Linear Algebra and its Applications 437 (2012) 1317–1331

Contents lists available at SciVerse ScienceDirect

Linear Algebra and its Applications

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

Symbolic computation of $A^{(2)}_{T,S}$-inverses using *QDR* factorization

Predrag S. Stanimirović [a,*,1], Dimitrios Pappas [b], Vasilios N. Katsikis [c], Ivan P. Stanimirović [a,1]

[a] *University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*
[b] *Department of Statistics, Athens University of Economics and Business, 76 Patission Str., 10434 Athens, Greece*
[c] *Technological Education Institute of Piraeus, Petrou Ralli & Thivon 250, 12244 Aigaleo, Athens, Greece*

ARTICLE INFO

Article history:
Received 30 November 2011
Accepted 14 April 2012
Available online 14 May 2012

Submitted by R.A. Brualdi

AMS classification:
15A09
15A23
68W30

Keywords:
Generalized inverse $A^{(2)}_{T,S}$
QDR factorization
Full-rank representation
Rational matrices
MATHEMATICA

ABSTRACT

Efficient evaluation of the full-rank *QDR* decomposition is established. A method and algorithm for efficient symbolic computation of $A^{(2)}_{T,S}$ inverses of a given rational matrix A is defined using the full-rank *QDR* decomposition of an appropriate rational matrix W. The algorithm is implemented using MATHEMATICA's ability to deal with symbolic expressions as well as with numbers. Examples including polynomial and rational matrices are presented. Some comparisons with well-known methods for symbolic evaluation of generalized inverses are obtained.

1. Introduction

As usual, by $\mathbb{C}(x)$ we denote the set of rational functions with complex coefficients. The set of $m \times n$ matrices with elements in $\mathbb{C}(x)$ is denoted by $\mathbb{C}(x)^{m\times n}$. By I we denote the identity matrix of an appropriate order. Following the standard notations, A^*, $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the conjugate transpose, the range, and the null space of $A \in \mathbb{C}(x)^{m\times n}$. In addition, $\mathrm{nrank}(A)$ denotes the normal

[*] Corresponding author.
E-mail addresses: pecko@pmf.ni.ac.rs (P.S. Stanimirović), dpappas@aueb.gr (D. Pappas), vaskats@gmail.com (V.N. Katsikis), ivan.stanimirovic@gmail.com (I.P. Stanimirović).

[1] The first and fourth authors gratefully acknowledge support from the Research Project 174013 of the Serbian Ministry of Science.

0024-3795/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.laa.2012.04.026

rank of A (rank over the set $\mathbb{C}(x)$) and $\mathbb{C}(x)_r^{m\times n}$ denotes the set of matrices from $\mathbb{C}(x)^{m\times n}$ with the normal rank r. Similarly, $\text{rank}(A)$ (resp. $\text{ind}(A)$) denotes the rank (respectively index) of a constant matrix A. The subset of constant complex matrices $\mathbb{C}^{m\times n}$ with rank r is denoted by $\mathbb{C}_r^{m\times n}$.

The fundamental result that defines conditions for the existence of outer inverses with prescribed range and null space of $A \in \mathbb{C}_r^{m\times n}$ is frequently used in the literature. We restate this result from [1]. If T is a subspace of $\mathbb{C}^n$ of dimension $\dim(T) = t \leqslant r$ and S is a subspace of $\mathbb{C}^m$ of dimension $\dim(S) = m - t$, then A has a $\{2\}$-inverse X satisfying $\mathcal{R}(X) = T$ and $\mathcal{N}(X) = S$ if and only if $AT \oplus S = \mathbb{C}^m$. The matrix X satisfying $XAX = X, \mathcal{R}(X) = T, \mathcal{N}(X) = S$ is unique and it is denoted by $A^{(2)}_{T,S}$.

The most important generalized inverses are particular appearances of outer inverses with prescribed range and null space and correspond to appropriate choices of matrices T and S. The Moore–Penrose $A^\dagger$ and the weighted Moore–Penrose inverse $A^\dagger_{M,N}$ are equal to (see, for example [1,21]):

$$A^\dagger = A^{(2)}_{\mathcal{R}(A^*),\mathcal{N}(A^*)}, \quad A^\dagger_{M,N} = A^{(2)}_{\mathcal{R}(A^\sharp),\mathcal{N}(A^\sharp)}, \tag{1.1}$$

where $A^\sharp = N^{-1}A^*M$ and N and M are positive definite matrices of the orders $n \times n$ and $m \times m$, respectively. The Drazin inverse A^D and the group inverse $A^\#$ of a given square matrix A are equal to (see [1,21]):

$$A^D = A^{(2)}_{\mathcal{R}(A^l),\mathcal{N}(A^l)},\ l \geqslant \text{ind}(A), \quad A^\# = A^{(2)}_{\mathcal{R}(A),\mathcal{N}(A)}. \tag{1.2}$$

If A is a L-positive semi-definite matrix and L is a subspace of $\mathbb{C}^n$ which satisfies $AL \oplus L^\perp = \mathbb{C}^n$, $S = \mathcal{R}(P_L A)$, then the Bott–Duffin inverse $A^{(-1)}_{(L)}$ and the generalized Bott–Duffin inverse $A^{(\dagger)}_{(L)}$ are defined by [2,21]:

$$A^{(-1)}_{(L)} = A^{(2)}_{L,L^\perp}, \quad A^{(\dagger)}_{(L)} = A^{(2)}_{S,S^\perp}. \tag{1.3}$$

Our basic motivation is the representation of the Moore–Penrose inverse $A^\dagger$ from [9]. The computational method introduced in [9] is derived from the QR decomposition of the matrix A. In the present paper we develop two extensions of this algorithm. First generalization consists in the fact that our algorithm is applicable for calculating an extensive class of $A^{(2)}_{T,S}$ inverses, not only for computing the Moore–Penrose inverse. Furthermore, an extension of this algorithm to the set of one-variable polynomial and rational matrices is presented. Instead of the QR decomposition of A we use the QDR decomposition of an appropriately chosen matrix W and thus derived full-rank factorization. The choice of QDR decomposition is critical in order to eliminate appearances of square roots in entries of the QR decomposition.

A number of different approaches for the generalized inversion of polynomial and rational matrices have been proposed. So far, the Leverrier–Faddeev algorithm, the Greville's partitioning method and the LDL^* decomposition are used in the symbolic implementation of generalized inverses. Methods for computing the Moore–Penrose inverse of polynomial matrices based on the Leverrier–Faddeev algorithm are investigated in [4,6–8].

Various extensions of the Greville's recursive algorithm from [5] which are applicable to rational and polynomial matrices have been established. The first result in this approach is the extension of the Greville's algorithm to the set of one-variable polynomial and/or rational matrices, introduced in [15]. The extension of results from [15] to the set of the two-variable rational and polynomial matrices is introduced in [11]. Wang's partitioning method from [20], aimed in the computation of the weighted Moore–Penrose inverse, is extended to the set of one-variable rational and polynomial matrices in the paper [18]. Also, the efficient algorithm for computing the weighted Moore–Penrose, appropriate for sparse polynomial matrices where only a few polynomial coefficients are nonzero, is established in [10]. In the paper [19] the Greville's recursive principle is generalized to $\{1\}$, $\{1, 3\}$, $\{1, 4\}$-inverses and the Moore–Penrose inverse and extended to the set of the one-variable rational and polynomial matrices.

