moore's conditions. The unique matrix X that satisfied these conditions was known as the Moore-Penrose inverse (abbreviated M-P) and is denoted by $A^{\dagger}$. For a subset $\{i,j,\dots,k\}$ of the set $\{1,2,3,4\}$, the set of $n\times m$ matrices satisfied the equations $(i),(j),\dots,(k)$, from among (2) is denoted by $A^{\{i,j,\dots,k\}}$. These concepts can be found in Ben-Israel and Greville's famous book [3] or Campell and Meyer's book [4]. In their famous books [3, 4], the next statement is valid for a rectangular matrix.

Lemma 1 (see [3, 4]). *Let $A\in C^{m\times n}$, $X\in C^{n\times m}$ be the M-P inverse of A if and only if X is a $\{2\}$-inverse of A with range $R(A^*)$ and null space $N(A^*)$.*

In the latest fifty years, there have been many famous specialists and scholars, who investigated the generalized inverse and published many articles and books. Its perturbation theories were introduced in [5–7], and the algebraical perturbations were in [8, 9]. Some minors, Cramer rulers, and sums of $A^{\dagger}$ can be seen in [10–13]. There are a large number of papers [9, 14–18] using various methods, iterative or not, for computing $A^{\dagger}$.

One handy method of computing the inverse of a nonsingular matrix A is the Gauss-Jordan elimination procedure by executing elementary row operations on the pair (A, I) to transform it into (I, A^{-1}). Moreover Gauss-Jordan elimination can be used to determine whether a matrix is nonsingular or not. However, one cannot directly use this method on a generalized inverse of a rectangular matrix or a square singular matrix A.

Recently, Author [19] proposed a Gauss-Jordan elimination algorithm to compute $A^\dagger$, which required $3n^3$ multiplications and divisions. More recently, Ji [20] improved author's algorithm [19] and pointed that only $2n^3$ multiplications and divisions are required. Following these lines, Stanimirović and Petković in [21] extended the method of [20] to $A^{(2)}_{T,S}$. But these three algorithms need also switching. Guo and Huang [22] executed row and column elementary transformations for computing M-P inverse $A^\dagger$ by applying the rank equalities of matrix A. They did not analyze the complexity of their algorithm. In this paper, we first study the total number arithmetic operation of GH-algorithm, then improve it, and present an alternative explicit formula for the M-P inverse of a matrix; the improvements save the total number arithmetic operation. We must point that GH-algorithm and our algorithm do not need to switch blocks of certain matrix in the process of computation.

The paper is organized as follows. The computational complexity of GH-Algorithm 3 for computing M-P inverse $A^\dagger$ is surveyed in the next section. In Section 3, we derive a novel explicit expression for $A^\dagger$, propose a new Gauss-Jordan elimination procedure for $A^\dagger$ based on the formula, and study the computational complexity of the new approach and Algorithm 7. In Section 4, an illustrative example is presented to explain the corresponding improvements of the algorithm.

2. The Computational Complexity of GH-Algorithm

In [22], Guo and Huang gave a method of elementary transformation for computing M-P inverse $A^\dagger$ by applying the rank equalities of matrix A.

Lemma 2. *Suppose that $A \in C^{m\times n}$, $X \in C^{k\times l}$, $1 \le k \le n$, $1 \le l \le m$. If*

$$r\begin{pmatrix} A^*AA^* & A^*\begin{pmatrix} I_l \\ 0 \end{pmatrix} \\ \begin{pmatrix} I_k & 0 \end{pmatrix} A^* & X \end{pmatrix} = r(A). \tag{3}$$

Then $X = \begin{pmatrix} I_k & 0 \end{pmatrix} A^\dagger \begin{pmatrix} I_l \\ 0 \end{pmatrix}$. In particualr when $k = n$, $l = m$ and $X = A^\dagger$.

In [22], the authors also considered an algorithm based on Lemma 2, which was stated as follows.

Algorithm 3. M-P inverse GH-Algorithm is as follows.

(1) Compute partitioned matrix $N_1 = \begin{pmatrix} A^*AA^* & A^* \\ A^* & 0 \end{pmatrix}$.

(2) Make the block matrix A^*AA^* of $N_1(1,1)$ become $I_{r(A)}$ by applying elementary transformations. Meanwhile, the block matrices A^* of $N_1(1,2)$ and $N_1(2,1)$ are accordingly transformed. A new partitioned matrix $N_2 = \begin{pmatrix} I_{r(A)} & 0 & B_1 \\ 0 & 0 & 0 \\ C_1 & 0 & 0 \end{pmatrix}$ is obtained.

