DOI: 10.2478/auom-2025-0023
An. Şt. Univ. Ovidius Constanţa

sciendo
Vol. 33(2),2025, 143–156

Evaluating $A_{T,S}^{(2)}$ Matrix Inverses from full-rank Singular Value Decomposition

Ivan Stanimirović

Abstract

Based on the compact SVD decomposition of the full rank of a predetermined matrix M, a method for calculating the inverses $A_{T,S}^{(2)}$ of matrix A is derived. As a consequence, the advantages of compact single value decompositions were developed with this new method. The method is then extended to a set of polynomial matrices. As a result, an algorithm is proposed for effective symbolic computation of $A_{T,S}^{(2)}$ inverses of a polynomial matrix. Some implementation details and comparative computation times compared to other similar methods are shown to illustrate the efficiency of the algorithm.

1 Introduction

The decomposition of a matrix into an appropriate canonical arrangement is named *matrix factorization.* There exist diverse matrix factorizations, utilized among specific classes of tasks. A matrix with elements of a commutative ring is named a symbolic matrix. The construction of matrices of minors is required for the task of inversion of a symbolic matrix [1]. To compute the generalized

Key Words: Generalized inverses, Singular Value Decomposition, MATHEMATICA, polynomial matrix, Hermitian matrix.

2010 Mathematics Subject Classification: Primary 15A23, 15A09; Secondary 68M40.

This research was supported by the Science Fund of the Republic of Serbia, GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications - QUAM. This research was supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-68/2020-14/200124.

Received: 04.07.2024

Accepted: 25.11.2024

inversions of A, the decomposition of the polynomial matrix A is frequently considered [9].

Applying a compact SVD decomposition of a matrix A to the representation of the generalized inverse $A_{T,S}^{(2)}$ from [10] is our central goal. The technique offered in [10] is shaped from the full-rank decomposition of A. Here we provide an addition to this procedure, appropriate not just for the scheming of the Moore-Penrose inverse but also for the broader set of $A_{T,S}^{(2)}$ inverses. We select the efficient compact SVD decomposition of the predetermined matrix M instead of the decomposition FG of A. Furthermore, in this paper, the expansion of this procedure is also established on a set of polynomial matrices with one unknown variable.

To implement generalized inversions of polynomial and rational matrices, SVD decomposition is a frequent choice (see [7]). In the compact SVD, the product $U_r\Sigma_r V_r^*$ is calculated, where only the r column vectors of U and the r row vectors of V^* (which correspond to nonzero singular values) are calculated (other vectors of U and $V*$ are not calculated). This is faster than a thin SVD if $r << n$. The matrix U_r is consequently of dimensions $m \times r$, while Σ_r is a diagonal $r \times r$ matrix, and V_r^* is $r \times n$ matrix. So, a compact SVD of $A(x)$ is a full rank factorization of the form

$$A(x) = U(x)\Sigma(x)V^*(x), \tag{1}$$

where $\Sigma(x)$ is diagonal matrix,

$$U(x) = (U_{ij}(x))_{i,j=1}^{m,r},\ V(x) = (V_{ij}(x))_{i,j=1}^{r,n} \tag{2}$$

and $u_{ij},\ \sigma_j,\ v_{ji},\ 1 \le j \le r$ are rational functions. In this notation, $V^*(x)$ means the conjugate transpose of $V(x)$.

As usual, the set of rational functions with complex coefficients is denoted by $\mathbb{C}(x)$, and the set of $m \times n$ matrices with elements in $\mathbb{C}(x)$ is denoted by $\mathbb{C}(x)^{m\times n}$. Observe the complex matrix $A(x) \in \mathbb{C}(x)^{n\times n}$ given in the polynomial formula with respect to the unidentified x:

$$A(x) = A_0 + A_1x + \cdots + A_qx^q = \sum_{i=0}^{q} A_ix^i \tag{3}$$

where A_i, $i = 0, \ldots, q$ are $n \times n$ constant matrices [2].

Several novel procedures are presented for the scheming of various categories of generalized inverses in [15, 16], based on the knowledge that the polynomial matrix may be described by a series of constant matrices. The complexity that increases with the degree of $A(x)$ is one of the complications that influence these methods.

The range of a matrix $A \in \mathbb{C}(x)^{m\times n}$ is indicated as $\mathbf{R}(A)$ and the null space of A as $\mathbf{N}(A)$. Furthermore, let us denote by $\mathrm{nr}(A)$ the *normal rank* of A (for example, the rank over the set $\mathbb{C}(x)$) and by $\mathbb{C}(x)^{m\times n}_r$ the class of matrices from $\mathbb{C}(x)^{m\times n}$ of the normal rank r. Moreover, let $\mathrm{rank}(A)$ denote the position of persistent matrix A and $\mathbb{C}^{m\times n}_r$ stands for the subset of complex $m \times n$ matrices $\mathbb{C}^{m\times n}$ with rank r. Now we recapture the meaning of $A^{(2)}_{T,S}$, summed in [10].

