Filomat 30:14 (2016), 3759–3770
DOI 10.2298/FIL1614759Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Two Methods for Computing the Drazin Inverse through Elementary Row Operations

Xiaodan Zhang[a], Xingping Sheng[b]

[a] *School of Information and Computer,Anhui Agricultural University, Hefei 230036, P.R.China*
[b] *School of Mathematics and Statistics, Fuyang Normal College, Fuyang 236037, P.R.China*

Abstract. In this paper, Let the matrix $A \in C^{n\times n}$ with $Ind(A) = k$, we first construct two bordered matrices based on [32], which gave a method for computing the null space of A^k by applying elementary row operations on the pair $\begin{pmatrix} A & I \end{pmatrix}$. Then two new Algorithms to compute the Drazin inverse A^d are presented based on elementary row operations on two partitioned matrices. The computational complexities of the two Algorithms are detailed analyzed. When the index $k = Ind(A) \geq 5$, the two Algorithms are all faster than the Algorithm by Anstreicher and Rothblum [32]. In the end, an example is presented to demonstrate the two new algorithms.

1. Introduction

Throughout the paper we shall use the notation of [1,2,3]. The symbol $C_r^{m\times n}$ denotes the set of all $m \times n$ complex matrices with rank r, C^n stands for the n dimensional complex space. I denotes the identity matrix. For $A \in C^{m\times n}$, the symbols $R(A)$, $N(A)$, $A^\dagger$, A^* and $r(A)$ denote its range, null space, M-P inverse, the conjugate transpose and rank, respectively. Here we recall that the index of $A \in C^{n\times n}$, denoted by $Ind(A)$, is the smallest nonnegative integer k such that $r(A^k) = r(A^{k+1})$.

In 1958 Drazin [4] showed that for any square $A \in C^{n\times n}$, there exists an unique matrix $X \in C^{n\times n}$ satisfying the following three equations

$$\begin{aligned} A^kXA &= A^k \qquad (1^k)\\ XAX &= X \qquad (2)\\ AX &= XA \qquad (5)\end{aligned}$$

where $k = Ind(A)$. This X is called the Drazin inverse of A and denoted by A^d. In particular, if $Ind(A) \leq 1$, the Drazin inverse is called the group inverse of A, denoted by A^g. Let $A \in C_r^{m\times n}$, T be a subspace of C^n of dimension $s \leq r$ and S be a subspace of C^m of dimension $m - s$ such that

$$AT \oplus S = C^m. \tag{1.1}$$

2010 *Mathematics Subject Classification*. Primary 15A09; Secondary 65F05
Keywords. keywords, Gauss-Jordan elimination, Drazin inverse, Elementary row operations, bordered matrix
Received: 25 October 2014; Accepted: 20 November 2015
Communicated by Dragana Cvetković-Ilić
Corresponding author: Xingping Sheng
Research supported by Anhui Provincial Natural Science Foundation(no.1508085MA12), Key projects of Anhui Provincial University excellent talent support program(no.gxyqZD2016188) and The University Natural Science Research key Project of Anhui Province (no.KJ2015A161)
Email addresses: zxd@ahau.edu.cn (Xiaodan Zhang), xingpingsheng@163.com (Xingping Sheng)

Then there exists an unique matrix X such that $XAX = X$ with $R(X) = T$ and $N(X) = S$. This matrix X is called the outer inverse, or {2}-inverse, of A with prescribed range T and null space S and denoted by $A^{(2)}_{T,S}$. In addition, suppose the matrix G satisfies $R(G) = T$ and $N(G) = S$, it is well know that

$$A^{(2)}_{T,S} = \begin{cases} A^{\dagger} & if \quad G = A^* \\ A^d & if \quad G = A^k \end{cases} \tag{1.2}$$

These concepts and properties can be found in the famous books [1, 2, 3].

The Drazin inverse occurs in a number of applications, for instance, finite Markov chains [5], singular differential and difference equations [2], multibody system dynamics [6] and so on.

In the latest fifty years, there have been many famous specialists and scholars, who investigated the Drazin inverse A^d. Its perturbation theories were introduced in [7-15]. The research on the representations of the Drazin inverse for block matrices can be seen in [16-22]. Many representations and computations for the Drazin inverse of a square matrix have also been widely researched [23-31].

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 $\begin{pmatrix} A & I \end{pmatrix}$ to transform it into $\begin{pmatrix} I & A^{-1} \end{pmatrix}$. Moreover Gauss-Jordan elimination can be used to determine whether or not a matrix is nonsingular. However, one can not directly use this method to compute Drazin inverse A^d on a square singular matrix A.

1987 Anstreicher and Rothblum [32] used this way to compute the index, generalized null spaces, and Drazin inverse (The idea will be recalled in the second section) . Recently, the authors [33-36] used Gauss-Jordan elimination methods to compute the $A^{\dagger}$ and $A^{(2)}_{T,S}$, respectively. More recently, these algorithms were further improved by Ji [37, 38], P.S. Stanimirovic and M.D. Petkovic [39].

