

# Reliable Numerical Methods for Polynomial Matrix Triangularization <sup>†</sup>

DIDIER HENRION<sup>‡</sup>

LAAS-CNRS  
7 Avenue du Colonel Roche  
31 077 Toulouse, Cedex 4  
France

MICHAEL ŠEBEK

Trnka Laboratory of Automatic Control  
Faculty of Electrical Engineering  
Czech University of Technology  
Institute of Information Theory and Automation  
Academy of Sciences of the Czech Republic  
182 08 Prague 8, Czech Republic

**Abstract** : Numerical procedures are proposed for triangularizing polynomial matrices over the field of polynomial fractions and over the ring of polynomials. They are based on two standard polynomial techniques: Sylvester matrices and interpolation. In contrast to other triangularization methods, the algorithms described in this paper only rely on well-worked numerically reliable tools. They can also be used for greatest common divisor extraction, polynomial rank evaluation or polynomial null-space computation.

**Key Words** : Triangularization, Polynomial Matrices, Numerical Methods.

---

<sup>†</sup>This work is part of the Barrande Project No. 97/005-97/026. It was also supported by the Grant Agency of the Czech Republic under contract No. 102/97/0861, by the Ministry of Education of the Czech Republic under contract No. VS97/034 and by the French Ministry of Education and Research under contract No. 10-INSA-96.

<sup>‡</sup>corresponding author. E-mail: [henrion@laas.fr](mailto:henrion@laas.fr). FAX: (33 5) 61 33 69 69.

# 1 Introduction

A common theme when performing computations involving polynomials is to transform a given polynomial matrix into a simpler equivalent form while preserving key properties of the original matrix. The triangularization over the ring of polynomials is a fundamental construction in this regard.

Reduction of polynomial matrices into triangular form has crucial relevance in linear system theory. Applications include solving matrix polynomial equations such as the Diophantine equation frequently arising in control problems [11, 12], extracting the greatest common divisor of polynomial matrices [4, 5, 9], computing the determinant of a polynomial matrix [18], as well as obtaining a triangular basis for modules over polynomial domains [4].

The first results were obtained by Hermite for matrices with integer entries and it was only realized later that they also held for entries that are polynomials. This led to the classical triangularization algorithm exposed in [4, 5, 9], a simple variation of Gaussian elimination with the extended Euclidean algorithm replacing division. It is well-known that this procedure results impractical for its bad numerical behavior.

The idea of converting triangularization over the ring of polynomials to that of solving linear systems over the field of reals appears to have first been used by Kaltofen *et al.* [10]. The goal of the authors was to establish a parallel complexity result and Storjohann [19, Chapter 4] reexamined their approach to obtain sequential results. The key point was to note that one is led to a linear system that does not involve unknown coefficients of the triangular form, but only coefficients of the transformation matrix. Parallel and sequential complexity bounds are also proposed in [20], where the triangular form is derived from the Popov form, also widely used in linear system control theory.

The technique of solving a problem involving polynomial matrices without using polynomial operations was also used in the early reference [3]. In this paper, Bitmead *et al.* showed that the extraction of the greatest common divisor of polynomial matrices can be performed over the field of reals. Their procedure, based on the theory of sub-resultants and Sylvester matrices, was recently specialized to triangularization in a particularly elegant way by Labhalla *et al.* [15].

Although it has not rigorously been proved, one can reasonably argue that the approach described in [3, 15] features numerical shortcomings that are similar to those of the standard Euclidean division algorithm. Indeed, the algorithm found in these references is designed to preserve the shift-invariant Toeplitz structure of the Sylvester matrix and is actually a modified form of Gaussian elimination with only partial pivoting. As a result, complete numerical stability cannot be assured. In [3] it is stated that this seems to be a fundamental limitation since one is naturally constrained to use unimodular rather than orthogonal transformations when performing polynomial reductions.

In a more general context, the interpolation approach, clearly exposed in [1] by Antsaklis and Gao has been shown to supply solutions to the most part of control problems involving polynomial matrices. Interpolation techniques also provide, in an elegant and flexible way, alternative interpretations of well-known linear system theory results. Applications include the resolution of matrix polynomial equations like for instance the symmetric matrix polynomial equation [7] or the Diophantine equation, pole assignment by state or output feedback,  $J$ -spectral factorization [13], or also  $\mathcal{H}_\infty$  formulation of the optimal control problem. The above are just a few of the many examples of the strong presence of interpolation results in the system and control literature. It should thus seem surprising that, up to the authors' knowledge, no results are available so far concerning an interpolation approach to triangularization of polynomial matrices.

Based on the above considerations, the objective of this paper is twofold. On the one hand, this paper fulfills the need of polynomial matrix triangularization algorithms that are numerically reliable, in contrast to already published methods. On the other hand, it aims at extending the domain of application of interpolation by showing that this useful technique also applies to triangularization. More specifically, two almost equivalent numerical procedures are proposed. The first one is based on Sylvester matrices, whereas the second one exclusively relies on interpolation theory.

The paper is organized as follows. In Section 2 classical facts are recalled on the reduction into triangular form of matrices over the field of polynomial fractions and over the ring of polynomials. A useful lemma is then exposed that provides an upper bound on the degree of the reduction matrix. In Section 3, equivalence is established between triangularization over the field of polynomial fractions and triangularization over the fields of real numbers. Sylvester matrices and interpolation theory are used for that purpose. Both approaches are shown to be closely related. Several aspects concerning interpolation are dealt with, such as the choice and the number of interpolation points. Two numerically stable algorithms are described for triangularization over the field of polynomial fractions. They can alternatively be used for polynomial rank evaluation and polynomial null-space extraction. In Section 4, some algebraic complements about degree minimization over ideals are exposed. They will be instrumental to the derivation of the triangular form over the ring of polynomials, using equivalently Sylvester matrices or interpolation theory. Two algorithms are then proposed that only rely on numerically stable routines. The techniques proposed in this paper are illustrated on a very simple example in Section 5. Finally, comparative numerical results are exposed in Section 6. They aim at illustrating advantages of our approach over other triangularization methods.

The algorithms presented here have been tested experimentally by running programs written in MATLAB<sup>1</sup> code as part of the Polynomial Control Toolbox [14, 17]. They should be included to the next release of the Toolbox. In the meanwhile, they are available upon request.