(3) Make the block matrices of $N_2(1,2)$ and $N_2(2,1)$ be zero matrices by applying matrix $I_{r(A)}$, which yields

$$N_3 = \begin{pmatrix} I_{r(A)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C_1B_1 \end{pmatrix} \tag{4}$$

Then $A^\dagger = C_1B_1$

Nevertheless, Guo and Huang [22] did not analyze the complexity of the numerical algorithm. In the following theorem, we will study the total number of arithmetic operations.

Theorem 4. *Let $A \in C_r^{m\times n}$; the total number of multiplications and divisions required in Algorithm 3 to compute M-P inverse $A^\dagger$ is*

$$\begin{aligned} &N(m,n,r) \\ &\quad = 2mn^2 + \frac{4m-r-1}{2}nr + (m-r)nr + mnr. \end{aligned} \tag{5}$$

Moreover, $N(m,n,r)$ is bounded above by $6mn^2 - (3/2)n^3 - (1/2)n^2$.

Proof. It needs $2mn^2$ multiplications to compute A^*AA^*. r row pivoting steps and column pivoting steps are needed to transform the partitioned matrix N_1 into N_2 following the $r(A^*AA^*) = r$. First row pivoting step involves $2m$ nonzero columns in $\begin{pmatrix} A^*AA^* & A^* \end{pmatrix}$. Thus, it needs $2m-1$ divisions and $(2m-1)(n-1)$ multiplications with a total of $(2m-1)n$ multiplications and divisions. On the second row pivoting step, there is one less column in the first part of the pair. There are $2m-1$ nonzero columns to deal with. These pivoting steps require $(2m-2)n$ operations. Following the same idea, the ith $(1 \le i \le r)$ pivoting steps require $(2m-i)n$ operations. So it requires $(2m-1)n + (2m-2)n + \cdots + (2m-r)n = ((4m-r-1)/2)nr$.

For simplicity, assume that $B_1 = \begin{pmatrix} I_r & B_{12} \end{pmatrix}$. Following the same line, this requires $(m-r)nr$ multiplications and divisions on the r column pivoting steps.

Then resume elementary row and columns operations on the matrix N_2 to transform it into N_3, which requires mnr multiplications, which is the count of C_1B_1.

Therefore, the total number of operations needed for computation of $A^\dagger$ is

$$\begin{aligned} &N(m,n,r) \\ &\quad = 2mn^2 + \frac{4m-r-1}{2}nr + (m-r)nr + mnr. \end{aligned} \tag{6}$$

Define a function $f(r) = 2mn^2 + ((4m-r-1)/2)nr + (m-r)nr + mnr$ for fixed m and n. Since $0 \le r \le \min\{m,n\} = n$, we have

$$\begin{aligned} &f'(r) \\ &\quad = \frac{4m-r-1}{2}n - \frac{1}{2}nr + (m-r)n - nr + mn \end{aligned}$$

$$
\begin{aligned}
&= \left(4m - \frac{1}{2} - 3r\right) n \\
&\geq \left(m - \frac{1}{2}\right) n > 0
\end{aligned}
\tag{7}
$$

which means that $f(r)$ is monotonically increasing over $[0, n]$ when $n \leq m$. Therefore $f(r)$ takes its maximum value when $r = n$, which implies

$$
f(r) \leq f(n) = 6mn^2 - \frac{3}{2}n^3 - \frac{1}{2}n^2. \tag{8}
$$

□

3. Main Results

The Gauss-Jordan row and column elimination procedure for the M-P inverse $A^{\dagger}$ of a matrix by Guo and Huang is based on the partitioned matrix $N_1 = \left(\begin{smallmatrix} A^*AA^* & A^* \\ A^* & 0 \end{smallmatrix}\right)$. In this section, we will first propose a modified Gauss-Jordan elimination process to compute $A^{\dagger}$and then summarize an algorithm of this method. Finally, the complexity of the algorithm is analyzed.