The algorithm for computing $\{1, 2, 3\}$, $\{1, 2, 4\}$ inverses and the Moore–Penrose inverse of a given rational matrix, based on the LDL^* factorization, is developed in [13]. Extension of that algorithm to the set of polynomial matrices whose elements are polynomials with real coefficients is presented in the same paper.

Computations which include square roots entries are inappropriate for symbolic and algebraic computations. Symbolic implementation of expressions $\sqrt{\sum_{i=0}^{q} A_i s^i}$ which include constant matrices A_i, $i = 0, \dots, q$ and the unknown s is a very complicated problem in procedural programming languages and a job whose execution requires a lot of processor time in packages for symbolic computation. In addition, the square root of some matrix polynomials often occurs when generating the *QR* factorization. Generating expressions that include square roots can be avoided by using the *QDR* decomposition. This possibility is of essential importance in symbolic polynomial computation. Similarly as in [13], our motivation in the present paper is to exploit the advantage of a square-root-free decomposition in symbolic calculations and extend the Algorithm *qrginv* from [9] to the set of polynomial matrices. What is the main reason to replace the LDL^* with the *QR* decomposition? The main disadvantage of the LDL^* decomposition is that it is applicable only to symmetric positive definite matrices. This drawback fix a limit to results from [13] to $\{1, 2, 3\}$, $\{1, 2, 4\}$-inverses and the Moore–Penrose inverse. Representations proposed in the present article are applicable to a more wider set of outer inverses with prescribed range and null space. For this purpose, instead of the LDL^* decomposition (used in [13]), we use the *QDR* factorization of a rational matrix in order to avoid entries containing square roots. Evidently, this form is appropriate for the manipulation with polynomial entries. Therefore, the proposed algorithm is highly suitable for the implementation in procedural programming languages, because of square root-free entries and the basic simplification method only requiring the evaluation of *gcd* of two polynomials.

The paper is organized as follows. Section 2 is divided in two subsections; in the first the algorithm for symbolic computation of $A_{T,S}^{(2)}$ inverses of one-variable polynomial or rational matrix A is introduced, whereas in the second the implementation details of the proposed algorithmic procedure are presented. Note that, the representation of outer inverses with prescribed range and null space is derived on the *QDR* matrix decomposition of an appropriately chosen matrix W. The algorithm is implemented in the symbolic computational language MATHEMATICA. In Section 3, several illustrative examples as well as the comparison with other known algorithms for symbolic computation of outer inverses are presented. The conclusions of our work are discussed in Section 4.

2. Symbolic computation of $A_{T,S}^{(2)}$ using *QDR* factorization

The basic *QDR* factorization of a matrix A generates three matrices: the matrix Q with rank equal to the rank of A, the diagonal matrix D and the matrix R, in stages. Here we propose an algorithm for the direct computation of the full-rank *QDR* decomposition, where the matrix Q is formed without zero columns, R is generated without zero rows and the diagonal matrix D is without both zero rows and zero columns. The *QDR* decomposition produces one more diagonal matrix with respect to the *QR* decomposition, but returns matrices with square root free entries, preferable for the symbolic computation.

Algorithm 2.1 Full-rank *QDR* decomposition of a rational matrix A

```
Require: Matrix $A \in \mathbb{C}(x)_s^{n\times m}$.
1: Construct the three zero matrices: $Q \in \mathbb{C}(x)^{n\times s}$, $D \in \mathbb{C}(x)^{s\times s}$, $R \in \mathbb{C}(x)^{s\times m}$.
2: For $i = \overline{1, s}$ repeat
    2.1: Set the matrix $B$ be equal to $A - QDR$.
    2.2: Determine the first next nonzero column of the matrix $B$ and denote it as $c$.
    2.3: Set the $i$th column of $Q$ be equal to $c$.
    2.4: For $j = \overline{i, m}$ set $R_{ij}$ be equal to the inner product of the vector $c$ with the $j$th column of $B$.
    2.5: Set the element $D_{ii}$ to the reciprocal of the squared 2-norm of the column $c$.
```

Notice that the equation of the form $A = QDR + B$ is stated at each step, where we start with $B = A$. At the end of Algorithm 2.1 we have $B = 0$ and $A = QDR$. Let us mention that the matrix R is upper triangular, and the columns of Q contain an orthogonal basis of the column space of A. The implementation of this algorithm in MATHEMATICA is given in Appendix.

In many cases, the Gram–Schmidt algorithm with column pivoting is required. At each stage the column of B with the largest 2-norm is picked, instead of the first nonzero column. Then the matrix R is column permuted upper triangular matrix, and the columns of Q again contain an orthogonal basis for the column space of A.

Example 2.1. Consider the next two matrices:

$$F = \begin{bmatrix} 3 & -2 \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Choose the matrix W that is equal to their product

$$W = FG = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The QDR decomposition of the matrix W is determined as

$$\begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & \frac{1}{10} \\ -1 & \frac{3}{10} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 10 & -7 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{10} & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Example 2.2. Consider the following polynomial matrix:

$$W = \begin{bmatrix} -3 - 4x^2 & 2 - 7x & 4 \\ -9x & -3 + 3x^2 & -5 \\ -2x + 9x^2 & 9x^2 & -5 \end{bmatrix}.$$

Applying Algorithm 2.1 we get the following matrices from the QDR decomposition of W:

$$Q = \begin{bmatrix} -3 - 4x^2 & \frac{x(81+170x-694x^2+657x^3-747x^4+324x^5)}{9+109x^2-36x^3+97x^4} & \frac{3x(90-217x+53x^2+3742x^3-1388x^4-3444x^5+5565x^6+2052x^7)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \\ -9x & -3 + 3x^2 + \frac{9x(-6+48x-8x^2-17x^3+81x^4)}{9+109x^2-36x^3+97x^4} & -\frac{x(-180+2279x+3163x^2-10909x^3+8706x^4+10329x^5-14904x^6+8208x^7)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \\ x(-2 + 9x) & \frac{x(-12+231x-448x^2+1019x^3-9x^4+144x^5)}{9+109x^2-36x^3+97x^4} & \frac{3(-135-552x+675x^2+2603x^3-2674x^4-1292x^5+4108x^6-60x^7+912x^8)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \end{bmatrix},$$

$$D=\begin{bmatrix} \frac{1}{9+109x^2-36x^3+97x^4} & 0 & 0 \\ 0 & \frac{9+109x^2-36x^3+97x^4}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} & 0 \\ 0 & 0 & \frac{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8}{\left(45+94x-113x^2-15x^3+228x^4\right)^2} \end{bmatrix},$$

$$R=\begin{bmatrix} 9+109x^2-36x^3+97x^4 & -6+48x-8x^2-17x^3+81x^4 & -12+55x-61x^2 \\ 0 & \frac{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8}{9+109x^2-36x^3+97x^4} & -\frac{-135-654x+1135x^2+716x^3+1882x^4+6048x^5+879x^6}{9+109x^2-36x^3+97x^4} \\ 0 & 0 & \frac{\left(45+94x-113x^2-15x^3+228x^4\right)^2}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \end{bmatrix}.$$

Many representations for various generalized inverses of prescribed rank as well as for the generalized inverses with prescribed range and kernel are known in the literature. The most useful representation for the research in the current paper is the following full-rank representation of outer inverses with prescribed range and null space from [12].