Definition 1.1. [10] *If $A \in \mathbb{C}^{m\times n}_r$, T is a subspace of $\mathbb{C}^n$ of dimension $t \le r$ and S is a subspace of $\mathbb{C}^m$ of dimension $m - t$, then A has a $\{2\}$-inverse X such that $\mathcal{R}(X) = T$ and $\mathcal{N}(X) = S$ if and only if $AT \oplus S = \mathbb{C}^m$, in which case X is unique and is denoted by $A^{(2)}_{T,S}$.*

Definition 1.2. [10] *In the case where $A \in \mathbb{C}^{m\times n}_r$, T is a subspace of $\mathbb{C}^n$ of measurement $t \le r$ and S is a subspace of $\mathbb{C}^m$ of measurement $m - t$, at that point A has a $\{2\}$ reverse X with the end goal that $\mathcal{R}(X) = T$ and $\mathcal{N}(X) = S$ if and only if $AT \oplus S = \mathbb{C}^m$, in which case X is remarkable and is indicated by $A^{(2)}_{T,S}$.*

Probably the most utilized cases of generalized inverses are outer inverses with predefined range and null space. Many important generalized inverses, like Moore-Penrose $X^\dagger$, weighted Moore-Penrose $X^\dagger_{M,N}$, Drazin X^D and group inverse $X^\#$, along with the Bott-Duffin inverse $X^{(-1)}_{(L)}$ and the generalized Bott-Duffin inverse $X^{(\dagger)}_{(L)}$, can be united and presented by generalized inverses $X^{(2)}_{T,S}$, where the corresponding matrices T and S are considered in [2]. For a given matrix $X \in \mathbb{C}^{m\times n}(x)$ the next presentations are valid (see [2, 3, 17, 18]):

$$\begin{aligned} &X^\dagger = X^{(2)}_{\mathcal{R}(X^*),\mathcal{N}(X^*)},\ X^\dagger_{M,N} = X^{(2)}_{\mathcal{R}(X^\sharp),\mathcal{N}(X^\sharp)},\ X^D = X^{(2)}_{\mathcal{R}(X^k),\mathcal{N}(X^k)},\\ &X^\# = X^{(2)}_{\mathcal{R}(X),\mathcal{N}(X)},\ X^{(-1)}_{(L)} = X^{(2)}_{L,L^\perp},\ X^{(\dagger)}_{(L)} = X^{(2)}_{S,S^\perp}. \end{aligned} \tag{4}$$

Our goal is to develop a new algorithm for symbolic computation of the $A^2_{T,S}$ inverse from SVD decomposition of polynomial matrices. A motivation is to apply a polynomial matrix factorization which is free of square roots of polynomials, thus avoiding problems with symbolic computations of such matrices arising from the existence of square roots of polynomials.

This work is structured as following. The calculation of outer inverses using compact SVD is first observed. Consequently, a new strategy is resulting grounded on the theorems from [10] using the compact singular value factorization. In Section 3, the aforementioned procedure is focused and specified on the class of polynomial matrices. Several numerical examples and comparison studies were carried out in the fourth section, as well as execution specifics. The conclusion is drawn in the last section.

2 Main results

Numerous full-rank presentations of dissimilar classes of generalized inverses were presented, either of arranged rank or of prescribed range and kernel. We observe the next full–rank presentation of outer inverses of predecided range and null space, proposed by Sheng and Chen [10].

Theorem 2.1. [10]
Consider a matrix $A\in\mathbb{C}^{m\times n}_r$ and subspaces $T\subseteq\mathbb{C}^n$ of normal rank $s\leq r$ and $S\subseteq\mathbb{C}^m$ of normal rank $m-s$. Assume that a matrix $X\in\mathbb{C}^{n\times m}$ fulfills $\mathcal{R}(X)=T,\mathcal{N}(X)=S$, and has a full-rank factorization $X=FG$. If A has an $A^{(2)}_{T,S}$ inverse, at that point the accompanying proclamations hold: (1) *GAF is an invertible matrix;* (2) $A^{(2)}_{T,S}=F(GAF)^{-1}G=A^{(2)}_{\mathcal{R}(F),\mathcal{N}(G)}$.

Theorem 2.2, deriving a representation of outer inverses, is proposed as a corollary of Theorem 2.1.

Theorem 2.2. *Observe a matrix $A\in\mathbb{C}(x)^{m\times n}_s$ of normal rank s, and the compact SVD decomposition of an random matrix $X\in\mathbb{C}(x)^{m\times n}_s$ of the rank $r\leq s$, where $U\in\mathbb{C}(x)^{m\times r}_r$, $\Sigma\in\mathbb{C}(x)^{r\times r}_r$ and $V\in\mathbb{C}(x)^{n\times r}_r$. Consider the set*

$$\mathbb{C}_r(X)=\{z|\ \mathrm{nr}(X)=rank(X(z))=rank(\Sigma(z)V^*(z)A(z)U(z))=r\}. \tag{5}$$

If the equality

$$\mathrm{nr}(X)=\mathrm{nr}(\Sigma VAU)=r \tag{6}$$

is valid, the next statement holds on $\mathbb{C}_r(X)$:

$$A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}=U(V^*AU)^{-1}V^*=A^{(2)}_{\mathcal{R}(X),\mathcal{N}(X)}. \tag{7}$$