**Notations** : The following notations will be used throughout the paper.  $\otimes$  denotes

---

<sup>1</sup>MATLAB is a registered trademark of The MathWorks, Inc.

the Kronecker product,  $I_n$  is the identity matrix of dimension  $n$  and  $A'$  is the transpose of matrix  $A$ .  $\mathbb{R}$  is the field of real numbers,  $\mathbb{R}[\mathbf{s}]$  is the ring of polynomials of  $\mathbf{s}$  with coefficients in  $\mathbb{R}$  and  $\mathbb{R}(\mathbf{s})$  is the field of rational fractions of  $\mathbf{s}$  with coefficients in  $\mathbb{R}$ . Finally, the notations  $\text{row}_i A(\mathbf{s})$ ,  $\text{col}_j A(\mathbf{s})$  and  $\text{ent}_{i,j} A(\mathbf{s})$  respectively stand for the  $i$ -th row,  $j$ -th column and  $i, j$ -th entry of polynomial matrix  $A(\mathbf{s})$ . The degree of a polynomial matrix  $A(\mathbf{s})$ , denoted by  $\deg A(\mathbf{s})$ , is the maximum degree of the entries of  $A(\mathbf{s})$ .

## 2 Preliminaries

In this section, the notion of a shape of a polynomial matrix is defined. Some standard facts on reduction of polynomial matrices into triangular forms are recalled. Distinction is made between triangular forms over field  $\mathbb{R}(\mathbf{s})$  and triangular forms over ring  $\mathbb{R}[\mathbf{s}]$ . Finally, a useful upper bound is derived on the degree of the polynomial matrix that is used to triangularize a polynomial matrix.

**Definition 1** *The shape  $\Sigma$  of a full column rank  $n$ -by- $m$  matrix  $A$  is the series  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  of positive integers  $\sigma_i = 1, 2, \dots, n$  such that  $\text{ent}_{\sigma_i, i} A$  is the uppermost non-zero entry of  $\text{col}_i A$ , called leading entry.*

**Definition 2** *A shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  whose components are strictly increasing, i.e.  $\sigma_1 < \dots < \sigma_m$ , is called a lower left triangular shape. A matrix with lower left triangular shape is in lower left triangular form.*

**Definition 3** *Elementary column operations for polynomial matrices are defined over  $\mathbb{R}(\mathbf{s})$  (respectively over  $\mathbb{R}[\mathbf{s}]$ ) as*

- *Interchange of any two columns.*
- *Addition to any column of a polynomial multiple of any other column.*
- *Scaling any column by any nonzero element of  $\mathbb{R}[\mathbf{s}]$  (resp.  $\mathbb{R}$ ).*

*These elementary operations can be represented by post-multiplication by a polynomial matrix invertible over  $\mathbb{R}(\mathbf{s})$ , or non-singular (resp. invertible over  $\mathbb{R}[\mathbf{s}]$ , or unimodular).*

**Theorem 1** *Any polynomial matrix  $A(\mathbf{s})$  can be transformed into lower left triangular form  $T(\mathbf{s})$  by application of elementary column operations over  $\mathbb{R}(\mathbf{s})$  (respectively over  $\mathbb{R}[\mathbf{s}]$ ), or equivalently by post-multiplication by a non-singular (resp. unimodular) reduction matrix  $U(\mathbf{s})$ , i.e.*

$$A(\mathbf{s})U(\mathbf{s}) = T(\mathbf{s}). \tag{1}$$

**Proof :** The proof is by construction and consists in an algorithm based on polynomial Euclidean division through elementary column operations. See [4, §2.3.4], [5, §G-3] or [9, §6.3.1].  $\square$

**Remark 1** A polynomial matrix  $A(\mathbf{s})$  has generally several triangular forms over  $\mathbb{R}(\mathbf{s})$  or  $\mathbb{R}[\mathbf{s}]$ . However, all triangular forms of  $A(\mathbf{s})$  have the same triangular shape. Indeed, if  $A(\mathbf{s})$  has two triangular forms with different shapes, one can check that it is not possible to obtain one triangular form from the other using elementary column operations only, thus leading to a contradiction.

**Remark 2** If every leading entry of a lower left triangular form  $T(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$  is monic and of higher degree than any entry at the left of it, then  $T(\mathbf{s})$  is in Hermite column form, also called reduced column echelon form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ . The Hermite form of a square matrix is unique, see e.g. [5, §G-3] or [9, §6.7.1].

Some important comments are in order.

- Suppose that the identity matrix is appended at the bottom of an arbitrary rank  $n$ -by- $m$  polynomial matrix  $A^\dagger(\mathbf{s})$  and let  $A(\mathbf{s})$  be the resulting full column rank compound matrix. Triangularization of  $A(\mathbf{s})$  reads

$$A(\mathbf{s})U(\mathbf{s}) = \begin{bmatrix} A^\dagger(\mathbf{s}) \\ I_m \end{bmatrix} U(\mathbf{s}) = \begin{bmatrix} T^\dagger(\mathbf{s}) \\ U(\mathbf{s}) \end{bmatrix} = T(\mathbf{s}). \quad (2)$$

In the above equation  $T^\dagger(\mathbf{s})$  is a triangular form of  $A^\dagger(\mathbf{s})$ . Hence a triangular form of a polynomial matrix of arbitrary rank and the corresponding reduction matrix have simultaneously been computed through triangularization of a full column rank matrix. Based on this and without loss of generality, in the paper only full column rank polynomial matrices will be considered.

- Throughout the paper only lower left triangular forms and elementary column operations are considered. Obviously, dual results also apply to upper right triangular forms and elementary row operations.
- In contrast to the triangular form over  $\mathbb{R}[\mathbf{s}]$ , the more general triangular form over  $\mathbb{R}(\mathbf{s})$  is apparently of no direct use in control theory. It is considered in this paper for two reasons. First, it may indirectly be used for polynomial matrix rank evaluation or polynomial null-space extraction. Second and most importantly, it proves to be a useful intermediate step towards the derivation of the triangular form over  $\mathbb{R}[\mathbf{s}]$ .
- The algorithms presented in this paper are only designed to compute triangular forms over  $\mathbb{R}(\mathbf{s})$  or  $\mathbb{R}[\mathbf{s}]$ , not necessarily the Hermite form. The reason behind this is numerical and will be exposed later on.