Proposition 2.1 [12]. *Let $A \in \mathbb{C}_r^{m\times n}$, T be a subspace of $\mathbb{C}^n$ of dimension $s \leqslant r$ and let S be a subspace of $\mathbb{C}^m$ of dimension $m-s$. In addition, suppose that $W \in \mathbb{C}^{n\times m}$ satisfies $\mathcal{R}(W)=T$, $\mathcal{N}(W)=S$. Let W has an arbitrary full-rank decomposition, that is $W=FG$. If A has a $\{2\}$-inverse $A_{T,S}^{(2)}$, then:*

(1) *GAF is an invertible matrix;*
(2) $A_{T,S}^{(2)} = F(GAF)^{-1}G = A_{\mathcal{R}(F),\mathcal{N}(G)}^{(2)}$.

In the particular case, a full-rank representation of the Drazin inverse A^D based on an arbitrary full-rank decomposition of A^l, $l \geqslant \mathrm{ind}(A)$, is introduced in [17].

An alternative explicit expression for the generalized inverse $A_{T,S}^{(2)}$, which is based on the usage of the group inverse, is given in [22]. The characterization, the representation theorem and the limiting expression for $A_{T,S}^{(2)}$ are derived in [22] using this representation.

The authors of the paper [3] established a basic representation and a general representation theorem for the outer inverse $A_{T,S}^{(2)}$. Based on this representation, several specific representations and iterative methods for computing $A_{T,S}^{(2)}$ are presented in [3].

The next statement represents a full-rank representation for outer inverses with prescribed range, null space and rank, of the same general form as in Proposition 2.1. The statement is valid for rational matrices and it is based on the full-rank factorization of W arising from the QDR decomposition defined in Algorithm 2.1.

Lemma 2.1. *Let $A \in \mathbb{C}(x)_r^{m\times n}$ be given. For an arbitrary matrix $W \in \mathbb{C}(x)_s^{n\times m}$, $s \leqslant r$, consider its QDR decomposition produced by Algorithm 2.1, of the form*

$$W = QDR,$$

where $Q \in \mathbb{C}(x)_s^{n\times s}$, $D \in \mathbb{C}(x)_s^{s\times s}$ *is a diagonal matrix and* $R \in \mathbb{C}(x)_s^{s\times m}$ *is an upper triangular matrix. Let us assume that the condition*

$$\mathrm{nrank}(W) = \mathrm{nrank}(RAQ) = s \tag{2.1}$$

is satisfied. Define the set

$$\mathbb{C}_s(W) = \{x_c \in \mathbb{C} |\ \mathrm{nrank}(W) = \mathrm{rank}(W(x_c)) = \mathrm{rank}(R(x_c)A(x_c)Q(x_c)) = s\}. \tag{2.2}$$

Then the following statement is valid on the set $\mathbb{C}_s(W)$*:*

$$\begin{aligned} A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} &= Q(RAQ)^{-1}R \\ &= A^{(2)}_{\mathcal{R}(W),\mathcal{N}(W)}. \end{aligned} \tag{2.3}$$

Proof. Obviously, the factorization

$$W = QDR = (QD)(R), \tag{2.4}$$

represents a full-rank factorization of W on the set $\mathbb{C}_s(W)$. Since D and RAQ are invertible, they satisfy the reverse order law property $(RAQD)^{-1} = D^{-1}(RAQ)^{-1}$. Now, the first identity in (2.3) follows from Proposition 2.1 and

$$\begin{aligned} QD(RAQD)^{-1}R = Q(RAQ)^{-1}R \\ = A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}. \end{aligned}$$

The identity

$$A^{(2)}_{\mathcal{R}(W),\mathcal{N}(W)} = A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$$

is evidently satisfied in $\mathbb{C}_s(W)$ from (2.4) and invertibility of D. □

Remark 2.1. Notice that for a given matrix $A \in \mathbb{C}(x)_r^{m\times n}$ arbitrarily chosen matrix $W \in \mathbb{C}(x)_s^{n\times m}$, $s \leqslant r$, produces corresponding outer inverse with prescribed range and null space of the form (2.3), where (2.4) is the *QDR* decomposition of W. The outer inverse $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ represents a function on the set $\mathbb{C}(x)$. Elements of the outer inverse, denoted by g_{ij}, are also functions on $\mathbb{C}(x)$. Then the domain of $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ is $C_s(W) \bigcap_{i,j} \mathrm{Dom}(g_{ij})$, where $\mathrm{Dom}(g_{ij})$ denotes the domain of g_{ij}.

Taking into account representations (1.1)–(1.3) for main outer inverses, we get the following representations.

Corollary 2.1. *For a given matrix* $A \in \mathbb{C}(x)_r^{m\times n}$ *and arbitrarily chosen matrix* $W \in \mathbb{C}(x)_s^{n\times m}$ *with the QDR decomposition defined in* (2.4) *the following statements are valid in* $\mathbb{C}_s(W) \bigcap_{i,j} \mathrm{Dom}(g_{ij})$:

$$\begin{aligned}
&\text{(a)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{\dagger} \text{ in the case of } W = A^*; \\
&\text{(b)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{\dagger}_{M,N} \text{ in the case of } W = A^{\sharp}; \\
&\text{(c)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{\#} \text{ in the case of } W = A; \\
&\text{(d)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{D} \text{ in the case of } W = A^{l},\ l \geqslant \mathrm{ind}(A); \\
&\text{(e)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{(-1)}_{(L)} \text{ in the case of } \mathcal{R}(W) = L,\ \mathcal{N}(W) = L^{\perp}; \\
&\text{(f)}\ \ A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{(\dagger)}_{(L)} \text{ in the case of } \mathcal{R}(W) = S,\ \mathcal{N}(W) = S^{\perp}.
\end{aligned} \tag{2.5}$$

The lack of numerical calculations that are based on the representation (2.3) is calculation of the inverse matrix. Numerically more stable approach for computing (2.3) is to solve the set of equations

$$RAQX = R \tag{2.6}$$

and then compute the matrix product

$$A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = QX. \tag{2.7}$$

Now we can propose the following Algorithm 2.2 for the evaluation of $A^{(2)}_{T,S}$ inverses of a given rational matrix.