Proof. From the equality

$$X=U\Sigma V^*, \tag{8}$$

representing a full–rank factorization of X on $\mathbb{C}_s(X)$, using the identity (2) proceeding from Theorem 2.1 and the expression

$$U(\Sigma V^*AU)^{-1}\Sigma V^*=A^{(2)}_{\mathcal{R}(U),\mathcal{N}(\Sigma V^*)}$$

the following can be obtained:

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

The statement $A^{(2)}_{\mathcal{R}(X),\mathcal{N}(X)} = A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ is valid on the subset $\mathbb{C}_r(X)$, according to equality (8). □

The accompanying theorem gives rules for developing the matrix X in request to produce several types of generalized inverses. Several types of generalized rational matrix inverses can be determined by choosing the appropriate X and generating the corresponding compact SVD factorization.

Corollary 2.1. *Consider $M \in \mathbb{C}(x)^{n\times n}_s$ of the normal rank s and a matrix $X \in \mathbb{C}(x)^{n\times n}_r$, where $r \le s$. The next statements are satisfied on the class $\mathbb{C}_r(X)$:*

$$M^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)} = \begin{cases} M^{\dagger}, & X = M^*; \\ M^{\#}, & X = M; \\ M^{D}, & X = M^k,\ k \ge ind(M); \\ M^{(-1)}_{(L)}, & \mathcal{R}(X) = L,\ \mathcal{N}(X) = L^{\perp}; \\ M^{(\dagger)}_{(L)}, & \mathcal{R}(X) = S,\ \mathcal{N}(X) = S^{\perp} \end{cases} \tag{9}$$

Proof. Following from Theorem 2.2 and expressions (4). □

Corollary 2.2. *Consider $M \in \mathbb{C}(x)^{m\times n}_s$ of the normal rank s and the compact SVD factorization of a matrix $X \in \mathbb{C}(x)^{m\times n}_r$ of the rank $r \le s$, where $X = U\Sigma V^*$ and $U \in \mathbb{C}(x)^{m\times r}_r$, $V \in \mathbb{C}(x)^{n\times r}_r$, and $\Sigma \in \mathbb{C}(x)^{r\times r}_r$ is the diagonal matrix. Let us introduce the set*

$$\mathbb{C}_r(X) = \{z|\ \mathrm{nr}(X) = rank(X(z)) = rank(V^*(z)M(z)U(z)) = r\}. \tag{10}$$

If the equality

$$\mathrm{nr}(X) = \mathrm{nr}(V^*MU) = r \tag{11}$$

holds, the next statement is satisfied on the set $\mathbb{C}_r(X)$:

$$M^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)} = U(V^*XU)^{-1}V^* = M^{(2)}_{\mathcal{R}(X),\mathcal{N}(X)}. \tag{12}$$

Proof. It proceeds from Corollary 2.1. □

We will consider the effective method for evaluating (12) as to solve the equations

$$V^*MUY = V^* \tag{13}$$

with respect to Y and then computing the matrix product UY:

$$M^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)} = UY. \tag{14}$$

Next we formulate the method for computing $M^{(2)}_{T,S}$ inverses of M as follows.

Algorithm 2.1 Computing $M^{(2)}_{T,S}$ from the compact SVD decomposition of M. **(Algorithm SVDATS2)**

Require: The matrix $M \in \mathbb{C}(x)^{m\times n}_s$ of the normal rank s.
1: Define matrix $X \in \mathbb{C}(x)^{m\times n}$ of the rank $r \leq s$.
2: Calculate the compact SVD decomposition of the matrix X.
3: Solve the equation (13) to determine the variable matrix Y.
4: **return** The generalized inverse matrix $M^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ as the matrix product UY from (14).

3 Computing $A^{(2)}_{T,S}$ inverses of polynomial matrices

Thereat, SVD-P algorithm from [6] is utilized to define the compact SVD decomposition of a polynomial matrix. In [7], the iterative method for evaluating the SVD of a polynomial matrix is delivered. It is formed on iterative calculations of QR factorization of the certain matrix and its change to a nearly diagonal polynomial matrix. Otherwise, the SVD may be calculated iteratively from the eigenvalue factorization and the second-order sequential best rotation (SBR2) method, as presented in [7].

Computing the QR factorization of a polynomial matrix is evidently an additionally complex problem compared to expressing the exact factorization of a scalar matrix, since every component of the matrix $A(z)$ now contains a series of polynomial factors. To change a single element of the matrix to zero, all the numbers of this element must be changed to zero, and this cannot be done by Givens rotations [6]. In its place, a analogous method is employed to that used when generating the Para unitary change matrix compulsory within the SBR2 algorithm and so the Para unitary polynomial matrix $Q(z)$ is formulated as a series of basic rotation matrices interspersed with delay matrices.