**Lemma 1** *Given a full column rank polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$ , let  $r_i \in [1, n]$ ,  $i = 1, \dots, n$  and  $c_j \in [1, m]$ ,  $j = 1, \dots, m$  be two series of integers such that  $\deg \text{row}_{r_1} A(\mathbf{s}) \geq \dots \geq \deg \text{row}_{r_n} A(\mathbf{s})$  and  $\deg \text{col}_{c_1} A(\mathbf{s}) \geq \dots \geq \deg \text{col}_{c_m} A(\mathbf{s})$ . Define*

$$\hat{d}_U = \min \left\{ \sum_{i=1}^{m-1} \deg \text{row}_{r_i} A(\mathbf{s}), \sum_{j=1}^{m-1} \deg \text{col}_{c_j} A(\mathbf{s}) \right\}. \quad (3)$$

*Then there always exists a reduction matrix  $U(\mathbf{s})$  of degree  $d_U \leq \hat{d}_U$  such that  $T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ , hence over  $\mathbb{R}(\mathbf{s})$ .*

**Proof :** The proof is adapted from [15, Proposition 4] and [19, Theorem 6]. Let  $U(\mathbf{s})$  be a unimodular matrix such that  $T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s})$  is in lower left triangular form, and let  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  be the shape of  $T(\mathbf{s})$  from which is defined the non-singular matrix

$$A_\Sigma(\mathbf{s}) = \begin{bmatrix} \text{row}_{\sigma_1} A(\mathbf{s}) \\ \vdots \\ \text{row}_{\sigma_m} A(\mathbf{s}) \end{bmatrix}$$

and the non-singular lower left triangular matrix  $T_\Sigma(\mathbf{s}) = A_\Sigma(\mathbf{s})U(\mathbf{s})$ . We aim to show that  $\deg U(\mathbf{s}) \leq \hat{d}_U$ . Since  $U(\mathbf{s})$  is unimodular and  $T_\Sigma(\mathbf{s})$  is square triangular,  $\deg \det A_\Sigma(\mathbf{s}) = \deg \det T_\Sigma(\mathbf{s}) = \sum_{i=1}^m \deg \text{col}_i T(\mathbf{s})$ . Now let  $A_\Sigma^{\text{adj}}(\mathbf{s})$  denote the polynomial adjoint matrix of  $A_\Sigma(\mathbf{s})$ , so that  $\det A_\Sigma(\mathbf{s})U(\mathbf{s}) = A_\Sigma^{\text{adj}}(\mathbf{s})T_\Sigma(\mathbf{s})$ . The latter equation yields

$$\begin{aligned} d_U = \deg U(\mathbf{s}) &\leq \deg A_\Sigma^{\text{adj}}(\mathbf{s}) + \deg T_\Sigma(\mathbf{s}) - \deg \det A_\Sigma(\mathbf{s}) \\ &\leq \deg A_\Sigma^{\text{adj}}(\mathbf{s}) + \max_{i=1, \dots, m} \deg \text{col}_i T(\mathbf{s}) - \sum_{i=1}^m \deg \text{col}_i T(\mathbf{s}) \\ &\leq \deg A_\Sigma^{\text{adj}}(\mathbf{s}). \end{aligned}$$

In order to prove the lemma, it is now sufficient to show that  $\deg A_\Sigma^{\text{adj}}(\mathbf{s}) \leq \hat{d}_U$ , where  $\hat{d}_U$  is given in (3). To see that, note that entries of  $A_\Sigma^{\text{adj}}(\mathbf{s})$  are minors of order  $m - 1$  of  $A_\Sigma(\mathbf{s})$ . The degree of the minor obtained by removing the  $i$ -th row and  $j$ -th column of  $A(\mathbf{s})$  is less than or equal to  $\sum_{k=1}^m \deg \text{row}_k A_\Sigma(\mathbf{s}) - \deg \text{row}_i A_\Sigma(\mathbf{s})$  (row-wise) and is also less than or equal to  $\sum_{k=1}^m \deg \text{col}_k A_\Sigma(\mathbf{s}) - \deg \text{col}_j A_\Sigma(\mathbf{s})$  (column-wise). Upon applying this reasoning to all the entries in  $A_\Sigma^{\text{adj}}(\mathbf{s})$ , the required bound is obtained.  $\square$

### 3 Triangular form over $\mathbb{R}(\mathbf{s})$

In this section, equivalence is established between triangularization over  $\mathbb{R}(\mathbf{s})$  and over  $\mathbb{R}$ . For that purpose, two techniques are used that are standard when dealing with polynomial matrices: Sylvester matrices and interpolation. As a result, two algorithms for triangularization over  $\mathbb{R}(\mathbf{s})$  are proposed. Their respective numerical properties are discussed at length.

### 3.1 Sylvester matrix approach

Let  $\{e_1, e_2, \dots, e_n\}$  be the canonical basis of the free module  $\mathbb{R}^n[\mathbf{s}]$ . Given a polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$  of degree  $d_A$ , let  $\mathcal{A} \subset \mathbb{R}^n[\mathbf{s}]$  denote the submodule generated by the columns of  $A(\mathbf{s})$ . Further, let  $\mathbb{R}^n[\mathbf{s}]_d \subset \mathbb{R}^n[\mathbf{s}]$  represent the submodule spanned by all elements of  $\mathbb{R}^n[\mathbf{s}]$  of degree less than or equal to  $d$  and let  $\mathcal{A}_d = \mathcal{A} \cap \mathbb{R}^n[\mathbf{s}]_d$  be the corresponding submodule of  $\mathcal{A}$ .

Let  $P$  stand for the orthogonal permutation matrix of dimension  $n(d+1)$  such that, for arbitrary row vectors  $v_{ij}$

$$P[v'_{10} \cdots v'_{n0} | v'_{11} \cdots v'_{n1} | \cdots | v'_{1d} \cdots v'_{nd}]' = [v'_{10} v'_{11} \cdots v'_{1d} | \cdots | v'_{n0} v'_{n1} \cdots v'_{nd}]'$$

Note that  $P$  can be used to commute the Kronecker product [16]. Indeed, for any matrices  $X, Y$  of compatible dimensions

$$P(X \otimes Y) = (Y \otimes X)P. \quad (4)$$

The Sylvester matrix of order  $d_U$  of polynomial matrix  $A(\mathbf{s}) = A_0 + \mathbf{s}A_1 + \cdots + \mathbf{s}^{d_A}A_{d_A}$  is

$$\langle A(\mathbf{s}) \rangle_{d_U} = \begin{bmatrix} A_{d_A} & & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ A_1 & & A_{d_A} & 0 \\ A_0 & & \vdots & A_{d_A} \\ 0 & \ddots & A_1 & \vdots \\ \vdots & & A_0 & A_1 \\ 0 & & 0 & A_0 \end{bmatrix} \in \mathbb{R}^{n(d+1) \times m(d_U+1)} \quad (5)$$

where  $d = d_A + d_U$ . The row permuted Sylvester matrix of  $A(\mathbf{s})$  is denoted by