In [33, 34], the first author, Chen and Gong proposed an algorithm for computing M-P inverse $A^{\dagger}$ and the outer inverse $A^{(2)}_{T,S}$ starts from elementary row operations on the pair $\begin{pmatrix} GA & I \end{pmatrix}$. Then, Ji [37], Stanimirovic and Petkovic [39] proposed an alternative explicit expressions for $A^{\dagger}$ and $A^{(2)}_{T,S}$, respectively. These methods begin with the elementary row operations on the pair $\begin{pmatrix} G & I \end{pmatrix}$ and do not need to compute A^*A or GA. More recently the first Author and Chen [35] start with the elementary row and column operations on the partitioned matrix $\begin{pmatrix} GAG & G \\ G & 0 \end{pmatrix}$ for computing $A^{(2)}_{T,S}$, then in [36] the author improved the algorithm [35] to compute M-P inverse $A^{\dagger}$. In [38] Ji proposed a new method for computing the outer inverse $A^{(2)}_{T,S}$ (The algorithm will be also restated in the second section) by applying elementary row operations also on the pair $\begin{pmatrix} G & I \end{pmatrix}$.

All algorithm for computing the out inverse $A^{(2)}_{T,S}$ need to know the matrix G. But for singular square matrix $A \in C^{n\times n}$ with $Ind(A) = k$ to compute Drazin inverse A^d, the matrix G satisfied $R(G) = R(A^k) = T$ and $N(G) = N(A^k) = S$ is difficult to find without known the $Ind(A)$. If we know the $Ind(A) = k$, these methods not only increase the computational cost to compute the A^k, but also it worsen the condition number.

In this paper, we will propose two alternative methods of elementary row operations for Drazin A^d by applying row operations first on $\begin{pmatrix} A & I \end{pmatrix}$, second on $\begin{pmatrix} A^* & I \end{pmatrix}$. Our approach is like the one in [36, 38] by working a bordered matrix and the Drazin is easy read off from the computed result but there is no need for forming A^k.

The paper is organized as follows. The ideals of computational A^d in [32] and $A^{(2)}_{T,S}$ in [38] are repeated in the next section. In section 3, we derive two novel explicit expressions for A^d, propose two like Gauss-Jordan elimination procedure for A^d based on the formula. In section 4, An illustrative example are presented to explain the corresponding improvements of the algorithm.

2. Preliminaries

The following two lemmas will be used repeatedly in the following sections.

Lemma 2.1[3] let $A \in C_r^{n\times n}$ with $Ind(A) = k$ and $r(A^k) = s \le r$, and $U, V^* \in C_{n-s}^{n\times(n-s)}$ be matrices whose column form bases for $N(A^k)$ and $N(A^{k^*})$ respectively. Then

$$D = \begin{pmatrix} A & U \\ V & 0 \end{pmatrix} \tag{2.1}$$

is nonsingular and

$$D^{-1} = \begin{pmatrix} A^d & U(VU)^{-1} \\ (VU)^{-1}V & -(VU)^{-1}VAU(VU)^{-1} \end{pmatrix} \tag{2.2}$$

Lemma 2.2[23] Let $A \in C_r^{n\times n}$ with $Ind(A) = k$ and $r(A^k) = s \le r$, $A^k = PQ$ is a full-rank factorization of A^k. Then

(1)QAP is an invertible complex matrix.

(2)$A^d = P(QAP)^{-1}Q$.

In [32], Anstreicher and Rothblum begin with elementary row operations on the pair $\begin{pmatrix} A & I \end{pmatrix}$ to compute the index, generalized null spaces, and Drazin inverse. Here we repeat the ideal of their algorithm in detail as following.

Consider a square $A \in C^{n\times n}$ with $Ind(A) = k$. In the course of the algorithm a sequence of pairs of matrices $\begin{pmatrix} A^{(i)} & B^{(i)} \end{pmatrix}$ are generated, where $\begin{pmatrix} A^{(0)} & B^{(0)} \end{pmatrix} = \begin{pmatrix} A & I \end{pmatrix}$. Given $\begin{pmatrix} A^{(i)} & B^{(i)} \end{pmatrix}$, execute row operations on $A^{(i)}$ to convert it into a matrix whose nonzero rows are linearly independent; moreover, if $A^{(i)}$ is found to be nonsingular, the algorithm terminates. Simultaneously, execute the same row operations on $B^{(i)}$. Let $\overline{A}^{(i)}$ and $\overline{B}^{(i)}$ be the result of executing the above row operations on $A^{(i)}$ and $B^{(i)}$, respectively. If $\overline{A}^{(i)}$ has zero rows, exchange these rows with the corresponding rows of $\overline{B}^{(i)}$ and get $A^{(i+1)} = \begin{pmatrix} \overline{A}_1^{(i)} \\ \overline{B}_2^{(i)} \end{pmatrix} = \begin{pmatrix} A_1^{(i+1)} \\ A_2^{(i+1)} \end{pmatrix}$, $B^{(i+1)} = \begin{pmatrix} \overline{B}_1^{(i)} \\ 0 \end{pmatrix} = \begin{pmatrix} B_1^{(i+1)} \\ 0 \end{pmatrix}$, then proceed to iteration $i+1$. The authors show that if k is the index of A, then the algorithm will always terminate on exactly the kth iteration. Moreover, the rows shuffled on iterations $0, \dots, i-1$, for $i = 1, \dots, k$, are a basis of the left null space of A^i. In addition, the authors also show that if on iteration k, $A^{(k)}$ is transformed into the identity matrix, i.e., $\overline{A}^{(k)} = I$, and $\widehat{A}$ is defined to be the resulting matrix $\overline{B}^{(k)}$, then the Drazin inverse of A is equal to $\widehat{A}^{k+1}A^k$.