Algorithm 2.2 Computing the $A^{(2)}_{T,S}$ inverse of a rational matrix A using the *QDR* decomposition of appropriately chosen rational matrix W.
(Algorithm QDRATS)

```
Require: The matrix $A \in \mathbb{C}(x)^{m\times n}_r$.
1: Choose an arbitrary matrix $W \in \mathbb{C}(x)^{n\times m}$ of normal rank $s \leqslant r$.
2: Generate the full-rank QDR decomposition of the matrix $W$ by applying Algorithm 2.1.
3: Solve the matrix equation (2.6) with respect to unknown matrix $X$.
4: Compute the output $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = QX$.
```

2.1. Symbolic computation of $A^{(2)}_{T,S}$ inverse of a rational matrix

The implementation of Algorithm *QDRATS* is mainly based on the symbolic data processing possibilities incorporated in the package MATHEMATICA. In this subsection we want to accelerate this implementation and to adapt it to procedural programming languages. For this purpose, we consider arbitrary rational matrix $A(x) \in \mathbb{C}(x)^{m\times n}$ in the general form $A(x) = \frac{\overline{A}(x)}{\overline{\overline{A}}(x)}$, where the matrices $\overline{A}(x)$ and $\overline{\overline{A}}(x)$ are given in the polynomial form with respect to the unknown x:

$$\overline{A}(x) = \sum_{i=0}^{\overline{a}_q} \overline{A}_i x^i, \quad \overline{\overline{A}}(x) = \sum_{i=0}^{\overline{\overline{a}}_q} \overline{\overline{A}}_i x^i, \tag{2.8}$$

where $\overline{A}_i, i = 0, \ldots, \overline{a}_q$ and $\overline{\overline{A}}_i, i = 0, \ldots, \overline{\overline{a}}_q$ are $m \times n$ constant matrices. Thus, Q and R are rational matrices with elements having the forms:

$$\begin{aligned} Q_{ij}(x) &= \frac{\sum_{k=0}^{\overline{q}_q} \overline{q}_{k,i,j} x^k}{\sum_{k=0}^{\overline{\overline{q}}_q} \overline{\overline{q}}_{k,i,j} x^k}, \quad 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant s, \\ R_{ij}(x) &= \frac{\sum_{k=0}^{\overline{r}_q} \overline{r}_{k,i,j} x^k}{\sum_{k=0}^{\overline{\overline{r}}_q} \overline{\overline{r}}_{k,i,j} x^k}, \quad 1 \leqslant i \leqslant s,\ i \leqslant j \leqslant n, \end{aligned} \tag{2.9}$$

where $\overline{q}_q$ and $\overline{r}_q$ (resp. $\overline{\overline{q}}_q$ and $\overline{\overline{r}}_q$) are maximal exponents of the numerators (resp. denominators) of the matrices Q and R, respectively. Notice that here and in the rest of the paper, variables with one bar determine numerator's coefficients and variables with two bars are used to denote denominator's coefficients.

Theorem 2.1. *Let $A \in \mathbb{C}(x)^{m\times n}_r$, $W \in \mathbb{C}(x)^{n\times m}_s$ be rational matrices, where $s \leqslant r$. Consider the full-rank QDR decomposition of the matrix W where the elements of the matrices Q, R are of the forms (2.9). Assuming that the condition (2.1) holds, let us denote an arbitrary (i, j)th element of the inverse matrix $N = (RAQ)^{-1}$ by*

$$N_{ij}(x) = \frac{\sum_{k=0}^{\overline{n}_q} \overline{n}_{k,i,j} x^k}{\sum_{k=0}^{\overline{\overline{n}}_q} \overline{\overline{n}}_{k,i,j} x^k}. \tag{2.10}$$

Then an arbitrary (i,j)th element of $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ for $x \in \mathbb{C}_s(W)$ can be evaluated as

$$\left(A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}\right)_{ij}(x) = \frac{\overline{\Gamma}_{ij}}{\overline{\overline{\Gamma}}_{ij}},$$

where

$$\overline{\Gamma}_{ij}(x) = \sum_{t=0}^{\overline{\overline{\Gamma}}_q - \overline{\overline{\alpha}}_q + \overline{\alpha}_q} \left(\sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{s} \sum_{t_1=0}^{t} \overline{\alpha}_{t_1,i,j,k,l} \gamma_{t-t_1,i,j,k,l} \right) x^t, \tag{2.11}$$

$$\overline{\overline{\Gamma}}_{ij}(x) = \text{Polynomial LCM} \left\{ \sum_{t=0}^{\overline{\overline{\alpha}}_q} \overline{\overline{\alpha}}_{t,i,j,k,l} x^t \middle| k = \overline{1, \min\{j,s\}},\ l = \overline{1,s} \right\} = \sum_{t=0}^{\overline{\overline{\Gamma}}_q} \overline{\overline{\gamma}}_{t,i,j} x^t, \tag{2.12}$$

where for $k = \overline{1, \min\{j,s\}}$, $l = \overline{1,s}$, values $\gamma_{t,i,j,k,l}$, $0 \leqslant t \leqslant \overline{\overline{\Gamma}}_q - \overline{\overline{\alpha}}_q$ are the coefficients of the polynomial

$$\Gamma_{i,j,k,l}(x) = \frac{\overline{\overline{\Gamma}}_{ij}(x)}{\sum_{t=0}^{\overline{\overline{\alpha}}_q} \overline{\overline{\alpha}}_{t,i,j,k,l} x^t}$$

and the next two notations are used:

$$\overline{\alpha}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{q}_{t_2,i,l} \overline{n}_{t_1-t_2-t_3,l,k} \overline{r}_{t_3,k,j}, \quad 0 \leqslant t \leqslant \overline{\alpha}_q = \overline{q}_q + \overline{n}_q + \overline{r}_q, \tag{2.13}$$

$$\overline{\overline{\alpha}}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{\overline{q}}_{t_2,i,l} \overline{\overline{n}}_{t_1-t_2-t_3,l,k} \overline{\overline{r}}_{t_3,k,j}, \quad 0 \leqslant t \leqslant \overline{\overline{\alpha}}_q = \overline{\overline{q}}_q + \overline{\overline{n}}_q + \overline{\overline{r}}_q. \tag{2.14}$$

Proof. We assume that the inverse matrix $N = (RAQ)^{-1} = \{N_{ij}(x)\}_{i,j=0}^{s}$ is determined by (2.10). Then

$$\begin{aligned}(QN)_{ij}(x) &= \sum_{l=0}^{s} Q_{il}(x) N_{lj}(x) = \sum_{l=1}^{s} \frac{\sum_{k=0}^{\overline{q}_q} \overline{q}_{k,i,l} x^k}{\sum_{k=0}^{\overline{\overline{q}}_q} \overline{\overline{q}}_{k,i,l} x^k} \frac{\sum_{k=0}^{\overline{n}_q} \overline{n}_{k,l,j} x^k}{\sum_{k=0}^{\overline{\overline{n}}_q} \overline{\overline{n}}_{k,l,j} x^k} \\ &= \sum_{l=1}^{s} \frac{\sum_{k=0}^{\overline{q}_q + \overline{n}_q} \left(\sum_{k_1=0}^{k} \overline{q}_{k_1,i,l} \overline{n}_{k-k_1,l,j} \right) x^k}{\sum_{k=0}^{\overline{\overline{q}}_q + \overline{\overline{n}}_q} \left(\sum_{k_1=0}^{k} \overline{\overline{q}}_{k_1,i,l} \overline{\overline{n}}_{k-k_1,l,j} \right) x^k}.\end{aligned}$$