$$\mathbf{R}_d = P \langle A(\mathbf{s}) \rangle_{d_U}. \quad (6)$$

Finally, let

$$\{\{\mathbf{s}^d e_1, \dots, \mathbf{s} e_1, e_1\}, \dots, \{\mathbf{s}^d e_n, \dots, \mathbf{s} e_n, e_n\}\} \quad (7)$$

be a generating basis for  $\mathbb{R}^n[\mathbf{s}]_d$ . By construction, for any  $i = 1, \dots, n$  and  $j = 0, \dots, d_U$ , the  $(i + (d_U - j)n)$ -th column of matrix  $\mathbf{R}_d$  is a real vector representation with respect to basis (7) of the polynomial vector  $\text{col}_i(\mathbf{s}^j A(\mathbf{s})) \in \mathcal{A}_d \subset \mathbb{R}^n[\mathbf{s}]_d$ . Let  $\mathcal{R}_d \subset \mathcal{A}_d$  denote the submodule spanned by all the elements  $\text{col}_i(\mathbf{s}^j A(\mathbf{s}))$  for  $i = 1, \dots, n$  and  $j = 0, \dots, d_U$ . By a slight abuse of terminology, we will henceforth say that matrix  $\mathbf{R}_d$  generates submodule  $\mathcal{R}_d$  in basis (7).

**Lemma 2** *Triangularizing over  $\mathbb{R}(\mathbf{s})$  a full column rank polynomial matrix  $A(\mathbf{s})$  of degree  $d_A$  by a reduction matrix  $U(\mathbf{s})$  of degree  $d_U$  amounts to triangularizing over  $\mathbb{R}$  the row permuted Sylvester matrix  $\mathbf{R}_d$ , where  $d = d_A + d_U$ .*

**Proof :** Triangularization of  $A(\mathbf{s}) \in \mathbb{R}^{m \times n}[\mathbf{s}]$  by  $U(\mathbf{s})$  yields matrix polynomial equation (1) where  $T(\mathbf{s})$  has triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ . Let

$$U(\mathbf{s}) = ([\mathbf{s}^{d_U} \ \dots \ \mathbf{s} \ 1] \otimes I_m) \underbrace{\begin{bmatrix} U_{d_U} \\ \vdots \\ U_1 \\ U_0 \end{bmatrix}}_{\bar{U}}, \quad T(\mathbf{s}) = ([\mathbf{s}^d \ \dots \ \mathbf{s} \ 1] \otimes I_n) \underbrace{\begin{bmatrix} T_d \\ \vdots \\ T_1 \\ T_0 \end{bmatrix}}_{\bar{T}}. \quad (8)$$

By equating coefficients of increasing powers of  $\mathbf{s}$ , matrix polynomial equation (1) is equivalent to the linear system

$$\langle A(\mathbf{s}) \rangle_{d_U} \bar{U} = \bar{T}. \quad (9)$$

By construction, triangularization of  $\mathbf{R}_d$  over  $\mathbb{R}$  provides a triangular basis for submodule  $\mathcal{R}_d$  from which triangular matrix  $T(\mathbf{s})$  can be built. More precisely, suppose that  $\mathbf{U}$  is a non-singular matrix such that

$$\mathbf{R}_d \mathbf{U} = \mathbf{T} \quad (10)$$

has triangular shape  $\bar{\Sigma} = \{\bar{\sigma}_1, \dots, \bar{\sigma}_{m(d_U+1)}\}$ . Define column index sets

$$\mathcal{C}_i = \{k : i(d+1) - d \leq \bar{\sigma}_k \leq i(d+1)\}, \quad i = 1, \dots, n. \quad (11)$$

With these notations, a matrix  $\bar{U}$  such that  $\text{col}_i \bar{U} = \text{col}_{k_i} \mathbf{U}$  for any  $k_i \in \mathcal{C}_{\sigma_i}$ ,  $i = 1, \dots, m$  is solution to (9). Indeed each column  $\text{col}_{k_i} \mathbf{T}$  is a representation w.r.t. basis (7) of a polynomial vector of shape  $\{\sigma_i\}$  belonging to  $\mathcal{R}_d$ , that is to say, of a column  $\text{col}_i T(\mathbf{s})$  of a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ .  $\square$

Lemma 2 states that if the degree  $d_U$  of a matrix  $U(\mathbf{s})$  that reduces  $A(\mathbf{s})$  into triangular form  $T(\mathbf{s})$  is given, matrices  $U(\mathbf{s})$  and  $T(\mathbf{s})$  can be derived from triangularization (10). However, one usually does not know  $d_U$  beforehand, so let

$$\hat{d} = d_A + \hat{d}_U \quad (12)$$

where  $\hat{d}_U$  was defined in (3). Combination of Lemma 1 and Lemma 2 yields the following result.

**Theorem 2** *Triangularizing over  $\mathbb{R}(\mathbf{s})$  a full column rank polynomial matrix  $A(\mathbf{s})$  amounts to triangularizing over  $\mathbb{R}$  the row permuted Sylvester matrix  $\mathbf{R}_{\hat{d}}$ .*

**Proof :** Existence of a matrix  $U(\mathbf{s})$  of degree  $d_U$  greater than or equal to  $\hat{d}_U$  that brings  $A(\mathbf{s})$  into triangular form is guaranteed by Lemma 1. The links existing between triangularization of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  and triangularization of  $\mathbf{R}_{\hat{d}}$  over  $\mathbb{R}$  when  $\hat{d} = d_A + \hat{d}_U$  were stated in Lemma 2.  $\square$



**Corollary 1** *Suppose that a polynomial matrix  $A^\dagger(\mathbf{s})$  of degree  $d_A$  does not necessarily have full column rank. Build the row permuted Sylvester matrix  $\mathbf{R}_{\hat{d}}$  corresponding to the full column rank compound matrix  $A(\mathbf{s})$  in (2). Triangularization of  $\mathbf{R}_{\hat{d}}$  over  $\mathbb{R}$  allows to evaluate the rank of  $A^\dagger(\mathbf{s})$  and to compute a polynomial basis for the polynomial right null-space of  $A^\dagger(\mathbf{s})$ .*

**Proof :** Let  $d_A$  and  $m$  respectively denote the degree and the number of columns of  $A(\mathbf{s})$ . According to Theorem 1 and the proof of Lemma 2, triangularization (10) yields the triangular shape  $\Sigma$  of  $T(\mathbf{s})$ , a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ . The rank of  $A^\dagger(\mathbf{s})$  is then readily the number of components of  $\Sigma$  less than or equal to the number of rows of  $A^\dagger(\mathbf{s})$ . Similarly, the  $m - \text{rank } A^\dagger(\mathbf{s})$  rightmost columns of  $U(\mathbf{s})$  in (2) generate a polynomial right null-space of  $A^\dagger(\mathbf{s})$  of degree less than or equal to  $\hat{d}_U = \hat{d} - d_A$ .  $\square$

**Remark 3** Corollary 1 is an alternative to the method proposed in [2] where the null-space of a polynomial matrix is computed via a matrix pencil approach.