Anstreicher and Rothblum's results are summarized in the following Algorithm:

Algorithm 2.1 Drazin inverse AR-Algorithm is stated as follows:

(1) In put $A \in C^{n\times n}$ with $Ind(A) = k$;

(2) Perform elementary row operations on the pair $\begin{pmatrix} A^{(i)} & B^{(i)} \end{pmatrix}$ into $\begin{pmatrix} \overline{A}^{(i)} & \overline{B}^{(i)} \end{pmatrix}$, where $\begin{pmatrix} A^{(0)} & B^{(0)} \end{pmatrix} = \begin{pmatrix} A & I \end{pmatrix}$

(3) If $A^{(i)}$ is nonsingular, then $\overline{A}^{(i)} = I$ and $\overline{B}^{(i)} = \widehat{A}$ then stop; else, $i = i+1, i = 0, 1, \dots, k$;

(4) Compute the output $A^d = \widehat{A}^{k+1}A^k$.

Algorithm 2.1 also generates the basis of the left null space of A^k, which is restated as the following lemma.

Lemma 2.3[32] Let $A \in C^{n\times n}$ with $Ind(A) = k$ and and $r(A^k) = s \le r$. Suppose that the Algorithm 2.1 is applied to A, then the algorithm terminates on iteration k. Furthermore, for $i = 1, \dots, k$, the union of the rows of $A_2^{(1)}, \dots, A_2^{(k)}$ is linearly independent and forms a basis of $\mathrm{null}(A^k)^T$.

In [32], Anstreicher and Rothblum also studied the computational complexity of the shuffle algorithm 2.1. The upper bound on the total number of arithmetic operations required to execute the algorithm is

$n^3 + nN(n - \frac{N}{k})$, where $N = n - s$. However we confirm that the upper bound is $2kn^3 + nN(n - \frac{N}{k})$ because from the value $A^d = \widehat{A^{k+1}}A^k$ of last step, $2k + 1$ matrices are multiplied.

Lemma 2.4[32] Let $A \in C^{n\times n}$ with $Ind(A) = k$ and and $r(A^k) = s \leq r$ (or $N = dim(nillA^k) = n - s$). Suppose that the Algorithm 2.1 is applied to A, with the algorithm terminates on iteration k. Furthermore, suppose that the algorithm is implemented so that $\overline{A}(i)$ is the row reduced echelon form of $A^{(i)}.i = 0, 1, \ldots, k$. Then an upper bound on the total number of arithmetic operations required to execute the algorithm is $2kn^3 + nN(n - \frac{N}{k})$.

In [38] Ji proposed a new method for computing the outer inverse $A^{(2)}_{T,S}$ by applying elementary row operations also on the pair $\begin{pmatrix} G & I \end{pmatrix}$. Here we will review the ideas for computing A^d.

He shows that there exists elementary matrix $P \in C^{n\times n}$ such that

$$P\begin{pmatrix} A^{k^*} & I \end{pmatrix} = \begin{pmatrix} PA^{k^*} & P \end{pmatrix} = \begin{pmatrix} B & I \end{pmatrix} \tag{2.3}$$

where $B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}$ and $B_1 \in C^{s\times n}_s$.

Then he applies elementary row operations on the pair $\begin{pmatrix} B^* & I \end{pmatrix}$, or equivalently there exists a nonsingular matrix $Q \in C^{n\times n}$ such that

$$Q^*\begin{pmatrix} B^* & I \end{pmatrix} = \begin{pmatrix} C & Q^* \end{pmatrix} \tag{2.4}$$

where $C = \begin{pmatrix} I_s & 0 \\ 0 & 0 \end{pmatrix}$. If the matrices P and Q are partitioned into

$$P^* = \begin{pmatrix} P_1 & P_2 \end{pmatrix} \quad and \quad Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \tag{2.5}$$

where $P_2 \in C^{m\times(m-s)}_{m-s}$ and $Q_2 \in C^{n\times(n-s)}_{n-s}$, then

$$R(P_2) = N(A^k) \quad and \quad N(Q_2) = R(A^k). \tag{2.6}$$

According to the Lemma 2.1, we know the bordered matrix (2.1) becomes

$$D = \begin{pmatrix} A & P_2 \\ Q_2 & 0 \end{pmatrix}. \tag{2.7}$$

We can compute the inverse D^{-1} by applying the Gauss-Jordan elimination procedure to the matrix $\begin{pmatrix} D & I \end{pmatrix}$ and read off the A^d from the inverse D^{-1}.

The above procedure for computing the Drazin inverse A^d using Gauss-Jordan elimination will be described as follows:

Algorithm 2.2 Drazin inverse Ji-Algorithm is stated as follows:

(1) In put $A \in C^{n\times n}$, compute $Ind(A) = k$, A^k and $r(A^k) = s$;

(2) Execute elementary row operations on $\begin{pmatrix} A^{k^*} & I \end{pmatrix}$ to get $\begin{pmatrix} B & P \end{pmatrix}$ where $B \in C^{n\times n}$ is in the reduced row echelon form;

(3) Execute elementary row operations on $\begin{pmatrix} B^* & I \end{pmatrix}$ to get $\begin{pmatrix} C & Q^* \end{pmatrix}$ where $C = \begin{pmatrix} I_s & 0 \\ 0 & 0 \end{pmatrix}$;

(4) Partition P and Q according to (2.5) and form the matrix D in (2.7);

(5) Perform elementary row operations on the matrix $\begin{pmatrix} D & I \end{pmatrix}$ until $\begin{pmatrix} I & D^{-1} \end{pmatrix}$ is reached and return the submatrix of D^{-1} consisting of the first n rows and the first n columns, i.e., A^d.