Theorem 5. *Let $A \in C_r^{m\times n}$; there exist an elementary row operation matrix $E_1 = \underset{n}{\left(\begin{smallmatrix} E_{11} \\ E_{12} \end{smallmatrix}\right)} \begin{smallmatrix} r \\ n-r \end{smallmatrix}$ and an elementary column operation matrix $F_1 = \begin{pmatrix} \underset{r}{F_{11}} & \underset{m-r}{F_{12}} \end{pmatrix} m$, such that*

$$
\begin{gathered}
B = E_1 A^* = \begin{pmatrix} E_{11} \\ E_{12} \end{pmatrix} A^* = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}, \\
C = A^* F_1 = A^* \begin{pmatrix} F_{11} & F_{12} \end{pmatrix} = \begin{pmatrix} C_1 & 0 \end{pmatrix},
\end{gathered}
\tag{9}
$$

where $\left(\begin{smallmatrix} B_1 \\ 0 \end{smallmatrix}\right)$ and $(C_1 \;\; 0)$ are the row and column reduced echelon form of A^*, respectively.

Further, there exists an elementary row operation matrix $E_2 \in C^{r\times r}$ such that

$$
E_2 B_1 A C_1 = I_r. \tag{10}
$$

Then

$$
A^{\dagger} = C_1 E_2 B_1 = C_1 \left(B_1 A C_1\right)^{-1} B_1 = A^* F_{11} E_2 E_{11} A^*. \tag{11}
$$

Proof. For $r(A) = r(A^*) = r$, there exist two elementary row and column operation matrices $E_1 = \underset{n}{\left(\begin{smallmatrix} E_{11} \\ E_{12} \end{smallmatrix}\right)} \begin{smallmatrix} r \\ n-r \end{smallmatrix}$ and $F_1 = \begin{pmatrix} \underset{r}{F_{11}} & \underset{m-r}{F_{12}} \end{pmatrix} m$, such that

$$
\begin{gathered}
B = E_1 A^* = \begin{pmatrix} E_{11} A^* \\ E_{12} A^* \end{pmatrix} = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}, \\
C = A^* F_1 = \begin{pmatrix} A^* F_{11} & A^* F_{12} \end{pmatrix} = \begin{pmatrix} C_1 & 0 \end{pmatrix}.
\end{gathered}
\tag{12}
$$

It is easy to check that $B_1 \in C_r^{r\times n}$ and $C_1 \in C_r^{m\times r}$; then the matrix $B_1 A C_1$ is nonsingular, which implies that there exists another elementary row operation matrix $E_2 \in C^{r\times r}$ such that

$$
E_2 B_1 A C_1 = I_r. \tag{13}
$$

The above formula also shows that $E_2 = (B_1 A C_1)^{-1}$.

Denote that $X = C_1 E_2 B_1 = A^* F_{11} E_2 E_{11} A^*$; it is obvious that $R(X) \supset R(A^*)$ and $N(X) \subset N(A^*)$. If we can prove $r(X) = r(A^*) = r$, then $R(X) = R(A^*)$ and $N(X) = N(A^*)$.

In fact, C_1 is a full column rank matrix and E_2 is an invertible matrix, which implies that $r(X) = r(B_1) = r(A^*) = r$.

By deducing, we obtain that

$$
XAX = C_1 E_2 B_1 A C_1 E_2 B_1 = C_1 E_2 B_1 = X. \tag{14}
$$

This means that X is a 2-inverse of A with $R(X) = R(A^*)$ and $N(X) = N(A^*)$. From Lemma 1, we know that $X = A^{\dagger}$. □

Remark 6. The representation of $A^{\dagger}$ in Theorem 5 is consistent with the one in [3], although we use Gauss-Jordan elimination procedure.

According to the representation introduced in Theorem 5, we summarize the following algorithm for computing M-P inverse $A^{\dagger}$.

Algorithm 7. M-P inverse-Sheng algorithm is as follows.

(1) Input: $A \in C_r^{m\times n}$.

(2) Execute elementary row operations on first n rows of the partitioned matrix $M_1 = \left(\begin{smallmatrix} 0 & A^* \\ A^* & 0 \end{smallmatrix}\right)$ into $M_2 = \left(\begin{smallmatrix} 0 & B \\ A^* & 0 \end{smallmatrix}\right)$, where $B = \left(\begin{smallmatrix} B_1 \\ 0 \end{smallmatrix}\right)$ is a reduced row-echelon matrix.

(3) Perform elementary column operations on first m columns of the partitioned matrix M_2 into $M_3 = \left(\begin{smallmatrix} 0 & B \\ C & 0 \end{smallmatrix}\right)$, where matrix $C = (C_1 \;\; 0)$ has a reduced column-echelon form.