It must be noticed that Lemma 1 only provides an upper bound  $\hat{d}_U$  on  $d_U$ . Hence, a reduction matrix  $U(\mathbf{s})$  of degree  $d_U$  strictly less than  $\hat{d}_U$  may exist. Since the value of  $d_U$  is generally unknown, one can alternatively

- triangularize  $\mathbf{R}_{\hat{d}}$  only once, or
- triangularize  $\mathbf{R}_d$  successively for  $d = d_A + d_U$  and  $d_U = 0, 1, \dots, \hat{d}_U$  until a matrix  $U(\mathbf{s})$  of minimal possible degree  $d_U$  is found.

### Algorithm 1 : Triangularization over $\mathbb{R}(\mathbf{s})$ Sylvester matrix approach

- Input* A full column rank polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$  of degree  $d_A$ .  
*Output* A non-singular reduction matrix  $U(\mathbf{s}) \in \mathbb{R}^{m \times m}[\mathbf{s}]$  such that  $A(\mathbf{s})U(\mathbf{s}) = T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ .
- Step 0* Let  $d_U = \hat{d}_U$  as in (3).  
*Step 1* Build permuted Sylvester matrix  $\mathbf{R}_d$  as in (6), where  $d = d_A + d_U$ .  
*Step 2* Triangularize  $\mathbf{R}_d$  over  $\mathbb{R}$  as in (10).  
*Step 3* Extract a triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  such that sets  $\mathcal{C}_{\sigma_i}$  in (11) are not empty.  
*Step 4* Build matrix  $\bar{U}$  such that  $\text{col}_i \bar{U} = \text{col}_{k_i} \mathbf{U}$  for any  $k_i \in \mathcal{C}_{\sigma_i}$ ,  $i = 1, \dots, m$ .  
*Step 5* Recover  $U(\mathbf{s})$  as in (8).

**Algorithm 1' :Triangularization over  $\mathbb{R}(\mathbf{s})$   
Iterative Sylvester matrix approach**

Only the following steps differ from Algorithm 1

*Step 0* Let  $d_U = 0$ .

*Step 3* Extract a triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  such that sets  $\mathcal{C}_{\sigma_i}$  in (11) are not empty. If it is not possible, let  $d_U = d_U + 1$  and go to Step 1.

### 3.2 Interpolation Approach

An interpolation technique for triangularizing a polynomial matrix over  $\mathbb{R}(\mathbf{s})$  is now derived. Relationships with the Sylvester matrix approach and the standard interpolation procedure described in [1] are examined.

In this section, it is supposed that matrix coefficients of  $A(\mathbf{s}) \in \mathbb{R}^{m \times n}[\mathbf{s}]$  are not known. The polynomial matrix is rather given through constant values  $A(\mathbf{s}_i) \in \mathbb{R}^{m \times n}$  taken at distinct interpolation points  $\mathbf{s}_i \in \mathbb{R}$ ,  $i = 0, 1, \dots, d$ . Furthermore, it is assumed that column degrees, or at least the degree  $d_A$  of  $A(\mathbf{s})$ , are known.

Matrix polynomial equation (1) can be written

$$L(\mathbf{s})\bar{U} = T(\mathbf{s}) \quad (13)$$

where  $\bar{U}$  is defined in (9) and

$$L(\mathbf{s}) = [\mathbf{s}^{d_U} \ \cdots \ \mathbf{s} \ 1] \otimes A(\mathbf{s}). \quad (14)$$

Using relation (14) and values  $A(\mathbf{s}_i)$ , build the row permuted interpolated matrix

$$\mathbf{L}_d = P \begin{bmatrix} \mathbf{s}_d^{d_U} A(\mathbf{s}_d) & \cdots & \mathbf{s}_d A(\mathbf{s}_d) & A(\mathbf{s}_d) \\ \vdots & & & \vdots \\ \mathbf{s}_1^{d_U} A(\mathbf{s}_1) & \cdots & \mathbf{s}_1 A(\mathbf{s}_1) & A(\mathbf{s}_1) \\ \mathbf{s}_0^{d_U} A(\mathbf{s}_0) & \cdots & \mathbf{s}_0 A(\mathbf{s}_0) & A(\mathbf{s}_0) \end{bmatrix} = P \begin{bmatrix} L(\mathbf{s}_d) \\ \vdots \\ L(\mathbf{s}_1) \\ L(\mathbf{s}_0) \end{bmatrix}. \quad (15)$$

**Lemma 3** *Triangularizing over  $\mathbb{R}(\mathbf{s})$  a full column rank polynomial matrix  $A(\mathbf{s})$  of degree  $d_A$  by a reduction matrix  $U(\mathbf{s})$  of degree  $d_U$  amounts to triangularizing over  $\mathbb{R}$  the row permuted interpolated matrix  $\mathbf{L}_d$ , where  $d = d_A + d_U$ .*

**Proof :** One can easily check that matrix  $\mathbf{L}_d$  is a Sylvester matrix of order  $d_U$  multiplied to the left by a block Vandermonde matrix and a permutation matrix  $P$ , viz.

$$\mathbf{L}_d = P(M_d \otimes I_m)\langle A(\mathbf{s}) \rangle_{d_U} \quad (16)$$

where

$$M_d = \begin{bmatrix} \mathbf{s}_d^d & \cdots & \mathbf{s}_d & 1 \\ \vdots & & & \vdots \\ \mathbf{s}_1^d & \cdots & \mathbf{s}_1 & 1 \\ \mathbf{s}_0^d & \cdots & \mathbf{s}_0 & 1 \end{bmatrix}.$$

Note that Vandermonde matrix  $M_d$  is invertible since the  $\mathbf{s}_i$  are supposed to be distinct. Using relations (4) and (6), one gets

$$\mathbf{L}_d = (I_m \otimes M_d)P\langle A(\mathbf{s}) \rangle_{d_U} = (I_m \otimes M_d)\mathbf{R}_d.$$

With the notations of the proof of Lemma 1, matrix  $\mathbf{L}_d$  generates submodule  $\mathcal{R}_d$  in the basis

$$\{\{\mathbf{s}^d e_1, \dots, \mathbf{s} e_1, e_1\}M_d^{-1}, \dots, \{\mathbf{s}^d e_n, \dots, \mathbf{s} e_n, e_n\}M_d^{-1}\}.$$