Another part of the PQRD-BC method contains an iterative procedure to force the constants related with all polynomial elements located under the diagonal of a specific column of the polynomial matrix $A(x)$ to be sufficiently small, which is done by using an array of EPGR processes. The algorithm operates as a series of steps, where at each step all constants related with the polynomial elements positioned below the diagonal of one column of the polynomial matrix $A(x)$ are made adequately small.

A method called SVD-P, by applying the polynomial QR decomposition (QRD-P) method is presented in [7]. The SVD-P works by iteratively using two QRD-Ps onto a polynomial matrix A in order to obtain a diagonal polynomial matrix Σ. Also, the choice maker is allowed to specify how small

the coefficients looking in off-diagonal elements must be driven for convergence. More meaningfully, the SVD-P using the QRD-P method is said to have generally smaller relative errors and the orders of the matrices in the result decomposition.
A paraunitary matrix, over $\mathbb{C}$, is a square matrix $U(z)$ such that $U(z)U^*(z^{-1}) = I$, where * denotes complex conjugate transpose.
Let $A(x) \in C^{p\times q}$ be a given polynomial matrix. The SVD-P by QRD-P algorithm returns as the result two polynomial paraunitary matrices $U(x) \in C^{p\times p}$ and $V(x) \in C^{q\times q}$ and a diagonal polynomial matrix $\Sigma(x)$ (see [7]) such that

$$U(x)A(x)V^*(x) \cong \Sigma(x). \tag{15}$$

Often, a precise diagonalization of the matrix $A(x)$ is not possible to gain, as each element of A is a polynomial. This is the reason for the approximate equality in (15). However, in [7] it is proved that a satisfactory approximation can generally be obtained. From (15) it is obvious that

$$A(x) \cong U^*(x^{-1})\Sigma(x)V(x^{-1}), \tag{16}$$

since U^* and V^* are also paraunitary matrices.

Theorem 3.1 is derived from Theorem 2.1 on polynomial matrices A and M.

Theorem 3.1. *Consider a polynomial matrix $A \in \mathbb{C}(x)^{m\times n}_s$ of normal rank s, an arbitrary Hermitian polynomial matrix $M \in \mathbb{C}(x)^{m\times n}_r$ of the normal rank $r \le s$ and let $M = U\Sigma V^*$ be the compact SVD decomposition of M, where $U \in \mathbb{C}(x)^{m\times r}_r$, $V \in \mathbb{C}(x)^{n\times r}_r$ and $\Sigma \in \mathbb{C}(x)^{r\times r}_r$ are of the form* (2)*. Denote the set $\mathbb{C}_r(M)$ as in* (10) *and an arbitrary (i,j)-th element of the inverse matrix $N = (V^*AU)^{-1}$ by*

$$n_{i,j}(x) = \sum_{k=0}^{\overline{n}_q} \overline{n}_{k,i,j}x^k \Big/ \sum_{k=0}^{\overline{\overline{n}}_q} \overline{\overline{n}}_{k,i,j}x^k. \tag{17}$$

If the condition (11) *is satisfied, then an arbitrary (i,j)-th element of $A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ can be calculated by*

$$\left(A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}\right)_{ij}(x) = \frac{\overline{\Theta}_{i,j}(x)}{\overline{\overline{\Theta}}_{i,j}(x)},$$

for $x \in \mathbb{C}_r(M)$, where $\overline{\Theta}_{i,j}(x)$ and $\overline{\overline{\Theta}}_{i,j}(x)$ are polynomials of the form

$$\overline{\Theta}_{i,j}(x) = \sum_{t=0}^{\overline{\overline{\Theta}}_q - \overline{\overline{\gamma}}_q + \overline{\gamma}_q} \left(\sum_{k=1}^{\min\{j,r\}} \sum_{l=1}^{\min\{i,r\}} \sum_{t_1=0}^{t} \overline{\gamma}_{t_1,i,j,k,l} \theta_{t-t_1,i,j,k,l} \right) x^t, \quad (18)$$

$$\overline{\overline{\Theta}}_{i,j}(x) = \text{PolynomialLCM} \left\{ \sum_{t=0}^{\overline{\overline{\gamma}}_q} \overline{\overline{\gamma}}_{t,i,j,k,l} x^t \Big| k = \overline{1,r},\ l = \overline{1,r} \right\} \quad (19)$$

$$= \sum_{t=0}^{\overline{\overline{\Theta}}_q} \overline{\overline{\theta}}_{t,i,j} x^t, \quad (20)$$

where for $k = \overline{1,r}$, $l = \overline{1,r}$, the coefficients $\theta_{t,i,j,k,l}$, $0 \le t \le \overline{\overline{\Theta}}_q - \overline{\overline{\Theta}}_q$ are the coefficients of the polynomial

$$\Theta_{i,j,k,l}(x) = \frac{\overline{\overline{\Theta}}_{i,j}(x)}{\sum_{t=0}^{\overline{\overline{\Theta}}_q} \overline{\overline{\Theta}}_{t,i,j,k,l} x^t}$$

and the following denotations are used:

$$\overline{\gamma}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{u}_{t_2,i,l} \overline{n}_{t_1-t_2-t_3,l,k} \overline{v}^*_{t_3,j,k}, \quad 0 \le t \le \overline{\gamma}_q = \overline{u}_q + \overline{n}_q + \overline{v}_q, \quad (21)$$

$$\overline{\overline{\gamma}}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{\overline{u}}_{t_2,i,l} \overline{\overline{n}}_{t_1-t_2-t_3,l,k} \overline{\overline{v}}^*_{t_3,j,k}, \quad 0 \le t \le \overline{\overline{\gamma}}_q = \overline{\overline{u}}_q + \overline{\overline{n}}_q + \overline{\overline{v}}_q. \quad (22)$$