3. Main Results

Using algorithm 2.1 or algorithm 2.2 for computing the Drzain inverse A^d, we must to calculate the matrix A^k. In this section, we will propose two like Gauss-Jordan methods to compute A^d,which is not need to compute A^k, then summary two algorithms of these methods.

Theorem 3.1 Let $A \in C^{n\times n}$ with $Ind(A) = k$ and $r(A^k) = s$, the two sequence matrices $\{A_i^{(2)}\}, i = 1, 2, \ldots, k$ and $\{A_i^{*(2)}\}, i = 1, 2, \ldots, k$ are generated by applying Algorithm 2.1 to A and A^*, respectively. If we denote $B = \begin{pmatrix} A_1^{(2)} \\ A_2^{(2)} \\ \vdots \\ A_k^{(2)} \end{pmatrix}$, $C^* = \begin{pmatrix} A_1^{*(2)} \\ A_2^{*(2)} \\ \vdots \\ A_k^{*(2)} \end{pmatrix}$ and $M = \begin{pmatrix} A & C \\ B & 0 \end{pmatrix}$. Then

(1) Matrices $B \in C_{n-s}^{(n-s)\times n}$ and $C \in C_{n-s}^{n\times(n-s)}$ are all full rank, further $BA^k = 0$ and $A^kC = 0$, or Equivalent to

$$N(B) = R(A^k) \quad and \quad R(C) = N(A^k). \tag{3.1}$$

(2) M is invertible matrix and

$$M^{-1} = \begin{pmatrix} A^d & C(BC)^{-1} \\ (BC)^{-1}B & -(BC)^{-1}BAC(BC)^{-1} \end{pmatrix}. \tag{3.2}$$

Proof From Algorithm 2.1, Lemma 2.1 and lemma 2.3, we know the above result is correct.

In summary of the above Theorem, we have the following Algorithm for computing A^d.

Algorithm 3.1 Drazin inverse ZS-Algorithm 1:

(1) In put $A \in C^{n\times n}$ with $Ind(A) = k$;

(2) Perform elementary row operations on the pair $\begin{pmatrix} A^{(i)} & B^{(i)} \end{pmatrix}$ into $\begin{pmatrix} \overline{A}^{(i)} & \overline{B}^{(i)} \end{pmatrix}(i = 0, 1, \ldots, k)$, where $\begin{pmatrix} A^{(0)} & B^{(0)} \end{pmatrix} = \begin{pmatrix} A & I \end{pmatrix}$ to generate the sequence matrices $\{A_i^{(2)}\}, i = 1, 2, \ldots, k$, denote $B = \begin{pmatrix} A_1^{(2)} \\ A_2^{(2)} \\ \vdots \\ A_k^{(2)} \end{pmatrix}$;

(3) Perform elementary row operations on the pair $\begin{pmatrix} A^{*(i)} & B^{*(i)} \end{pmatrix}$ into $\begin{pmatrix} \overline{A}^{*(i)} & \overline{B}^{*(i)} \end{pmatrix}(i = 0, 1, \ldots, k)$, where $\begin{pmatrix} A^{*(0)} & B^{*(0)} \end{pmatrix} = \begin{pmatrix} A^* & I \end{pmatrix}$ to generate the sequence matrices $\{A_i^{*(2)}\}, i = 1, 2, \ldots, k$, denote $C^* = \begin{pmatrix} A_1^{*(2)} \\ A_2^{*(2)} \\ \vdots \\ A_k^{*(2)} \end{pmatrix}$;

(4) Form the partitioned matrix $M = \begin{pmatrix} A & C \\ B & 0 \end{pmatrix}$;

(5) Perform elementary row operations on the matrix $\begin{pmatrix} M & I \end{pmatrix}$ until $\begin{pmatrix} I & M^{-1} \end{pmatrix}$ is reached and return the submatrix of M^{-1} consisting of the first n rows and the first n columns, i.e., A^d.

Here, an example is given to demonstrate the process of computing the matrices B and C. Take matrix A from [32], where

$$A = \begin{pmatrix} 2 & 4 & 6 & 5 \\ 1 & 4 & 5 & 4 \\ 0 & -1 & -1 & 0 \\ -1 & -2 & -3 & -3 \end{pmatrix}.$$

Elementary row operations transform $\begin{pmatrix} A & I \end{pmatrix}$ into

$$\left(\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & -2 & -2 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 3 \end{array}\right).$$

we exchange row of zeros with the corresponding row of the right hand matrix. This yields

$$\left(\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & -2 & -2 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 2 & 3 & 0 & 0 & 0 & 0 \end{array}\right).$$

One then resumes elementary row operations, which result in

$$\left(\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & -2 & -2 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right).$$

A second exchange row of zeros with the corresponding row of the right hand matrix

$$\left(\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & -2 & -2 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \end{array}\right).$$

Elementary row operations are now finally used to convert the left hand matrix into the identity, yielding

$$\left(\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 1 & -3 & -4 & -2 \\ 0 & 1 & 0 & 0 & 0 & -1 & -3 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \end{array}\right).$$

Denote $B = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 0 & 0 & -1 & 1 \end{pmatrix}$, from Algorithm 2.1 and Theorem 3.1 we know $Ind(A) = 2$ and $N(B) = R(A^2)$. We easy to check $BA^2 = 0$.