Triangularization of  $\mathbf{L}_d$  over  $\mathbb{R}$  thus provides a triangular basis for submodule  $\mathcal{R}_d$  from which a triangular matrix  $T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s})$  of shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  can be built. Suppose that  $\mathbf{U}$  is a non-singular matrix such that

$$\mathbf{L}_d \mathbf{U} = \mathbf{T} \tag{17}$$

has triangular shape  $\bar{\Sigma} = \{\bar{\sigma}_1, \dots, \bar{\sigma}_{m(d_U+1)}\}$ . Define column index sets  $\mathcal{C}_{\sigma_i}$  as in (11). As in the proof of Lemma 1, a matrix  $\bar{\mathbf{U}}$  such that  $\text{col}_i \bar{\mathbf{U}} = \text{col}_{k_i} \mathbf{U}$  for any  $k_i \in \mathcal{C}_{\sigma_i}$ ,  $i = 1, \dots, m$  is solution to (13).  $\square$

**Remark 4** Antsaklis and Gao [1] proposed an interpolation technique to solve polynomial matrix equation (1) where  $U(\mathbf{s})$  has to be found and  $A(\mathbf{s}), T(\mathbf{s})$  are known through interpolated values  $A(\mathbf{s}_i), T(\mathbf{s}_i)$ . The triangularization problem faced here is quite different since  $U(\mathbf{s})$  is unknown, interpolated values  $A(\mathbf{s}_i)$  are given but interpolated values  $T(\mathbf{s}_i)$  are unknown. However, the way matrix  $\mathbf{L}_d$  is built in (15) parallels the development leading to Theorem 3.1 in [1]. More specifically, equations (13), (17) in this paper and equations (3.7), (3.9) in [1] are respectively equivalent. Following the notations of [1], each row of matrix  $\mathbf{L}_d$  corresponds to a special choice of interpolation triplets  $(a_j, b_j, c_j)$  such that

$$\text{row}_j \mathbf{L}_d = a_j L(c_j) = b_j, \quad j = 1, 2, \dots, n(d+1)$$

where

$$\begin{array}{ll} a_1 = a_2 = \dots = a_n = e_1 & c_1 = c_{n+1} = \dots = c_{nd+1} = \mathbf{s}_d \\ a_{n+1} = a_{n+2} = \dots = a_{2n} = e_2 & c_2 = c_{n+2} = \dots = c_{nd+2} = \mathbf{s}_{d-1} \\ \vdots & \vdots \\ a_{nd+1} = a_{nd+2} = \dots = a_{n(d+1)} = e_n & c_n = c_{2n} = \dots = c_{n(d+1)} = \mathbf{s}_0. \end{array}$$

**Remark 5** Matrix polynomial equation (1) evaluated at interpolation points  $s_0, s_1, \dots, s_d$  reads

$$A(\mathbf{s}_i)U(\mathbf{s}_i) = T(\mathbf{s}_i), \quad i = 0, 1, \dots, d$$

where the  $A(\mathbf{s}_i)$  are given and  $U(\mathbf{s}), T(\mathbf{s})$  are to be found. If one intends to compute  $U(\mathbf{s})$  and  $T(\mathbf{s})$  by

1. computing matrices  $\mathbf{U}_i$  such that matrices  $A(\mathbf{s}_i)\mathbf{U}_i = \mathbf{T}_i$  have the same triangular shape  $\Sigma$  for all  $i = 0, 1, \dots, d$
2. recovering matrices  $\hat{U}(\mathbf{s}), \hat{T}(\mathbf{s})$  from interpolation relations  $\hat{U}(\mathbf{s}_i) = \mathbf{U}_i, \hat{T}(\mathbf{s}_i) = \mathbf{T}_i, i = 0, 1, \dots, d,$

then  $\hat{T}(\mathbf{s})$  has indeed triangular shape  $\Sigma$  but generally the matrix polynomial equation  $A(\mathbf{s})\hat{U}(\mathbf{s}) = \hat{T}(\mathbf{s})$  does not hold. In other words, triangularization of a polynomial matrix  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  is not equivalent to successive triangularizations of interpolated matrices  $A(\mathbf{s}_i)$  over  $\mathbb{R}$ . According to matrix polynomial equation (13),  $U(\mathbf{s})$  is actually derived from a constant matrix  $\bar{U}$  solution to

$$L(\mathbf{s}_i)\bar{U} = T(\mathbf{s}_i), \quad i = 0, 1, \dots, d.$$

Triangularization of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  is therefore equivalent to a simultaneous triangularization of interpolated matrices  $L(\mathbf{s}_i)$  over  $\mathbb{R}$  by the same reduction matrix  $\bar{U}$ .

Lemma 3 states that once  $d_A$  and  $d_U$  are given, interpolated values  $A(\mathbf{s}_i), i = 0, 1, \dots, d = d_A + d_U$  can be used to derive a reduction matrix  $U(\mathbf{s})$  and a triangular form  $T(\mathbf{s})$  of  $A(\mathbf{s})$ . However, the degree of reduction matrix  $U(\mathbf{s})$  is generally not known beforehand. As a result, one usually does not know how many interpolation points are required. The following theorem provides an upper bound.

**Theorem 3** *Triangularizing over  $\mathbb{R}(\mathbf{s})$  a full column rank polynomial matrix  $A(\mathbf{s})$  amounts to triangularizing over  $\mathbb{R}$  the row permuted interpolated matrix  $\mathbf{L}_{\hat{d}}$ .*

**Proof :** Combine Lemma 1 and Lemma 3. □

In view of Lemma 3 and Theorem 3, the number of interpolated values  $A(\mathbf{s}_0), \dots, A(\mathbf{s}_d)$  that are needed to triangularize  $A(\mathbf{s})$  lies within the interval  $d_A \leq d \leq \hat{d}$ . Since a reduction matrix may be found for a value of  $d$  strictly less than  $\hat{d}$ , one can alternatively

- triangularize  $\mathbf{L}_{\hat{d}}$  only once, or
- triangularize  $\mathbf{L}_d$  successively for  $d = d_A, d_A + 1, \dots, \hat{d}$  until  $U(\mathbf{s})$  is found for the minimal possible value of  $d$ .