Next we have the following calculations:

$$\begin{aligned}(Q(RAQ)^{-1}R)_{ij}(x) &= \sum_{k=1}^{\min\{j,s\}} (QN)_{ik}(x) R_{kj}(x) \\ &= \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{s} \frac{\sum_{t_1=0}^{\overline{q}_q + \overline{n}_q} \left(\sum_{t_2=0}^{t_1} \overline{q}_{t_2,i,l} \overline{n}_{t_1-t_2,l,k} \right) x^{t_1}}{\sum_{t_1=0}^{\overline{\overline{q}}_q + \overline{\overline{n}}_q} \left(\sum_{t_2=0}^{t_1} \overline{\overline{q}}_{t_2,i,l} \overline{\overline{n}}_{t_1-t_2,l,k} \right) x^{t_1}} \frac{\sum_{t_2=0}^{\overline{r}_q} \overline{r}_{t_2,k,j} x^{t_2}}{\sum_{t_2=0}^{\overline{\overline{r}}_q} \overline{\overline{r}}_{t_2,k,j} x^{t_2}} \\ &= \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{s} \frac{\sum_{t_1=0}^{\overline{q}_q + \overline{n}_q + \overline{r}_q} \left(\sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{q}_{t_2,i,l} \overline{n}_{t_1-t_2-t_3,l,k} \overline{r}_{t_3,k,j} \right) x^{t_1}}{\sum_{t_1=0}^{\overline{\overline{q}}_q + \overline{\overline{n}}_q + \overline{\overline{r}}_q} \left(\sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \overline{\overline{q}}_{t_2,i,l} \overline{\overline{n}}_{t_1-t_2-t_3,l,k} \overline{\overline{r}}_{t_3,k,j} \right) x^{t_1}}.\end{aligned}$$

Finally, according to the Eq. (2.3), for $x \in \mathbb{C}_s(W)$ the (i,j)th element of the inverse $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ is defined by:

$$\left(A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}\right)_{ij} = \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{s} \frac{\sum_{t=0}^{\overline{\alpha}_q} \overline{\alpha}_{t,i,j,k,l} x^t}{\sum_{t=0}^{\overline{\overline{\alpha}}_q} \overline{\overline{\alpha}}_{t,i,j,k,l} x^t} = \frac{\overline{\Gamma}_{ij}}{\overline{\overline{\Gamma}}_{ij}}, \tag{2.15}$$

where the denominator and numerator polynomials are computed by

$$\overline{\overline{\Gamma}}_{ij}(x) = \text{PolynomialLCM}\left\{\sum_{t=0}^{\overline{\overline{\alpha}}_q} \overline{\overline{\alpha}}_{t,i,j,k,l}x^t \,|\, k = \overline{1, \min\{j, s\}},\ l = \overline{1, s}\right\} = \sum_{t=0}^{\overline{\overline{\Gamma}}_q} \overline{\overline{\gamma}}_{t,i}x^t,$$

$$\overline{\Gamma}_{ij}(x) = \sum_{k=1}^{j}\sum_{l=1}^{s}\left(\Gamma_{i,j,k,l}(x)\sum_{t=0}^{\overline{\alpha}_q}\overline{\alpha}_{t,i,j,k,l}x^t\right),$$

where each polynomial $\Gamma_{i,j,k,l}(x)$ is equal to

$$\overline{\overline{\Gamma}}_{ij}(x) \Big/ \sum_{t=0}^{\overline{\overline{\alpha}}_q}\overline{\overline{\alpha}}_{t,i,j,k,l}x^t = \sum_{t=0}^{\overline{\overline{\Gamma}}_q-\overline{\overline{\alpha}}_q} \gamma_{t,i,j,k,l}x^t.$$

Therefore,

$$\overline{\Gamma}_{ij}(x) = \sum_{t=0}^{\overline{\overline{\Gamma}}_q-\overline{\overline{\alpha}}_q+\overline{\alpha}_q}\left(\sum_{k=1}^{\min\{j,s\}}\sum_{l=1}^{s}\sum_{t_1=0}^{t}\overline{\alpha}_{t_1,i,j,k,l}\gamma_{t-t_1,i,j,k,l}\right)x^t,$$

which coincides with the Eq. (2.11), and the proof is complete. □

Now, we are able to state an algorithm for the computation of the generalized inverse of a rational matrix. This algorithm uses the procedure for the evaluation of the full-rank *QDR* decomposition, and Algorithm 3.2 from [18] to compute the inverse $N^{-1}(x)$ of a given polynomial matrix $N(x)$. Notice that, in order to apply the Algorithm 3.2 from [18] to the rational matrix *RAQ*, one needs to transform it to the form involving the quotient of a polynomial matrix and a polynomial, in which case the polynomial acts as a constant in the evaluation of the inverse matrix.

Algorithm 2.3 Computation of a rational matrix $A_{T,S}^{(2)}$ inverse by using *QDR* factorization **(Algorithm QDRATS2)**

```
Require: Rational matrix $A(x) \in \mathbb{C}(x)_r^{m\times n}$.
1: Choose an arbitrary but fixed $n \times m$ rational matrix $W$ of normal rank $s \leqslant r$.
2: Generate the full-rank QDR decomposition of the matrix $W$ using the Algorithm 2.1. Transform the
   rational matrices $Q$, $R$ into the general form (2.9).
3: Transform the rational matrix $M = RAQ$ into the form:
      $M = \frac{1}{p(x)}M_1,$
   where $p(x)$ is a polynomial and $M_1$ is a polynomial matrix.
4: Find the inverse of the matrix $M_1$ using the Algorithm 3.2 from [18]. Generate the inverse matrix
   $N = M^{-1} = p(x)M_1^{-1}$, and transform it to the form:
      $N_{ij}(x) = \frac{\sum_{k=0}^{\overline{n}_q}\overline{n}_{k,i,j}x^k}{\sum_{k=0}^{\overline{\overline{n}}_q}\overline{\overline{n}}_{k,i,j}x^k}.$
5: Make the notations $\overline{\alpha}_q = \overline{q}_q + \overline{n}_q + \overline{r}_q,\ \overline{\overline{\alpha}}_q = \overline{\overline{q}}_q + \overline{\overline{n}}_q + \overline{\overline{r}}_q$, and for $i = \overline{1, n},\ j = \overline{1, m}$ perform
   Step 5.1–Step 5.5.
    5.1: For $k = \overline{1, \min\{j, s\}},\ l = \overline{1, s}$ do the following calculations:
      $\overline{\alpha}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1}\sum_{t_3=0}^{t_1-t_2}\overline{q}_{t_2,i,l}\overline{n}_{t_1-t_2-t_3,l,k}\overline{r}_{t_3,k,j}, \quad 0 \leqslant t \leqslant \overline{\alpha}_q,$
      $\overline{\overline{\alpha}}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1}\sum_{t_3=0}^{t_1-t_2}\overline{\overline{q}}_{t_2,i,l}\overline{\overline{n}}_{t_1-t_2-t_3,l,k}\overline{\overline{r}}_{t_3,k,j}, \quad 0 \leqslant t \leqslant \overline{\overline{\alpha}}_q.$
```

5.2: Evaluate the denominator polynomial of the (i,j)th element of $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ as

$$\text{PolynomialLCM}\left\{\sum_{t=0}^{\overline{\overline{\alpha}}_q}\overline{\overline{\alpha}}_{t,i,j,k,l}x^t \,|\, k=\overline{1,\min\{j,s\}},\ l=\overline{1,s}\right\}, \tag{2.16}$$

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

5.3: For each $k=\overline{1,\min\{j,s\}},\ l=\overline{1,s}$ compute the polynomial $\overline{\overline{\Gamma}}_{ij}(x)/\sum_{t=0}^{\overline{\overline{\alpha}}_q}\overline{\overline{\alpha}}_{t,i,j,k,l}x^t$ and denote it by $\Gamma_{i,j,k,l}(x)=\sum_{t=0}^{\overline{\overline{\Gamma}}_q-\overline{\overline{\alpha}}_q}\gamma_{t,i,j,k,l}x^t$.