(4) Compute $D = BAC = \left(\begin{smallmatrix} B_1 A C_1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and form the block matrix

$$
M_4 = \begin{pmatrix} D & B \\ C & 0 \end{pmatrix} = \begin{pmatrix} B_1 A C_1 & 0 & B_1 \\ 0 & 0 & 0 \\ C_1 & 0 & 0 \end{pmatrix}. \tag{15}
$$

(5) Execute the elementary row operations on first r rows of the partitioned matrix M_4 into $M_5 = \left(\begin{smallmatrix} I_r & 0 & \widetilde{B_1} \\ 0 & 0 & 0 \\ C_1 & 0 & 0 \end{smallmatrix}\right)$.

(6) Make the block matrices of $M_5(1,3)$ and $M_5(3,1)$ be zero matrices by applying elementary row and column transformations, respectively, through matrix I_r, which yields

$$
M_6 = \begin{pmatrix} I_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C_1\widetilde{B_1} \end{pmatrix}. \tag{16}
$$

Then $A^{\dagger} = C_1 \widetilde{B_1}$.

According to Algorithm 7, the next theorem will analyze the computational complexity of it.

Theorem 8. *The total number of multiplications and divisions required for Algorithm 7 to compute M-P inverse $A^{\dagger}$ of a matrix $A \in C_r^{m\times n}$ is*

$$
T(m, n, r) = \left(4mn - \frac{m+n}{2}\right) r + \frac{m-n}{2} r^2 - r^3. \tag{17}
$$

Further, the upper bound of $T(m,n,r)$ is less than $(7/2)mn^2 - (1/2)n^3 - ((m+n)/2)n$ when $n \leq m$.

Proof. For a matrix A with rank r, r pivoting steps are needed to make the partitioned matrix M_1 into M_2. First pivoting step involves m nonzero columns in A^*. Thus, it needs $m-1$ divisions and $(m-1)(n-1)$ multiplications with a total number of $(m-1)n$ multiplications and divisions. On the second pivoting step, there is one less column in the first part of the pair. There are $m-1$ nonzero columns to deal with. This pivoting step requires $(m-2)n$ operations. Following the same idea, the ith $(1 \leq i \leq r)$ pivoting step requires $(m-i)n$ operations. So these r pivoting steps require

$$\begin{aligned} &(m-1)\,n + (m-2)\,n + \cdots + (m-r)\,n \\ &\quad = \frac{2m-r-1}{2} nr \end{aligned} \tag{18}$$

multiplications and divisions to reach the matrix M_2.

Similarly, it needs $(n-1)m + (n-2)m + \cdots + (n-r)m = ((2n-r-1)/2)mr$ multiplications and divisions to change the matrix M_2 into M_3.

For simplicity, assume that $B_1 = (I_r \;\; B_{11})$ and $C_1 = \left(\begin{smallmatrix} I_r \\ C_{11} \end{smallmatrix}\right)$, which follows from that B and C are row-echelon and column-echelon reduced matrix, respectively.

$(m-r)rn + (n-r)r^2 = (mn-r^2)r$ multiplications are required to form $M_4 = \left(\begin{smallmatrix} B_1AC_1 & 0 & B_1 \\ 0 & 0 & 0 \\ C_1 & 0 & 0 \end{smallmatrix}\right)$ under the above assumption. Since every row of the partitioned matrix $(B_1AC_1 \;\; 0 \;\; B_1)$ has $m+1$ nonzero elements, each pivoting step needs mr multiplications and divisions. Thus, it requires mr^2 multiplications and divisions to obtain the matrix M_5.

Then resume elementary row and columns operations on the matrix M_5 to transform it into M_6. The complexity of this process is mnr multiplications, which is the count to compute $C_1\widetilde{B_1}$.