Proof. Since the entries of the inverse matrix $N = (V^*AU)^{-1} = \{n_{i,j}(x)\}_{i,j=0}^{r}$ are determined by (17), it follows

$$(UN)_{ij}(x) = \sum_{l=1}^{r} u_{i,l}(x) n_{l,j}(x) = \sum_{l=1}^{r} \frac{\sum_{k=0}^{\overline{u}_q} \overline{u}_{k,i,l} x^k}{\sum_{k=0}^{\overline{\overline{u}}_q} \overline{\overline{u}}_{k,i,l} x^k} \frac{\sum_{k=0}^{\overline{n}_q} \overline{n}_{k,l,j} x^k}{\sum_{k=0}^{\overline{\overline{n}}_q} \overline{\overline{n}}_{k,l,j} x^k}$$

$$= \sum_{l=1}^{r} \frac{\sum_{k=0}^{\overline{u}_q + \overline{n}_q} \left(\sum_{k_1=0}^{k} \overline{u}_{k_1,i,l} \overline{n}_{k-k_1,l,j} \right) x^k}{\sum_{k=0}^{\overline{\overline{u}}_q + \overline{\overline{n}}_q} \left(\sum_{k_1=0}^{k} \overline{\overline{u}}_{k_1,i,l} \overline{\overline{n}}_{k-k_1,l,j} \right) x^k}.$$

Therefore, the following equalities are valid:

$$\begin{aligned}(U(V^*AU)^{-1}V^*)_{ij}(x) &= \sum_{k=1}^{r}(UN)_{ik}(x)\cdot(V^*)_{kj}(x)\\ &= \sum_{k=1}^{r}\sum_{l=1}^{r}\frac{\sum\limits_{t_1=0}^{\overline{u}_q+\overline{n}_q}\left(\sum\limits_{t_2=0}^{t_1}\overline{u}_{t_2,i,l}\overline{n}_{t_1-t_2,l,k}\right)x^{t_1}}{\sum\limits_{t_1=0}^{\overline{\overline{v}}_q+\overline{\overline{n}}_q}\left(\sum\limits_{t_2=0}^{t_1}\overline{\overline{u}}_{t_2,i,l}\overline{\overline{n}}_{t_1-t_2,l,k}\right)x^{t_1}}\frac{\sum\limits_{t_2=0}^{\overline{v}_q}\overline{v}^*_{t_2,j,k}x^{t_2}}{\sum\limits_{t_2=0}^{\overline{\overline{v}}_q}\overline{\overline{v}}^*_{t_2,j,k}x^{t_2}}\\ &= \sum_{k=1}^{r}\sum_{l=1}^{r}\frac{\sum\limits_{t_1=0}^{\overline{u}_q+\overline{n}_q+\overline{v}_q}\left(\sum\limits_{t_2=0}^{t_1}\sum\limits_{t_3=0}^{t_1-t_2}\overline{u}_{t_2,i,l}\overline{n}_{t_1-t_2-t_3,l,k}\overline{v}^*_{t_3,j,k}\right)x^{t_1}}{\sum\limits_{t_1=0}^{\overline{\overline{l}}_q+\overline{\overline{n}}_q+\overline{\overline{v}}_q}\left(\sum\limits_{t_2=0}^{t_1}\sum\limits_{t_3=0}^{t_1-t_2}\overline{\overline{u}}_{t_2,i,l}\overline{\overline{n}}_{t_1-t_2-t_3,l,k}\overline{\overline{v}}^*_{t_3,j,k}\right)x^{t_1}}.\end{aligned}$$

According to Theorem 2.2, the equation (12) is satisfied for $x \in \mathbb{C}_r(M)$, and an arbitrary (i,j)-th element of the inverse $A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ is presented in the form

$$\left(A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}\right)_{ij} = \sum_{k=1}^{r}\sum_{l=1}^{r}\frac{\sum\limits_{t=0}^{\overline{\gamma}_q}\overline{\gamma}_{t,i,j,k,l}x^t}{\sum\limits_{t=0}^{\overline{\overline{\gamma}}_q}\overline{\overline{\gamma}}_{t,i,j,k,l}x^t} = \frac{\overline{\Theta}_{i,j}(x)}{\overline{\overline{\Theta}}_{i,j}(x)},$$