Similar, if we perform the above procedure on the pair $\begin{pmatrix} A^* & I \end{pmatrix}$, the matrix $C = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}$ is obtained, C satisfy $R(C) = N(A^2)$ or $A^2C = 0$.

According to Theorem 3.1, we know that B and C are full row rank and full column rank, respectively. We begin with the elementary row operations on $\begin{pmatrix} B^* & I \end{pmatrix}$. Let F be the product of all the elementary matrices representing these elementary row operations. we can write

$$F\begin{pmatrix} B^* & I \end{pmatrix} = \begin{pmatrix} FB^* & F \end{pmatrix} = \begin{pmatrix} \widetilde{B}^* & F \end{pmatrix} \tag{3.3}$$

where $\widetilde{B}^* = \begin{pmatrix} I_{n-s} \\ 0 \end{pmatrix}$.

If we start with the elementary row operations on $\begin{pmatrix} C & I \end{pmatrix}$. Let G be the product of all the elementary matrices representing these elementary row operations. we can write

$$G\begin{pmatrix} C & I \end{pmatrix} = \begin{pmatrix} GC & G \end{pmatrix} = \begin{pmatrix} \widetilde{C} & G \end{pmatrix} \tag{3.4}$$

where $\widetilde{C} = \begin{pmatrix} I_{n-s} \\ 0 \end{pmatrix}$.

Theorem 3.2 Let $A \in C^{n\times n}$ with $Ind(A) = k$ and $r(A^k) = s$, the two matrices B and C are generated by applying Algorithm 2.1 to A and A^*, respectively. F and G are two nonsingular matrices such that (3.3) and (3.4). If the matrices F and G are partitioned into

$$F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \quad and \quad G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \tag{3.5}$$

where $F_2^* \in C_s^{n\times s}$ and $G_2 \in C_s^{s\times n}$, such that

$$R(F_2^*) = N(B) = R(A^k) \quad and \quad N(G_2) = R(C) = N(A^k). \tag{3.6}$$

Further, we have

$$A^d = F_2^*(G_2AF_2^*)^{-1}G_2 \tag{3.7}$$

Proof In view of (3.3) and (3.4), we can write

$$\widetilde{B^*} = \begin{pmatrix} I_{n-s} \\ 0 \end{pmatrix} = FB^* = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} B^* = \begin{pmatrix} F_1B^* \\ F_2B^* \end{pmatrix} \tag{3.8}$$

and

$$\widetilde{C} = \begin{pmatrix} I_{n-s} \\ 0 \end{pmatrix} = GC = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} C = \begin{pmatrix} G_1C \\ G_2C \end{pmatrix} \tag{3.9}$$

By comparing both sideS (3.8) and (3.9), we have $F_2B^* = 0$ and $G_2C = 0$. This shows $BF_2^* = 0$.

Thus we have

$$R(F_2^*) \subset N(B) \quad and \quad R(C) \subset N(G_2). \tag{3.10}$$

Notice that

$$dim[R(F_2^*)] = s = dim[R(A^k)] = dim[N(B)] \tag{3.11}$$

and

$$dim[N(G_2^*)] = n - s = dim[N(A^k)] = dim[R(C)]. \tag{3.12}$$

From (3.10), (3.11) and (2.12), we know (3.6) is right.

Following form (3.6) and lemma 2.2, we have $A^d = F_2^*(G_2AF_2^*)^{-1}G_2$

According to the representation of A^d introduced in Theorem 3.2, we summary the following Algorithm for computing Drazin inverse A^d

Algorithm 3.2 Drazin inverse-ZS Algorithm 2:

(1) In put $A \in C^{n\times n}$ with $Ind(A) = k$;

(2) Perform elementary row operations on the pair $\begin{pmatrix} A^{(i)} & B^{(i)} \end{pmatrix}$ into $\begin{pmatrix} \overline{A}^{(i)} & \overline{B}^{(i)} \end{pmatrix}$($i = 0, 1, \ldots, k$), where $\begin{pmatrix} A^{(0)} & B^{(0)} \end{pmatrix} = \begin{pmatrix} A & I \end{pmatrix}$ to generate the sequence matrices $\{A_i^{(2)}\}, i = 1, 2, \ldots, k$, denote $B = \begin{pmatrix} A_1^{(2)} \\ A_2^{(2)} \\ \vdots \\ A_k^{(2)} \end{pmatrix}$;

(3) Perform elementary row operations on the pair $\begin{pmatrix} A^{*(i)} & B^{*(i)} \end{pmatrix}$ into $\begin{pmatrix} \overline{A}^{*(i)} & \overline{B}^{*(i)} \end{pmatrix}(i = 0, 1, \ldots, k)$, where $\begin{pmatrix} A^{*(0)} & B^{*(0)} \end{pmatrix} = \begin{pmatrix} A^{*} & I \end{pmatrix}$ to generate the sequence matrices $\{A_i^{*(2)}\}, i = 1, 2, \ldots, k$, denote $C^* = \begin{pmatrix} A_1^{*(2)} \\ A_2^{*(2)} \\ \vdots \\ A_k^{*(2)} \end{pmatrix}$;

(4) Execute elementary roe operations on $\begin{pmatrix} B^* & I \end{pmatrix}$ and $\begin{pmatrix} C & I \end{pmatrix}$ to get $F_2^* \in C_s^{n\times s}$ and $G_2 \in C_s^{s\times n}$, such that $R(F_2^*) = N(B) = R(A^k)$ *and* $N(G_2) = R(C) = N(A^k)$;

(5) Compute $G_2AF_2^*$ and form the block matrix

$$N_1 = \begin{pmatrix} G_2AF_2^* & G_2 \\ F_2^* & 0 \end{pmatrix} \longrightarrow N_2 = \begin{pmatrix} I_s & (G_2AF_2^*)^{-1}G_2 \\ F_2^* & 0 \end{pmatrix};$$

(6) Make the block matrices of $N_2(1, 2)$ and $N_2(2, 1)$ be zero matrices by applying elementary row and column transformations, respectively, through matrix I_s, which yields

$$N_3 = \begin{pmatrix} I_s & 0 \\ 0 & -F_2^*(G_2AF_2^*)^{-1}G_2 \end{pmatrix}$$

Then read off $A^d = F_2^*(G_2AF_2^*)^{-1}G_2$.