### Algorithm 2 : Triangularization over $\mathbb{R}(\mathbf{s})$ Interpolation Approach

- Input* Values  $A(\mathbf{s}_i) \in \mathbb{R}^{n \times m}$  of a full column rank polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$ , interpolated at distinct points  $\mathbf{s}_i \in \mathbb{R}$ ,  $i = 0, 1, 2, \dots$
- Output* A non-singular reduction matrix  $U(\mathbf{s}) \in \mathbb{R}^{m \times m}[\mathbf{s}]$  such that  $A(\mathbf{s})U(\mathbf{s}) = T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ .
- Step 0* Let  $d = \hat{d}$  as in (12).
- Step 1* Build interpolated matrix  $\mathbf{L}_d$  as in (15).
- Step 2* Triangularize  $\mathbf{L}_d$  over  $\mathbb{R}$  as in (17)
- Steps 3–4* See Algorithm 1.

**Algorithm 2' : Triangularization over  $\mathbb{R}(\mathbf{s})$   
Iterative interpolation approach**

Only the following steps differ from Algorithm 2.

- Step 0* Let  $d = d_A$ .
- Step 3* Extract triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  such that sets  $\mathcal{C}_{\sigma_i}$  in (11) are not empty. If it is not possible, let  $d = d + 1$  and go to Step 1.

### 3.3 Numerical Considerations

Algorithms 1 and 2 only rely upon numerically reliable procedures. Indeed, in both algorithms the crucial point is Step 2, where a matrix has to be triangularized over field  $\mathbb{R}$ . This triangularization can be performed using powerful and reliable numerical procedures such as Householder, modified Gram-Schmidt or Givens orthogonalizations – the so-called QR decomposition, where the reduction matrix is orthogonal [6].

The conditioning of interpolated matrix  $\mathbf{L}_d$  in Step 1 of Algorithm 2 clearly depends on the choice of interpolation points and more specifically on the conditioning of Vandermonde matrix  $M_d$  – see equation (16). If interpolation points  $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_d$  lie on the real axis then the condition number of  $M_d$  grows at least exponentially with  $d$ . The slower growth rate is achieved if the  $\mathbf{s}_i$  are the zeros of the  $(d + 1)$ -th degree Chebyshev polynomial. If the  $\mathbf{s}_i$  are the complex roots of unity,  $M_d$  is perfectly conditioned [8, §21.1] and Algorithm 1 and Algorithm 2 are numerically equivalent. Note however that the possibility of using complex interpolation points stays out of the scope of this paper and is subject to current research.

How to choose between non-iterative Algorithm 1 and iterative Algorithm 1' will depend on the problem being treated. Suppose that there exists a reduction matrix of low degree  $d_V^* \ll \hat{d}_V$ . Since matrix  $\mathbf{R}_d$  is  $n(d_A + d_V + 1)$  by  $m(d_V + 1)$ , a single triangularization for a high value of  $d_V = \hat{d}_V$  can easily be more demanding than several triangularizations for low values of  $d_V = 0, 1, \dots, d_V^*$ . The same comment holds for Algorithms 2 and 2', matrix  $\mathbf{L}_d$  and parameter  $d$ .

In Algorithms 1' and 2' matrices of increasing dimensions are triangularized until the lowest degree reduction matrix is found. At each step, the algorithms do not exploit the results computed at previous steps. As far as computational load is concerned, this may be considered as a drawback. At the other hand, it prevents the accumulation of rounding errors from step to step.

Finally, it must be noticed that for high values of  $n, m$  or  $d$ , matrices  $\mathbf{R}_d$  and  $\mathbf{L}_d$  may have large dimensions and unavoidable rounding errors may affect triangularizations (10) and (17). In that case, the notion of a shape of a polynomial matrix should preferably be replaced with the more realistic notion of an  $\varepsilon$ -shape, defined below.

**Definition 4** *The  $\varepsilon$ -shape of a full column rank  $n$ -by- $m$  matrix  $A$  is the series  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  of positive integers  $\sigma_i = 1, 2, \dots, n$  such that  $\text{ent}_{\sigma_i, i} A$  is the uppermost non-zero entry in  $\text{col}_i A$  of norm greater than  $\varepsilon$ .*

Step 3 of Algorithms 1, 1', 2 and 2' must be modified accordingly. The value of  $\varepsilon$  should depend on  $n, m, d$  and the floating point relative accuracy of the computer.

## 4 Triangular Form over $\mathbb{R}[\mathbf{s}]$

The algorithms described in the preceding section were designed to compute a reduction matrix  $U(\mathbf{s})$  invertible over  $\mathbb{R}(\mathbf{s})$ , but not necessarily unimodular. In the sequel it is explained how to extend the Sylvester matrix and interpolation triangularization algorithms over  $\mathbb{R}(\mathbf{s})$  so as to derive a unimodular reduction matrix. The instrumental key concept that will be introduced is polynomial degree minimization over ideals.

### 4.1 Sylvester matrix approach

Let  $\mathcal{E}^j \subset \mathbb{R}^n[\mathbf{s}]$  denote the submodule generated by  $\{e_j, \dots, e_n\}$ . Let the map  $p^j(\cdot) : \mathbb{R}^n[\mathbf{s}] \rightarrow \mathbb{R}[\mathbf{s}]$  extract the  $j$ -th component of a polynomial module and define the submodule  $\mathcal{A}^j = p^j(\mathcal{A} \cap \mathcal{E}^j)$ . Note that  $\mathcal{A}^j$  is an ideal over  $\mathbb{R}[\mathbf{s}]$ . The following theorem algebraically characterizes triangular forms over  $\mathbb{R}(\mathbf{s})$  and  $\mathbb{R}[\mathbf{s}]$ , as defined in Theorem 1.

**Theorem 4** *Given a polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$ , let  $T(\mathbf{s})$  stand for a matrix of triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  such that  $\text{col}_i T(\mathbf{s}) \in \mathcal{A} \cap \mathcal{E}^{\sigma_i}$ . Then  $T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ . Moreover, if each leading entry  $\text{ent}_{\sigma_i, i} T(\mathbf{s})$  has minimal degree in the ideal  $\mathcal{A}^{\sigma_i}$  for  $i = 1, \dots, m$ , then  $T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ .*

**Proof :** See [15, Proposition 1]. □

The above theorem states that the triangular form of a polynomial matrix  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$  is actually a special triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  with the additional requirement that leading entry degrees must be minimal. Clearly, the triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  as computed by Algorithm 1 does not necessarily verify these constraints. The sequel aims at proving that Algorithm 1 can be extended for that purpose.

Suppose that a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  of shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$  has been found for a given value of  $d$ . Define the submodule  $\mathcal{J}_d^{\sigma_i} = p^{\sigma_i}(\mathcal{R}_d \cap \mathcal{E}^{\sigma_i}) \subset \mathcal{A}^{\sigma_i}$ . One can check that  $\mathcal{J}_d^{\sigma_i} = \{0\}$  as soon as  $\sigma_i$  does not belong to  $\Sigma$ . According to Theorem 4, in order to derive a triangular form over  $\mathbb{R}[\mathbf{s}]$  it is necessary to find the minimal degree polynomials in ideals  $\mathcal{J}_d^{\sigma_i} \mathbb{R}[\mathbf{s}]$ .