Hence, the total number of complexity of Algorithm 7 is

$$\begin{aligned} T\,(m,n,r) &= \frac{2m-r-1}{2}nr + \frac{2n-r-1}{2}mr \\ &\quad + \left(mn - r^2\right)r + mr^2 + mnr \\ &= \left(4mn - \frac{m+n}{2}\right)r + \frac{m-n}{2}r^2 - r^3. \end{aligned} \tag{19}$$

With fixed m and n, define a function $g(r) = (4mn-(m+n)/2)r + ((m-n)/2)r^2 - r^3$ for $0 \leq r \leq \min\{m,n\} = n$. Then we have

$$\begin{aligned} g'\,(r) &= \left(4mn - \frac{m+n}{2}\right) + (m-n)\,r - 3r^2 \\ &= 3\left(mn - r^2\right) + (m-n)\,r + \left(mn - \frac{m+n}{2}\right) \\ &\geq (m-n) + \left(mn - \frac{m+n}{2}\right) > 0 \end{aligned} \tag{20}$$

which implies that $g(r)$ is also monotonically increasing over $[0,n]$ when $n \leq m$.

Therefore, when $r = n$, $g(r)$ obtains its maximum value, which yields

$$g\,(r) \leq g\,(n) = \frac{7}{2}mn^2 - \frac{1}{2}n^3 - \frac{m+n}{2}n. \tag{21}$$

Furthermore, we give two remarks: one is explaining the computation speed and the other is how to improve the accuracy of Algorithm 7. □

Remark 9. The algorithm in this paper does not need to switch block of certain matrix in the process computation, unlike the existing algorithm in [19–21]. The higher computational complexity is about $3n^3$ multiplications and divisions, that is, less than GH-algorithm [22], which requires $4.5n^3$ multiplications and divisions, when they are applied to the case of $r \approx n = m$ for $A^\dagger$.

Remark 10. In order to improve the accuracy of the algorithm, we must select nonzero entries in pivot row and column in each step of the Gauss-Jordan elimination. This improvement is based on the fact that Gauss-Jordan elimination is applied on matrices containing nonnegligible number zero elements.

4. Numerical Examples

In this section, we will use a numerical example to demonstrate our results. A handy method is used to compute $A^\dagger$ on a low order matrix.

Example 1. Use Algorithm 7 to compute the M-P inverse $A^\dagger$ of the matrix in [21], where

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 2 & 2 & 1 & 1 \end{pmatrix}. \tag{22}$$

Solution. Execute elementary row operations on the first four rows of the partitioned matrix $M_1 = \left(\begin{smallmatrix} 0 & A^* \\ A^* & 0 \end{smallmatrix}\right)$; we have

$$\begin{aligned} M_1 &= \begin{pmatrix} 0 & A^* \\ A^* & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\ \longrightarrow M_2 &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{23}$$

Then perform elementary column operations on the first three columns of matrix M_2, which yields

$$M_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\longrightarrow M_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{24}$$

Denote

$$B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -\frac{1}{2} & 0 \\ 1 & -\frac{1}{2} & 0 \end{pmatrix}. \tag{25}$$

By computing, we have

$$\begin{aligned} D &= BAC \\ &= \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 2 & 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -\frac{1}{2} & 0 \\ 1 & -\frac{1}{2} & 0 \end{pmatrix} \\ &= \begin{pmatrix} 7 & 0 & 0 \\ 5 & 3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{26}$$

We execute elementary row operations on the first two rows of the partitioned matrix $M_4 = \left(\begin{smallmatrix} D & B \\ C & 0 \end{smallmatrix}\right)$ again; we have

$$M_4 = \begin{pmatrix} 7 & 0 & 0 & 1 & 0 & 1 \\ 5 & 3 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\longrightarrow M_5 = \begin{pmatrix} 1 & 0 & 0 & \frac{1}{7} & 0 & \frac{1}{7} \\ 0 & 1 & 0 & -\frac{5}{21} & \frac{1}{3} & \frac{2}{21} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{27}$$

One then resumes elementary row and column operations on M_5, which results in

$$M_5 = \begin{pmatrix} 1 & 0 & 0 & \frac{1}{7} & 0 & \frac{1}{7} \\ 0 & 1 & 0 & -\frac{5}{21} & \frac{1}{3} & \frac{2}{21} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\longrightarrow M_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{7} & 0 & -\frac{1}{7} \\ 0 & 0 & 0 & \frac{5}{21} & -\frac{1}{3} & -\frac{2}{21} \\ 0 & 0 & 0 & -\frac{11}{42} & \frac{1}{6} & -\frac{2}{21} \\ 0 & 0 & 0 & -\frac{11}{42} & \frac{1}{6} & -\frac{2}{21} \end{pmatrix}. \tag{28}$$

Then we can obtain

$$A^{\dagger} = \begin{pmatrix} \frac{1}{7} & 0 & \frac{1}{7} \\ -\frac{5}{21} & \frac{1}{3} & \frac{2}{21} \\ \frac{11}{42} & -\frac{1}{6} & \frac{2}{21} \\ \frac{11}{42} & -\frac{1}{6} & \frac{2}{21} \end{pmatrix}. \qquad (29)$$

Conflict of Interests

The author declares that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

This project was supported by NSF China (no. 11226200), An-hui Provincial Natural Science Foundation (no. 10040606Q47), and The University Natural Science Research key Project of Anhui Province (KJ2013A204).