5.4: Calculate the numerator polynomial of the (i,j)th element of $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ as

$$\overline{\Gamma}_{ij}(x)=\sum_{t=0}^{\overline{\overline{\Gamma}}_q-\overline{\overline{\alpha}}_q+\overline{\alpha}_q}\left(\sum_{k=1}^{\min\{j,s\}}\sum_{l=1}^{s}\sum_{t_1=0}^{t}\overline{\alpha}_{t_1,i,j,k,l}\gamma_{t-t_1,i,j,k,l}\right)x^t.$$

5.5: Set the value of (i,j)th element of the matrix $A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)}$ to $\overline{\Gamma}_{ij}(x)/\overline{\overline{\Gamma}}_{ij}(x)$.

2.2. Implementation details

The complexity of the *QDR* decomposition is the same as the complexity of the *QR* decomposition. The implementation of Algorithm *QDRATS*2 is done in the package MATHEMATICA and presented in Appendix 4. We designed three main functions, called `QDRDecomposition[A_List]`, `QDRATS[A_List,W_List]` and `QDRAlgorithm[A_List,W_List]`, for testing and verification purposes.

The normal rank of a given rational matrix is computed using the standard MATHEMATICA function `MatrixRank`, which works on both numerical and symbolic matrices [23].

The polynomial $P(x)$ is primitive if all its coefficients are mutually co-prime. Rational functions can be stored as ordered pairs of primitive numerators and denominators. Notice that the simplification is crucial in Step 3 and Step 5.5, where the quotients of two polynomials are evaluated. The function *Simplify*[ ] performs a sequence of algebraic and other transformations on a given expression and returns the simplest form it finds [23]. The package MATHEMATICA is appropriate for symbolic calculations and has built-functions for manipulating with unevaluated expressions.

In procedural programming languages, this simplification can be done by using the greatest common divisor of two polynomials. The fast *gcd* algorithm considering Chinese remainder theorem and the simple Euclidean algorithm can be used for finding the greatest common divisor of two polynomials. Coefficients in intermediate results can expand greatly in performing the Euclidean algorithm for polynomial reminder. But, notice that one can evaluate the primitive part of the remainder. However, the primitive part calculation requires many greatest common divisors of coefficients which can also be large. Therefore, the Chinese Remainder Algorithm (CRA) can be used for the reconstruction of the *gcd* coefficients back to integers.

3. Experiments with polynomial and rational matrices

In the next few examples we will examine our algorithm and then test some different implementations in order to compare processor times for some random test matrices.

Example 3.1. Consider the following polynomial matrices

$$A = \begin{bmatrix} -4x^2-3 & 2-7x & 4 \\ -9x & 3x^2-3 & -5 \\ 9x^2-2x & 9x^2 & -5 \\ -4x^2-3 & 2-7x & 4 \end{bmatrix}, \quad W = \begin{bmatrix} 3 & 7x & 4 & 5 \\ -9x & 3x^2-3 & 5 & x+5 \\ -6 & -14x & -8 & -10 \end{bmatrix}.$$

The matrix W is chosen quite randomly, only with suitable dimensions. We have $r = \operatorname{nrank}(A) = 3$, $s = \operatorname{nrank}(W) = 2$. Algorithm 2.1 produces the following QDR factorization of W:

$$Q = \begin{bmatrix} 3 & \frac{9x\left(8x^2-1\right)}{9x^2+5} \\ -9x & \frac{15\left(8x^2-1\right)}{9x^2+5} \\ -6 & -\frac{18x\left(8x^2-1\right)}{9x^2+5} \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{81x^2+45} & 0 \\ 0 & \frac{9x^2+5}{45\left(1-8x^2\right)^2} \end{bmatrix},$$

$$R = \begin{bmatrix} 9\left(9x^2+5\right) & 3x\left(44-9x^2\right) & 60-45x & 75-9x(x+5) \\ 0 & \frac{45\left(1-8x^2\right)^2}{9x^2+5} & \frac{15(12x+5)\left(8x^2-1\right)}{9x^2+5} & \frac{15(16x+5)\left(8x^2-1\right)}{9x^2+5} \end{bmatrix}.$$

The expression `X = QDRAlgorithm[A,W]` produces the following outer inverse of A:

$$X = \left[\begin{matrix} -\frac{3\left(72x^4+108x^3-148x^2-3x+19\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{108x^4-875x^3+297x^2+98x-48}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ -\frac{3\left(172x^3-241x^2+39x+35\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{212x^4+199x^3+702x^2-59x-144}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{6\left(72x^4+108x^3-148x^2-3x+19\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & -\frac{2\left(108x^4-875x^3+297x^2+98x-48\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \end{matrix} \right.$$

$$\left. \begin{matrix} -\frac{3\left(72x^4+108x^3-148x^2-3x+19\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{108x^4-875x^3+297x^2+98x-48}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ -\frac{3\left(172x^3-241x^2+39x+35\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{212x^4+199x^3+702x^2-59x-144}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{6\left(72x^4+108x^3-148x^2-3x+19\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & -\frac{2\left(108x^4-875x^3+297x^2+98x-48\right)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \end{matrix} \right].$$

According to Lemma 2.1 we get

$$X = A^{(2)}_{\mathcal{R}(Q),\mathcal{N}(R)} = A^{(2)}_{\mathcal{R}(W),\mathcal{N}(W)}.$$

Using the MATHEMATICA's standard function `NullSpace` [23] we get

$$\mathcal{N}(R) = \begin{bmatrix} -\frac{8x^2-35x-15}{9\left(8x^2-1\right)} & -\frac{16x+5}{3\left(8x^2-1\right)} & 0 & 1 \\ -\frac{12x^2-35x-12}{9\left(8x^2-1\right)} & -\frac{12x+5}{3\left(8x^2-1\right)} & 1 & 0 \end{bmatrix}.$$

Also, one can verify

$$\mathcal{R}(Q) = \left\{ \frac{9x\left(8x^2-1\right)z}{9x^2+5} + 3y,\ \frac{15\left(8x^2-1\right)z}{9x^2+5} - 9x\,y,\ -\frac{18x\left(8x^2-1\right)z}{9x^2+5} - 6\,y \right\},$$

where y, z are arbitrary complex numbers.