where the numerator and denominator polynomials are evaluated as

$$\begin{aligned}\overline{\overline{\Theta}}_{i,j}(x) &= \text{PolynomialLCM}\left\{\sum_{t=0}^{\overline{\overline{\gamma}}_q}\overline{\overline{\gamma}}_{t,i,j,k,l}x^t\Big|k=\overline{1,r},\ l=\overline{1,r}\right\} = \sum_{t=0}^{\overline{\overline{\Theta}}_q}\overline{\overline{\theta}}_{t,i}x^t,\\ \overline{\Theta}_{i,j}(x) &= \sum_{k=1}^{j}\sum_{l=1}^{r}\left(\Theta_{i,j,k,l}(x)\sum_{t=0}^{\overline{\gamma}_q}\overline{\gamma}_{t,i,j,k,l}x^t\right),\end{aligned}$$

such that every polynomial $\Theta_{i,j,k,l}(x)$ is determined by

$$\Theta_{i,j,k,l}(x) = \overline{\overline{\Theta}}_{i,j}(x)/\sum_{t=0}^{\overline{\overline{\gamma}}_q}\overline{\overline{\gamma}}_{t,i,j,k,l}x^t = \sum_{t=0}^{\overline{\overline{\Theta}}_q-\overline{\overline{\gamma}}_q}\theta_{t,i,j,k,l}x^t.$$

Therefore, each numerator polynomial $\overline{\Theta}_{i,j}(x)$ is of the form (18), which completes the proof. □

Let us now propose the method for the estimate of generalized inverses of a polynomial matrix, agreeing to the prior theorem. It is based on the compact SVD decomposition of a polynomial matrix (where SVD-P algorithm from [6] is used), and calculating the inverse of a polynomial matrix (given by Algorithm 3.2 from [14]). In order to apply Algorithm 3.2 from [14] to the rational matrix V^*AU, it would be presented as a measure of a polynomial matrix and a polynomial.

The SVD-P by QRD-P algorithm applied here is an iterative technique where in every step the paraunitary matrices are computed using the QRD-P by columns strategy (see [7] for details), as it generally uses less steps for the convergence and it produces the upper triangular matrix of smallest order. These two properties of the stated Algorithm 3.2 are rather related.

Algorithm 3.2 Evaluating the $A^{(2)}_{T,S}$ inverse of a polynomial matrix using the compact SVD factorization of an arbitrary polynomial matrix M.**(Algorithm SVDATS2)**

Require: Polynomial matrix $A(x) \in \mathbb{C}(x)^{m\times n}_s$ of the normal rank s, the convergence parameter ϵ and the truncation parameter μ.

1: Choose an arbitrary fixed $m \times n$ polynomial matrix M of thr normal rank $r \leq s$.

2: Compute the SVD decomposition (15) of the matrix M using the SVD-P by QRD-P algorithm for the given convergence and truncation parameters.

3: Evaluate the compact SVD decomposition of M by taking the first r column vectors of the first matrix appearing in (15) and r row vectors of the last matrix from (15), in order to obtain the form (1)–(2).

4: Denote the entries of the polynomial matrices U and V appearing in (2) as,

$$u_{i,j}(x) = \sum_{k=\overline{u}_{min}}^{\overline{u}_{max}} u_{k,i,j}x^k, \qquad i,j=\overline{1,n},$$

$$v_{i,j}(x) = \sum_{k=\overline{v}_{min}}^{\overline{v}_{max}} v_{k,i,j}x^k, \qquad i,j=\overline{1,n},$$

5: Extract the denominator polynomial from the rational matrix V^*AU, such that the equality $V^*AU = \frac{1}{p(x)}P(x)$ is valid, where $p(x)$ is a polynomial and $P(x)$ is a polynomial matrix.

6: Compute the inverse matrix $P^{-1}(x)$ using the Algorithm 3.2 from [14]. Evaluate the inverse matrix $N = (V^*AU)^{-1}$ as the product $p(x)\cdot P^{-1}(x)$, where the entries are of the form (17).

7: Make the notations $\overline{\gamma}_q = \overline{u}_q + \overline{n}_q + \overline{v}_q$, $\overline{\overline{\gamma}}_q = \overline{\overline{u}}_q + \overline{\overline{n}}_q + \overline{\overline{v}}_q$, and for $i = \overline{1,m}, j = \overline{1,n}$ perform Step 5.1 – Step 5.5.

5.1: For $k=\overline{1,r},\ l=\overline{1,r}$ perform the following computations:

$$\begin{aligned}\overline{\gamma}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1}\sum_{t_3=0}^{t_1-t_2}\overline{u}_{t_2,i,l}\overline{n}_{t_1-t_2-t_3,l,k}\overline{v}^*_{t_3,j,k}, \quad 0\le t\le \overline{\gamma}_q,\\ \overline{\overline{\gamma}}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1}\sum_{t_3=0}^{t_1-t_2}\overline{\overline{u}}_{t_2,i,l}\overline{\overline{n}}_{t_1-t_2-t_3,l,k}\overline{\overline{v}}^*_{t_3,j,k}, \quad 0\le t\le \overline{\overline{\gamma}}_q.\end{aligned}$$

5.2: Compute the denominator polynomial of the (i,j)-th element of $A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ by

$$\text{PolynomialLCM}\left\{\sum_{t=0}^{\overline{\overline{\gamma}}_q}\overline{\overline{\gamma}}_{t,i,j,k,l}x^t\Big|k=\overline{1,r},\ l=\overline{1,r}\right\},$$

and denote it by $\overline{\overline{\Theta}}_{i,j}(x)=\sum_{t=0}^{\overline{\overline{\Theta}}_q}\overline{\overline{\theta}}_{t,i,j}x^t$.