4. Computational Complexities

We only count the multiplications and divisions. Let us first analysis the complexity of the algorithm 3.1.

The step 2 of algorithm 3.1 to get matrix B is the same as the step 2 and 3 of the Algorithm 2.1. The upper bound of the required arithmetic operations is $nN(n - \frac{N}{k})$. Following the same line the upper bound of the arithmetic operations for step 3 is also $nN(n - \frac{N}{k})$. The step 5 is to calculate the inverse matrix M^{-1}, the total operations is $(n + N)^3$.

Therefore, it requries

$$T_d(n, k, N) = 2nN(n - \frac{N}{k}) + (n + N)^3 \tag{4.1}$$

operations altogether for Algorithm 3.1 to compute the Drazin inverse A_d. If the matrix A is nonsingular, then $N = 0$ and $T_d(n, k, N) = n^3$ is the arithmetic operations of A^{-1}.

With fix n and k, $T_d(n, k, N)$ achieves its maximum vale at $N = n$. Hence we have

$$T_d(n, k, N) = 2nN(n - \frac{N}{k}) + (n + N)^3 \leq (10 - \frac{2}{k})n^3 \tag{4.2}$$

We have proved the following theorem:

Theorem 4.1 Let the square matrix A be same as the Lemma 2.4, it takes $T_d(n, k, N)$ divisions and multiplications for Algorithm 3.1 to compute the Drazin inverse where $T_d(n, k, N)$ is given in (4.1). Moreover $T_d(n, k, N) \leq (10 - \frac{2}{k})n^3$.

From Lemma 2.4 and Theorem 4.1, by a simple calculations we know that Algorithm 3.1 is faster than Algorithm 2.1 if $k \geq 5$.

The step 2 and step 3 of Algorithm 3.2 is the same as Algorithm 3.1. The upper bound of the required arithmetic operations for step 2 and setp3 is also $2nN(n - \frac{N}{k})$. In step 4, both $\begin{pmatrix} B^* & I \end{pmatrix}$ and $\begin{pmatrix} C & I \end{pmatrix}$ are $n\times(n+N)$ and require N pivoting steps. The first pivoting step on $\begin{pmatrix} B^* & I \end{pmatrix}$ involves $N+1$ nonzero columns and it requires N divisions and $(n-1)N$ multiplications with a total of nN operatons. The next each pivoting step also deals with $N + 1$ nonzero columns. Adding up, it takes nN^2 operations to compute the matrix F_2^*. Similarly, it also takes nN^2 operations to compute G_2.

In step 5, it requires $nN(n+N)$ multiplications to compute $G_2AF_2^*$. Since first pivoting step on $\begin{pmatrix} G_2AF_2^* & G_2 \end{pmatrix}$ involves $n+N$ nonzero columns and it requires $n+N-1$ divisions and $(n+N-1)(N-1)$ multiplications with a total of $(n+N-1)N$ operations. The second pivoting step deals with one less nonzero columns. It requires $n+N-2$ divisions and $(n+N-2)(N-1)$ multiplications with a total of $(n+N-2)N$. Continuing this way, the Nth pivoting step handles with $n+1$ nonzero columns and it requires n divisions and $n(N-1)$ multiplications with a total of nN. Adding up, it takes $(n+N-1)N+(n+N-2)N+\ldots+nN = \frac{nN(n+2N-1)}{2}$ operations to compute $(G_2AF_2^*)^{-1}G_2$.

Then resume elementary row and columns operations on the matrix N_2 to transform it into N_3. The complexity of this process is n^2N multiplications, which is the count to compute $F_2^*(G_2AF_2^*)^{-1}G_2$.

Hence, the total number of complexity of Algorithm 3.2 is

$$T'_d(N,N,k) = 2nN(n-\frac{N}{k}) + 2nN + nN(n+N) + \frac{nN(n+2N-1)}{2} + n^2N = \frac{7}{2}n^2N + 2nN(2-\frac{1}{k}) \tag{4.3}$$

Similarly, with fix n and k, $T'_d(n,k,N)$ achieves its maximum vale at $N=n$. Hence we have

$$T'_d(n,k,N) = \frac{7}{2}n^2N + 2nN(2-\frac{1}{k}) \le (\frac{15}{2}-\frac{2}{k})n^3 \tag{4.2}$$

We have proved the following theorem:

Theorem 4.2 Let the square matrix A be same as the Lemma 2.4, it takes $T'_d(n,k,N)$ divisions and multiplications for Algorithm 3.1 to compute the Drazin inverse where $T'_d(n,k,N)$ is given in (4.3). Moreover $T'_d(n,k,N) \le (\frac{15}{2}-\frac{2}{k})n^3$.