On the other hand, the expression `QDRAlgorithm[A,Transpose[A]]` implies computations corresponding to the case of $W = A^T$, and produces the Moore–Penrose inverse of A:

$$A^{\dagger}=\begin{bmatrix} -\frac{15(2x^2+1)}{456x^4-30x^3-226x^2+188x+90} & \frac{-36x^2+35x-10}{228x^4-15x^3-113x^2+94x+45} & \frac{12x^2-35x-2}{228x^4-15x^3-113x^2+94x+45} & -\frac{15(2x^2+1)}{456x^4-30x^3-226x^2+188x+90} \\ \frac{5x(9x+7)}{456x^4-30x^3-226x^2+188x+90} & \frac{16x^2-8x-15}{228x^4-15x^3-113x^2+94x+45} & \frac{20x^2+36x+15}{228x^4-15x^3-113x^2+94x+45} & \frac{5x(9x+7)}{456x^4-30x^3-226x^2+188x+90} \\ \frac{3x(9x^3+25x^2-9x+2)}{456x^4-30x^3-226x^2+188x+90} & -\frac{x(36x^3-63x^2+59x-4)}{228x^4-15x^3-113x^2+94x+45} & \frac{3(4x^4+20x^2-6x-3)}{228x^4-15x^3-113x^2+94x+45} & \frac{3x(9x^3+25x^2-9x+2)}{456x^4-30x^3-226x^2+188x+90} \end{bmatrix}.$$

The same result is obtained for $W = A$ and $W = A^k,\ k \geqslant 2$, since the Drazin and group inverse are equal to $A^{\dagger}$.

In the following example we will consider a rational matrix which Moore–Penrose inverse and the group inverse are different.

Example 3.2. The following rational matrix

$$A_1=\begin{bmatrix} \frac{144(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 0 & \frac{108+175x+4x^2}{252+175x-1148x^2} \\ \frac{75+372x}{252+175x-1148x^2} & 1 & 0 & \frac{3(25+124x)}{7(-36-25x+164x^2)} \\ -\frac{99(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 1 & \frac{99(-1+8x^2)}{7(-36-25x+164x^2)} \\ \frac{144(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 0 & \frac{108+175x+4x^2}{252+175x-1148x^2} \end{bmatrix}$$

is equal to the product AX_1 of the matrix A from Example 3.1 and its outer inverse X_1:

$$X_1=\begin{bmatrix} -\frac{6(-452+395x+3418x^2-6852x^3+7344x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{-10+35x-36x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{-2-35x+12x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{3(356+1665x+3616x^2-11954x^3+3208x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \\ \frac{6(435+784x-2434x^2+976x^3+6000x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{-15-8x+16x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{15+36x+20x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{-2610-13524x-2861x^2+26449x^3+15660x^4}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \\ \frac{3(-297-982x+6631x^2+9197x^3-34020x^4+33264x^5+7200x^6)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{x(4-59x+63x^2-36x^3)}{45+94x-113x^2-15x^3+228x^4} & \frac{3(-3-6x+20x^2+4x^4)}{45+94x-113x^2-15x^3+228x^4} & \frac{3(297+478x-4713x^2-11626x^3+17045x^4-6139x^5+3132x^6)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \end{bmatrix}.$$

The matrix A_1 is idempotent and clearly satisfies $\operatorname{ind}(A_1) = 1$. The Moore–Penrose inverse of A_1 is generated in the case $W = A_1^T$, and it is equal to

$$A_1^{\dagger}=\begin{bmatrix} \frac{9(2873+4500x-26336x^2-11200x^3+108320x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{8100+53301x+65400x^2+1488x^3}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{99(-108-175x+860x^2+1400x^3+32x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{9(2873+4500x-26336x^2-11200x^3+108320x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} \\ \frac{3(-900-89x+50600x^2+143344x^3)}{95652+187200x-637438x^2+2800x^3+3908768x^4} & \frac{42201+37800x-457103x^2+1400x^3+1954384x^4}{47826+93600x-318719x^2+1400x^3+1954384x^4} & -\frac{297(-25-124x+200x^2+992x^3)}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{3(-900-89x+50600x^2+143344x^3)}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{99(36-175x-1444x^2+1400x^3+9248x^4)}{95652+187200x-637438x^2+2800x^3+3908768x^4} & -\frac{297(-25-124x+200x^2+992x^3)}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{38025+93600x-161903x^2+1400x^3+1327120x^4}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{99(36-175x-1444x^2+1400x^3+9248x^4)}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{42642+118800x-110783x^2-200200x^3+622672x^4}{95652+187200x-637438x^2+2800x^3+3908768x^4} & \frac{432(-25-124x+200x^2+992x^3)}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{14256(1-8x^2)^2}{47826+93600x-318719x^2+1400x^3+1954384x^4} & \frac{42642+118800x-110783x^2-200200x^3+622672x^4}{95652+187200x-637438x^2+2800x^3+3908768x^4} \end{bmatrix}.$$

For $W = A_1$, we gain the group inverse, which is equal to the primordial matrix A_1.

Table 1
Mean processing times (in seconds) comparing several known algorithms to Algorithm 2.2 and Algorithm 2.3.

Test matrix from [24]	A_5	A_6	A_7	S_5	S_6	S_7	F_5	F_6	F_7
PseudoInverse [23]	0.3	0.6	1.1	0.1	0.5	0.9	0.5	0.9	1.4
Partitioning [15]	0.1	0.2	0.5	0.1	0.2	0.4	0.1	0.2	0.6
Lev.-Faddeev [6,16]	0.0	0.1	0.3	0.0	0.1	0.2	0.0	0.1	0.4
LDLGInverse [13]	8.8	73.5	1535.4	6.4	300.2	–	12.1	200.4	–
ModCholesky [14]	9.0	79.8	1588.8	8.7	323.5	–	13.2	212.1	–
QDRATS	1.2	13.9	197.2	0.1	1.1	73.0	1.5	16.4	235.8
QDRATS2	1.1	13.3	194.3	0.1	0.9	70.2	1.4	15.2	231.2

Example 3.3. Let us consider the matrix

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

and choose $W = A^*$. We have $\operatorname{nrank}(W) = 2$ and $\mathcal{C}_2(W) = \mathbb{C}\backslash\{1, 0\}$, since

$$\operatorname{rank}(W(1)) = \operatorname{rank}(W(0)) = 1 < 2.$$

Notations $W(1)$ and $W(0)$ denote constant matrices obtained by replacing the symbol x by the values $x = 1$ and $x = 0$, respectively. The Moore–Penrose inverse of A is given by

$$A^\dagger = \begin{bmatrix} \frac{1}{2-2x} & \frac{1}{x} \\ \frac{1}{2-2x} & \frac{1}{x} \\ \frac{1}{x-1} & -\frac{1}{x} \end{bmatrix}.$$

Clearly $A^\dagger$ is not defined in the case $x \in \{1, 0\}$ (or equivalently, in the case $x \notin \mathcal{C}_2(W)$).

The comparison of different algorithms for symbolic computation of the Moore–Penrose inverse is presented in Table 1. CPU time was used as a criterion for comparing observed algorithms. Algorithm *QDRATS*2 is tested in the particular case $W = A^*$. The processing times are obtained by applying MATHEMATICA implementation of various algorithms for the pseudoinverse computation on some test matrices from [24].

The first row of the table contains the name of the test matrix from [24], proposed by Zielke, where three groups of test matrices (A, S and F) are examined. The last row contains the processing times required by the *QDRATS*2 algorithm. Notice that *QDRATS* and *QDRATS*2 algorithms are less effective than the partitioning method and the Leverrier–Faddeev algorithm, because of the several matrix multiplications required, where the intermediate results and coefficients can greatly expand. Compared to the algorithms based on Cholesky and LDL^* decomposition, our algorithm is superior. Again, the main reason is the smaller number of matrix multiplications and therefore a smaller number of necessary simplifications. Denote that the stroke '–' means a long processor time required for the computation.