5.3: For $k,l=\overline{1,r}$ compute the polynomial $\overline{\overline{\Theta}}_{i,j}(x)/\sum\limits_{t=0}^{\overline{\overline{\gamma}}_q}\overline{\overline{\gamma}}_{t,i,j,k,l}x^t$ and denote it by $\Theta_{i,j,k,l}(x)=\sum\limits_{t=0}^{\overline{\overline{\Theta}}_q-\overline{\overline{\gamma}}_q}\theta_{t,i,j,k,l}x^t$.

5.4: Compute the numerator polynomial of the (i,j)-th element of $A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ as

$$\overline{\Theta}_{i,j}(x)=\sum_{t=0}^{\overline{\overline{\Theta}}_q-\overline{\overline{\gamma}}_q+\overline{\gamma}_q}\left(\sum_{k=1}^{r}\sum_{l=1}^{r}\sum_{t_1=0}^{t}\overline{\gamma}_{t_1,i,j,k,l}\theta_{t-t_1,i,j,k,l}\right)x^t,$$

5.5: **Return** the (i,j)-th entry of the matrix $A^{(2)}_{\mathcal{R}(U),\mathcal{N}(V^*)}$ as $\overline{\Theta}_{i,j}(x)/\overline{\overline{\Theta}}_{i,j}(x)$.

4 Illustrative examples

Through examples we study our method and eventually check various developments to make comparative processing timings using random test matrices.

Example 4.1. Consider the next randomly generated polynomial matrix

$$A(x)=\begin{bmatrix} 0.89-\frac{0.06}{x} & 0.16x+0.47 & \frac{0.33}{x}-0.04\\ -0.33-\frac{0.06}{x} & 0.18x+0.47 & \frac{0.34}{x}-0.33\\ -0.44-\frac{0.08}{x} & 0.64-0.23x & 0.24+\frac{0.45}{x}\end{bmatrix}.$$

These rational matrices used in SVD are gained:

$$U(x) = \begin{bmatrix} -0.867 & -\frac{0.441}{x} & 0.006 \\ 0.257 & -\frac{0.444}{x} & -0.783 \\ 0.467 & -\frac{0.602}{x} & 0.51 \end{bmatrix}, \Sigma(x) = \begin{bmatrix} -1.067 & 0 & 0 \\ 0 & -1.338 & 0 \\ 0 & 0 & 0.544 \end{bmatrix},$$

$$V(x) = \begin{bmatrix} 0.971 & 0.171x & -0.048 \\ -0.105 & 0.802x & 0.565 \\ 0.149 & -0.545x & 0.806 \end{bmatrix}.$$

Therefore:

$$A^{\dagger} = A^{\#} =$$

$$\begin{bmatrix} \frac{-1+x}{-85-83x-2x^2+x^3} & \frac{9+17x}{85+83x+2x^2-x^3} & \frac{3+10x}{85+83x+2x^2-x^3} & \frac{26(1+x)}{-85-83x-2x^2+x^3} \\ \frac{9+17x}{85+83x+2x^2-x^3} & \frac{4+36x-2x^2+3x^3}{-85-83x-2x^2+x^3} & \frac{27-2x+2x^2-2x^3}{85+83x+2x^2-x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} \\ \frac{3+10x}{85+83x+2x^2-x^3} & \frac{27-2x+2x^2-2x^3}{85+83x+2x^2-x^3} & \frac{9-16x+x^2-x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} \\ \frac{26(1+x)}{-85-83x-2x^2+x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} & \frac{4-4x+8x^3}{-85-83x-2x^2+x^3} \end{bmatrix}.$$

Consequently, the proposed algorithmic complexity will be nearly $O(n^3m^2)$, wherever $O(n^3)$ is the complexity of the SVD and m is the largest exponent of the polynomials resulting within this method.

Realization can be done in the package MATHEMATICA which is appropriate for symbolic computations and has built-functions for manipulations with the expressions [19]. The function `Simplify[ ]` is used to make the necessary simplifications.

Example 4.2. Let us analyze the effectiveness of the SVDATS Algorithm. For that purpose, a gathering of various calculations for the assessment of Moore-Penrose inverse is thought about. The accompanying table presents mean times consumed by performing these calculations on three arrangements examples from [20], considering the halfway instance of $a = 1$.