From Lemma 2.4 and Theorem 4.2, by a simple calculations we know that Algorithm 3.2 is also faster than Algorithm 2.1 if $k \ge 4$.

5. Numerical Examples

In this section, we shall use an example to demonstrate our results.

Example 1£ Use Algorithm 3.1 and Algorithm 3.2 to compute the Drazin inverse A^d of the matrix in [32] where

$$A = \begin{pmatrix} 2 & 4 & 6 & 5 \\ 1 & 4 & 5 & 4 \\ 0 & -1 & -1 & 0 \\ -1 & -2 & -3 & -3 \end{pmatrix}.$$

Solution First, we will use Algorithm 3.1 to compute Drazin A^d.

Using Algorithm 2.1, we obtain matrices B, C and $ind(A)=2$, through elementary row operations on $\begin{pmatrix} A & I \end{pmatrix}$ and $\begin{pmatrix} A^* & I \end{pmatrix}$, respectively. Which are demonstrated in the third section.

$$B = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 0 & 0 & -1 & 1 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}$$

where matrices B and C are all full rank and satisfied $N(B) = R(A^2)$ and $R(C) = N(A^2)$, respectively.

Next, we construct block matrix

$$M = \begin{pmatrix} A & C \\ B & 0 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 6 & 5 & 1 & 0 \\ 1 & 4 & 5 & 4 & 1 & 0 \\ 0 & -1 & -1 & 0 & -1 & 1 \\ -1 & -2 & -3 & -3 & 0 & -1 \\ 1 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}.$$

From Lemma 2.1 , we know that matrix M is nonsingular and A^d can be read off from M^{-1}.
Then we perform elementary row operations transform $\begin{pmatrix} M & I \end{pmatrix}$ into $\begin{pmatrix} I & M^{-1} \end{pmatrix}$.

$$\begin{pmatrix} M & I \end{pmatrix} \to \begin{pmatrix} I & M^{-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 3 & -1 & 2 & 2 & -2 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 2 & 1 & 3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 & 5 & 5 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -2 & -3 & 0 & 0 \end{pmatrix}.$$

This yields

$$A^d = \begin{pmatrix} 3 & -1 & 2 & 2 \\ 2 & 1 & 3 & 3 \\ -1 & 0 & -1 & -1 \\ -1 & 0 & -1 & -1 \end{pmatrix}.$$

Second, we will use Algorithm 3.2 to compute A^d.
By applying the elementary row operations on $\begin{pmatrix} C & I \end{pmatrix}$, we get

$$\begin{pmatrix} C & I \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Denote $G_2 = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$, we can easy to check that G_2 is full rank and $N(G_2) = R(B) = N(A^2)$.
Similar, we apply elementary row operations on $\begin{pmatrix} B^* & I \end{pmatrix}$, we have

$$\begin{pmatrix} B^* & E \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ 3 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & -5 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}.$$

Let $F_2^* = \begin{pmatrix} -5 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$, then F_2^* is also full rank and $R(F_2^*) = N(B) = R(A^2)$
By computing, we have

$$G_2AF_2^* = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & 6 & 5 \\ 1 & 4 & 5 & 4 \\ 0 & -1 & -1 & 0 \\ -1 & -2 & -3 & -3 \end{pmatrix} \begin{pmatrix} -5 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ -1 & 0 \end{pmatrix}.$$

According to Algorithm 3.2, we execute elementary row operations on the first two rows of the partitioned matrix $N_1 = \begin{pmatrix} G_2AF_2^* & G_2 \\ F_2^* & 0 \end{pmatrix}$ again, we have

$$N_1 = \begin{pmatrix} 3 & 1 & -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 1 \\ -5 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \to N_2 = \begin{pmatrix} 1 & 0 & -1 & 0 & -1 & -1 \\ 0 & 1 & 2 & 1 & 3 & 3 \\ -5 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

One then resume elementary row and column operations on N_2, which results in

$$N_2 = \begin{pmatrix} 1 & 0 & -1 & 0 & -1 & -1 \\ 0 & 1 & 2 & 1 & 3 & 3 \\ -5 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \to N_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 1 & -2 & -2 \\ 0 & 0 & -2 & -1 & -3 & -3 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Then we can obtain

$$A^d = \begin{pmatrix} 3 & -1 & 2 & 2 \\ 2 & 1 & 3 & 3 \\ -1 & 0 & -1 & -1 \\ -1 & 0 & -1 & -1 \end{pmatrix}.$$