4. Conclusions

An algorithm for symbolic computation of $A^{(2)}_{T,S}$ inverses of one-variable rational matrices is derived. In this way, we initiate symbolic computation of generalized inverses which exploits the *QDR* decomposition of the fixed matrix W. The usage of square-root entries is avoided, which is of essential importance in symbolic computation of polynomial and rational expressions. As far as we know, the Leverrier–Faddeev algorithm, the Greville's partitioning method and the LDL^* decomposition based method are used in symbolic computation of generalized inverses. Some comparative processing times are provided on the set of rational test matrices.

Appendix

Algorithm 2.1 is implemented by the following MATHEMATICA function.

```
QDRDecomposition[A_List] :=
  Module[
   {i, j, k = 0, n = Length[A], m = Length[A[[1]]], r = MatrixRank[A], Q, D, R, B = A, A1},
   Q = Table[0, {r}, {n}];
   D = Table[0, {r}, {r}];
   R = Table[0, {r}, {m}];
   For[i = 1, i ≤ r, i++,
    A1 = Transpose[B];
    k++;
    While[(k ≤ n) && (Norm[A1[[k]]] == 0), k++];
    NextCol = A1[[k]];
    Q[[i]] = Simplify[NextCol];
    For[j = i, j ≤ m, j++,
     R[[i]][[j]] = Simplify[NextCol.A1[[j]]];];
    D[[i]][[i]] = Simplify[1 / (Norm[NextCol]^2), Element[x, Reals]];
    B = A - Transpose[Q].D.R;
   ];
   Return[{ExpandNumerator[Together[Transpose[Q]]],
     ExpandNumerator[ExpandDenominator[D]], ExpandNumerator[R]}];
  ];
```

The implementation of Algorithm 2.2 (*QDRATS*) is given in the next code, assuming that the matrix equation (2.6) is solved using the standard function `Inverse`:

```
QDRATS[A_List, W_List] :=
  Module[{N1, Q, Diag, R},
   {Q, Diag, R} = QDRDecomposition[W];
   N1 = Inverse[R.A.Q] // Simplify;
   Return[Simplify[Q.N1.R]];
   ];
```

We report the MATHEMATICA implementation of Algorithm 2.3, called *QDRATS2*.

```
QDRAlgorithm[A_List, W_List] :=
  Module[{N1, Q, Diag, R, i, j, k, l, n, m, r1, p, q, f, Num, Den, s = MatrixRank[W]},
   {n, m} = Dimensions[W]; {Q, Diag, R} = QDRDecomposition[W];
   N1 = ExpandNumerator[ExpandDenominator[Together[Simplify[Inverse[R.A.Q]]]]];
   f = Num = Den = Table[0, {n}, {m}];
   For[i = 1, i ≤ n, i++,
    For[j = 1, j ≤ m, j++,
      For[k = 1, k ≤ Min[j, s], k++,
       For[l = 1, l ≤ s, l++,
         Num[[k, l]] = 0; Den[[k, l]] = 0;
         For[r1 = 0, r1 ≤
            Max[Exponent[Numerator[Q[[i, l]]], x], Exponent[Denominator[Q[[i, l]]], x]] +
            Max[Exponent[Numerator[N1[[l, k]]], x], Exponent[Denominator[N1[[l, k]]], x]] +
            Max[Exponent[Numerator[R[[k, j]]], x], Exponent[Denominator[R[[k, j]]], x]],
            r1++,
              r1  r1-r2
          p = ∑    ∑   Coefficient[Numerator[Q[[i, l]]], x, r2] *
             r2=0 r3=0
             Coefficient[Numerator[N1[[l, k]]], x, r1 - r2 - r3] *
             Coefficient[Numerator[R[[k, j]]], x, r3];
              r1  r1-r2
          q = ∑    ∑   Coefficient[Denominator[Q[[i, l]]], x, r2] *
             r2=0 r3=0
             Coefficient[Denominator[N1[[l, k]]], x, r1 - r2 - r3] *
             Coefficient[Denominator[R[[k, j]]], x, r3];
          Num[[k, l]] += p * x^r1; Den[[k, l]] += q * x^r1;
         ];];];
                                           Min[j,s]    s             Num[[k, l]]
      f[[i, j]] = Together[Simplify[          ∑        ∑ Simplify[ ------------ ]]];
                                             k=1      l=1            Den[[k, l]]
     ];];
   Return[f];
  ];
```

References

[1] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, second ed., Springer, 2003.

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

[3] Y. Chen, X. Chen, Representation and approximation of the outer inverse $A_{T,S}^{(2)}$ of a matrix A, Linear Algebra Appl. 308 (2000) 85–107.

[4] G. Fragulis, B.G. Mertzios, A.I.G. Vardulakis, Computation of the inverse of a polynomial matrix and evaluation of its Laurent expansion, Internat. J. Control 53 (1991) 431–443.

[5] T.N.E. Greville, Some applications of the pseudo-inverse of matrix, SIAM Rev. 3 (1960) 15–22.

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

[7] N.P. Karampetakis, Computation of the generalized inverse of a polynomial matrix and applications, Linear Algebra Appl. 252 (1997) 35–60.

[8] N.P. Karampetakis, Generalized inverses of two-variable polynomial matrices and applications, Circuits Systems Signal Process. 16 (1997) 439–453.

[9] V.N. Katsikis, D. Pappas, A. Petralias, An improved method for the computation of the Moore–Penrose inverse matrix, Appl. Math. Comput. 217 (2011) 9828–9834.

[10] M.D. Petković, P.S. Stanimirović, M.B. Tasić, Effective partitioning method for computing weighted Moore–Penrose inverse, Comput. Math. Appl. 55 (2008) 1720–1734.

[11] M.D. Petković, P.S. Stanimirović, Symbolic computation of the Moore–Penrose inverse using partitioning method, Int. J. Comput. Math. 82 (2005) 355–367.

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

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

[14] P.S. Stanimirović, M.B. Tasić, Computing generalized inverses using LU factorization of matrix product, Int. J. Comput. Math. 85 (2008) 1865–1878.

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

[16] P.S. Stanimirović, M.B. Tasić, On the Leverrier–Faddeev algorithm for computing the Moore–Penrose inverse, J. Appl. Math. Comput. 35 (2011) 135–141.

[17] P.S. Stanimirović, D.S. Djordjević, Full-rank and determinantal representation of the Drazin inverse, Linear Algebra Appl. 311 (2000) 31–51.

[18] 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.

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

[20] G.R. Wang, Y.L. Chen, A recursive algorithm for computing the weighted Moore–Penrose inverse $A_{MN}^{\dagger}$, J. Comput. Math. 4 (1986) 74–85.

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

[22] Y. Wei, A characterization and representation of the generalized inverse $A_{T,S}^{(2)}$ and its applications, Linear Algebra Appl. 280 (1998) 87–96.

[23] S. Wolfram, The MATHEMATICA Book, fifth ed., Wolfram Media/Cambridge University Press, Champaign, IL 61820, USA, 2003.

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