References

[1] E. H. Moore, "On the reciprocal of the general algebra matrix," *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395, 1920.

[2] R. Penrose, "A generalized inverse for matrices," *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 406–413, 1955.

[3] A. Ben-Israel and T. N. E. Greville, *Generalized Inverse Theory and Applications*, Springer, New York, NY, USA, 2nd edition, 2003.

[4] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Dover, New York, NY, USA, 1979.

[5] G. W. Stewart, "On the continuity of the generalized inverse," *SIAM Journal on Applied Mathematics*, vol. 17, pp. 33–45, 1969.

[6] G. W. Stewart, "On the perturbation of pseudo-inverses, projections and linear least squares problems," *SIAM Review*, vol. 19, no. 4, pp. 634–662, 1977.

[7] P.-Å. Wedin, "Perturbation theory for pseudo-inverses," *BIT Numerical Mathematics*, vol. 13, no. 2, pp. 217–232, 1973.

[8] J. Ji, "The algebraic perturbation method for generalized inverses," *Journal of Computational Mathematics*, vol. 7, no. 4, pp. 327–333, 1989.

[9] L. Kramarz, "Algebraic perturbation methods for the solution of singular linear systems," *Linear Algebra and Its Applications*, vol. 36, pp. 79–88, 1981.

[10] J. Miao and A. Ben-Israel, "Minors of the Moore-Penrose inverse," *Linear Algebra and Its Applications*, vol. 195, pp. 191–207, 1993.

[11] J. Cai and G. L. Chen, "On the representation of A^{+}, A^{+}_{MN} and its applications," *Numerical Mathematics*, vol. 24, no. 4, pp. 320–326, 2002 (Chinese).

[12] J. A. Fill and D. E. Fishkind, "The Moore-Penrose generalized inverse for sums of matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 2, pp. 629–635, 2000.

[13] J. Ji, "Explicit expressions of the generalized inverses and condensed Cramer rules," *Linear Algebra and Its Applications*, vol. 404, no. 1–3, pp. 183–192, 2005.

[14] J.-F. Cai, M. K. Ng, and Y.-M. Wei, "Modified Newton's algorithm for computing the group inverses of singular Toeplitz matrices," *Journal of Computational Mathematics*, vol. 24, no. 5, pp. 647–656, 2006.

[15] X. Chen and J. Ji, "Computing the Moore-Penrose inverse of a matrix through sysmmetric rank one updates," *American Journal of Computational Mathematics*, vol. 1, pp. 147–151, 2011.

[16] M. D. Petkovi and P. S. Stanimirovi, "Iterative method for computing the MoorePenrose inverse based on Penrose equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 6, pp. 1604–1613, 2011.

[17] Y. Wei, J. Cai, and M. K. Ng, "Computing Moore-Penrose inverses of Toeplitz matrices by Newton's iteration," *Mathematical and Computer Modelling*, vol. 40, no. 1-2, pp. 181–191, 2004.

[18] V. N. Katsikis, D. Pappas, and A. Petralias, "An improved method for the computation of the Moore-Penrose inverse matrix," *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9828–9834, 2011.

[19] X. Sheng and G. Chen, "A note of computation for M-P inverse $A^{\dagger}$," *International Journal of Computer Mathematics*, vol. 87, no. 10, pp. 2235–2241, 2010.

[20] J. Ji, "Gauss-Jordan elimination methods for the Moore-Penrose inverse of a matrix," *Linear Algebra and Its Applications*, vol. 437, no. 7, pp. 1835–1844, 2012.

[21] P. S. Stanimirović and M. D. Petković, "Gauss-Jordan elimination method for computing outer inverses," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4667–4679, 2013.

[22] W. Guo and T. Huang, "Method of elementary transformation to compute Moore-Penrose inverse," *Applied Mathematics and Computation*, vol. 216, no. 5, pp. 1614–1617, 2010.