Test matrix	A_{10}	A_{50}	A_{100}	A_{150}	S_{10}	S_{50}	S_{100}	S_{150}	F_{10}	F_{50}	F_{100}	F_{150}
Partitioning [12]	0.01	0.48	2.75	14.55	0.03	1.18	9.20	73.55	0.01	0.45	2.81	15.3
Lev.-Faddeev [8]	0.04	2.76	43.78	-	0.11	2.51	44.37	-	0.13	2.67	42.84	-
PseudoInverse [19]	0.03	1.86	25.51	-	0.13	2.77	41.39	-	0.09	2.33	33.54	-
Courrieu [5]	0.02	0.76	5.84	39.86	0.01	0.37	2.29	14.66	0.01	0.70	5.78	35.43
LDLGInverse [11]	0.02	1.87	11.67	-	0.02	0.98	4.55	22.31	0.01	1.96	12.64	-
SVDATS	0.02	0.82	5.44	40.33	0.01	0.39	2.11	15.99	0.01	0.87	6.02	38.01

Table 1. Mean processor timings (in sec.) from several methods and SVDATS

The proposed SVDATS algorithm is the fastest for test matrices A_{10}, A_{100}, S_{10}, S_{100}, F_{10}, and nearly equal to the running times of other tested algorithms for other classes of matrices, making it a good candidate for solving the given problem in terms of the running time.

Rank deficient matrices are treated faster than full-rank matrices of the same size, what is the straight result of the smaller matrix sizes of L and D, created by the full-rank factorization. However, computation timings may expand greatly with the matrix size and density increasing.

5 Conclusion

In this paper, we have observed the symbolic computation of outer inverses on rational and polynomial matrices using compact SVD, and a new technique is developed based on compact singular value factorization. We have proven that our algorithm is highly efficient in terms of running times. Future research may include developing analogous algorithms to evaluate $A^2_{T,S}$ generalized inverses.

References

[1] A. G. Akritas, G. I. Malaschonok, *Computations in Modules over Commutative Domains*, Lecture Notes in Computer Science, **4770**, (2007), 11–23.

[2] A. Ben-Israel, T.N.E. Greville, *Generalized inverses: Theory and Applications*, Second Ed., Springer, 2003.

[3] Y. Chen, *The generalized Bott–Duffin inverse and its application*, Linear Algebra Appl. **134** (1990), 71–91.

[4] P. Courrieu, *Fast solving of weighted pairing least-squares systems*, Journal of Computational and Applied Mathematics **231** (2009), 39–48.

[5] P. Courrieu, *Fast Computation of Moore-Penrose Inverse Matrices*, Neural Information Processing - Letters and Reviews **8** (2005), 25–29.

[6] J. Foster, J. McWhirter, M. Davies, J. Chambers, An algorithm for calculating the QR and singular value decompositions of polynomial matrices. IEEE Transactions on Signal Processing, 58(3), (2009) 1263–1274.

[7] J. Foster, *Algorithms and techniques for polynomial matrix decompositions*, Ph.D. dissertation, School Eng., Cardiff Univ., Cardiff, U.K., 2008.

[8] J. Jones, N.P. Karampetakis and A.C. Pugh, *The computation and application of the generalized inverse via Maple*, J. Symbolic Comput. **25** (1998), 99–124.

[9] A.N. Malyshev, *"Matrix equations: Factorization of matrix polynomials" M. Hazewinkel (ed.)*, Handbook of Algebra I, Elsevier (1995), 79–116.

[10] X. Sheng, G. Chen, *Full-rank representation of generalized inverse $A_{T,S}^{(2)}$ and its applications*, Comput. Math. Appl. **54** (2007), 1422–1430.

[11] I.P. Stanimirović, M.B. Tasić, *Computation of generalized inverses by using the LDL* decomposition*, Appl. Math. Lett., **25** (2012), 526–531.

[12] P.S. Stanimirović and M.B. Tasić, *Partitioning method for rational and polynomial matrices*, Appl. Math. Comput. **155** (2004), 137–163.

[13] M. Tasić, I. Stanimirović, *Symbolic computation of Moore-Penrose inverse using the LDL* decomposition of the polynomial matrix*, Filomat **27:8** (2013), 1393–1403.

[14] M.B. Tasić, P.S. Stanimirović, M.D. Petković, *Symbolic computation of weighted Moore-Penrose inverse using partitioning method*, Appl. Math. Comput. **189** (2007), 615–640.

[15] M.B. Tasić, P.S. Stanimirović, *Symbolic and recursive computation of different types of generalized inverses*, Appl. Math. Comput. **199** (2008), 349–367.

[16] M.B. Tasić, P.S. Stanimirović, *Differentiation of generalized inverses for rational and polynomial matrices*, Appl. Math. Comput. **216** (2010), 2092–2106.

[17] G. Wang, Y. Wei, S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, Beijing/New York, 2004.

[18] Y. Wei, C. Deng, *A note on additive results for the Drazin inverse*, Linear and Multilinear Algebra **59:12** (2011), 1319–1329.

[19] S. Wolfram, *The* MATHEMATICA *Book, 5th ed.*, Wolfram Media/Cambridge University Press, Champaign, IL 61820, USA, 2003.

[20] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36** (1986), 105–162.

Ivan Stanimirović,
Department of Computer Science,
Faculty of Sciences and Mathematics, University of Niš,
Višegradska 33, 18000 Niš, Serbia.
Email: ivan.stanimirovic@pmf.edu.rs