References

[1] A. Ben-Israel and T.N.E. Greville, Generalized inverse Theory and Applications, 2nd Edition, NewYork: Springer Verlag, 2003.
[2] S.L. Campbell and C.D. Meyer, Generalized inverses of Linear Transformations, New York: Dover Publ., 1979.
[3] G.R. Wang, Y. Wei and S. Qiao, Generalized Inverses: Theory and Computa- tions, Science Press, Beijing/New York, China, 2004.
[4] M.P. Drazin, Pseudoinverse in associative rings and semigroups, Amer. Math. Monthly. 65(1958)506-514.
[5] C.D. Meyer, The condition of a finite Markov chain and perturbation bounds for the limiting probabilities, SIAM J. Algebraic Discrete Math., 1(1980)273-283.
[6] B. Simeon, C. F*ü*hrer and P. Rentrop, The Drazin inverse in multibody system dynamics, Numer. Math. 64(1993)521-539.
[7] N. Castro-Gonzalez, J. Robles, and J. Y. Velez-Cerrada, Characterizations of a class of matrices and perturbation of the Drazin inverse, SIAM J. Matrix Anal. Appl., 30(2008)882-897.
[8] N. Castro-Gonzalez and J. Y. Velez-Cerrada, On the perturbation of the group generalized inverse for a class of bounded operators in Banach spaces, J. Math. Anal. Appl., 34(2008)1213-1223.
[9] J.J. Koliha, Error bounds for a general perturbation of the Drazin inverse, Appl. Math. Comput., 126(2002)181-185.
[10] Y. Wei and G. Wang, The perturbation theory for the Drazin inverse and its application, Linear Algebra Appl., 258(1997)179-186.
[11] Y. Wei, The Drazin inverse of updating of a square matrix with application to perturbation formula, Appl. Math. Comput., 108(2000)77-83.
[12] Y. Wei, Perturbation bound of theDrazin inverse,Appl.Math. Comput., 125(2002)231-244.
[13] Y. Wei and X. Li, An improvement on perturbation bounds for the Drazin inverse, Numer. Linear Algebra Appl., 10(2003)563-575.
[14] Y. Wei, X. Li and F. Bu, A perturbation bound of the Drazin inverse of a matrix by separation of simple invariant subspaces, SIAM J. Matrix Anal. Appl., 27(2005)72-81.
[15] Q. Xu, C. Song and Y. Wei, The stable perturbation of the Drazin inverse of the square matrices, SIAM J. Matrix Anal. Appl., 31(2010)1507-1520.
[16] Y. Wei, Expressions for the Drazin inverse of 2×2 block matrix, Linear Multilinear Algebra., 45(1998)131-146.
[17] C. Deng and Y. Wei, Representations for the Drazin inverse of 2×2 block-operator matrix with singular Schur complement, Linear Algebra Appl., 435(2011)2766-2783.
[18] R.E. Hartwig, G. Wang and Y. Wei, Some additive results on Drazin inverse, Linear Algebra Appl., 322(2001)207-217.
[19] C.D. Meyer and N.J. Rose, The index and the Drazin inverse of block triangular matrices, SIAM J. Appl. Math., 33(1977)1-7.
[20] D. Mosić and D.S. Djordjević, Representation for the generalized Drazin inverse of block matrices in Banach algebras, Appl. Math. Comput., 218(2012)12001-12007.
[21] D. Mosić and D.S. Djordjević, Formulae for the generalized Drazin inverse of a block matrix in terms of Banachiewicz-Schur forms, J. Math. Anal. Appl., 413(2014)114-120.
[22] X. Sheng and G. Chen, Some generalized inverses of partitioned matrix and quotient identity of generalized Schur complement, Appl. Math. Comput., 196(2008)174-184.
[23] P.S.Ctanimirović and D.S.Djordjević, Full-rank and determininantal representation of the Drazin inverse. Linear Algerbra and Its Application. 311(2000)131-151.
[24] Y. Wei, A characterization and representation of the Drazin inverse, SIAM J. Matrix Anal. Appl. 17(1996)744-747.
[25] Y. Wei, Successive matrix squaring algorithm for computing the Drazin inverse, Appl. Math. Comput., 108(2000)2-3, 67-75.
[26] Y. Wei, H.Wu, The representation and approximation for Drazin inverse, J. Comput. Appl. Math., 126(2000)1-2,417-432.
[27] Y. Chen, Representation and approximation for the Drazin inverse A^d, Appl. Math. Comput., 119(2001)147-160.
[28] R.E. Hartwig, Schurs theorem and Drazin inverse, Pacific J. Math., 78(1978)133-238.
[29] R.E. Hartwig, A method for calculating A^d, Math. Japon., 26(1981)37-43.
[30] X. Li and Y. Wei, Iterative methods for the Drazin inverse of a matrix with a complex spectrum, Appl. Math. Comput., 147(2004)3, 855-862.
[31] X. Sheng and G. Chen, An oblique projection iterative method to compute Drazin inverse and group inverse, Appl. Math. Comput. 211(2009)2,417-421.

[32] K.M.Anstreicher and U.G. Rothblum, Using Gauss-Jordan elimination to compute the Index, generalized nulispaces and Drazin inverse, Linear Algebra Appl., 85(1987), 221-239.
[33] X. Sheng and G. Chen, A note of computation for M-P inverse $A^{\dagger}$, Int. J. Comput. Math., 87(2010), 2235-2241.
[34] X. Sheng, G. Chen and Y. Gong, The representation and computation of generalized inverse $A^{(2)}_{T,S}$, J. Comput. Appl. Math., 213(2008), 248-257.
[35] X. Sheng and G. Chen, Innovation based on Gaussian elimination to compute generalized inverse $A^{(2)}_{T,S}$, Comput. Math. Appl., 65 (2013) 1823-1829.
[36] X. Sheng, Execute elementary row and column operations on the partitioned matrix to compute M-P inverse $A^{\dagger}$, Abstr. Appl. Anal., Volume 2014, Article ID 596049, 6 pages.
[37] J. Ji, Gauss-Jordan elimination methods for the Moore-Penrose inverse of a matrix, Linear Algebra Appl. 437(2012), 1835-1844.
[38] J. Ji, Computing the outer and group inverses through elementary row operations, Comput. Math. Appl.(2014),http://dx.doi.org/10.1016/j.camwa.2014.07.016.
[39] P.S. Stanimirovic and M.D. Petkovic, Gauss-Jordan elimination method for computing outer inverses, Appl. Math. Comput., 219(2013), 4667-4679.