**Lemma 4** *Given a full column rank polynomial matrix  $A(\mathbf{s})$  of degree  $d_A$ , suppose that triangularization Algorithm 1 resulted in a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$  with a reduction matrix of degree  $d_U$ , and let  $d = d_A + d_U$ . Then, for any  $i = 1, \dots, m$  the minimal degree polynomial of ideal  $\mathcal{J}_d^{\sigma_i} \mathbb{R}[\mathbf{s}]$  belongs to submodule  $\mathcal{J}_d^{\sigma_i}$ .*

Before the exposition of the proof of Lemma 4, the following fundamental result [15, §3.3] is recalled.

**Lemma 5** *Let  $\mathcal{H}_0$  be a submodule in  $\mathbb{R}[\mathbf{s}]_d$  and let the iteration  $\mathcal{H}_{k+1} = \mathbf{s}\mathcal{H}_k + \mathcal{H}_k$  define a submodule sequence in  $\mathbb{R}[\mathbf{s}]$  for  $k \geq 0$ . For a given  $k$ , if  $\dim \mathcal{H}_{k+1} = \dim \mathcal{H}_k + 1$  then the minimum degree polynomial of ideal  $\mathcal{H}_0 \mathbb{R}[\mathbf{s}]$  belongs to submodule  $\mathcal{H}_k$ .*

**Proof of Lemma 5 :** For any  $k \geq 0$ ,  $\mathcal{H}_{k+1}$  contains  $\mathcal{H}_k$  and at least one polynomial of higher degree than any polynomial of  $\mathcal{H}_k$ , whence  $\dim \mathcal{H}_{k+1} \geq \dim \mathcal{H}_k + 1$ . If  $\dim \mathcal{H}_{k+1} = \dim \mathcal{H}_k + 1$  holds for a given value of  $k$ , then the equality also holds for higher values of  $k$ , thus showing that a minimal degree polynomial was already found in  $\mathcal{H}_k$ .  $\square$

**Proof of Lemma 4 :** By construction  $\mathcal{R}_{d+1} = \mathbf{s}\mathcal{R}_d + \mathcal{R}_d$  and therefore  $\mathcal{J}_{d+1}^{\sigma_i} \subset \mathbf{s}\mathcal{J}_d^{\sigma_i} + \mathcal{J}_d^{\sigma_i}$  for  $i = 1, \dots, m$  and any  $d \geq d_A$ . Triangularization of  $\mathbf{R}_d$  resulted in a triangular shape  $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ , so  $\dim \mathcal{R}_d = \sum_{i=1}^m \dim \mathcal{J}_d^{\sigma_i} = \text{rank } \mathbf{R}_d$ . Since  $A(\mathbf{s})$  has full column rank  $m$ ,  $\text{rank } \mathbf{R}_{d+1} = \text{rank } \mathbf{R}_d + m$ , hence  $\dim \mathcal{R}_{d+1} = \sum_{i=1}^m \dim \mathcal{J}_{d+1}^{\sigma_i} = \sum_{i=1}^m \dim \mathcal{J}_d^{\sigma_i} + m$  and  $\dim \mathcal{J}_{d+1}^{\sigma_i} = \dim \mathcal{J}_d^{\sigma_i} + 1$ . According to Lemma 5, the minimal degree polynomial of ideal  $\mathcal{J}_d^{\sigma_i} \mathbb{R}[\mathbf{s}]$  can be found in submodule  $\mathcal{J}_d^{\sigma_i}$ .  $\square$

Lemma 4 provides the theoretical guarantee that one can extract minimal degree polynomials by working directly on matrix  $\mathbf{R}_d$ . Let the sets  $\mathcal{C}_i$  be as in (11). Define  $\mathbf{R}_d^{\sigma_i} = \{\text{col}_j \mathbf{R}_d \cap \text{row}_k \mathbf{R}_d \text{ for all } j \in \mathcal{C}_{\sigma_i} \text{ and all } k \in [\sigma_i(d+1) - d, \sigma_i(d+1)]\}$ . Submodule  $\mathcal{J}_d^{\sigma_i}$  is therefore generated by matrix  $\mathbf{R}_d^{\sigma_i}$  in the basis  $\{\mathbf{s}^d, \dots, \mathbf{s}, 1\}$ . Since  $\mathbf{R}_d$  is triangular, so is  $\mathbf{R}_d^{\sigma_i}$ . As a result, the rightmost column of matrix  $\mathbf{R}_d^{\sigma_i}$ , i.e. the column indexed by the maximal element in  $\mathcal{C}_i$ , is a representation of the minimal degree polynomial in  $\mathcal{J}_d^{\sigma_i}$ . The following theorem has been shown.

**Theorem 5** *Triangularizing over  $\mathbb{R}[\mathbf{s}]$  a full column rank polynomial matrix  $A(\mathbf{s})$  amounts to triangularizing over  $\mathbb{R}$  the row permuted Sylvester matrix  $\mathbf{R}_d$ .*

Finally, the following corollary to Theorem 5 underlines the fact that the triangularization techniques presented here can also be used for greatest common divisor extraction.

**Corollary 2** *Let  $A(\mathbf{s}) = [B_1(\mathbf{s}) \cdots B_p(\mathbf{s})]$ . Extracting the greatest common left divisor of matrices  $B_1(\mathbf{s}), \dots, B_p(\mathbf{s})$  amounts to triangularizing over  $\mathbb{R}$  the row permuted Sylvester matrix  $\mathbf{R}_d$ .*

**Proof :** See Theorem 1 and, by example, Lemma 6.3-3 in [9]. □

### Algorithm 3 : Triangularization over $\mathbb{R}[\mathbf{s}]$ Sylvester matrix approach

*Input* A full column rank polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$  of degree  $d_A$ .  
*Output* A unimodular reduction matrix  $U(\mathbf{s}) \in \mathbb{R}^{m \times m}[\mathbf{s}]$  such that  $A(\mathbf{s})U(\mathbf{s}) = T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ .

Only the following step differs from Algorithm 1.

*Step 4* Extract matrix  $\bar{U}$ , such that  $\text{col}_i \bar{U} = \text{col}_{\hat{k}_i} \mathbf{U}$ , where  $\hat{k}_i$  is the maximal element in  $\mathcal{C}_{\sigma_i}$ ,  $i = 1, \dots, m$ .

### Algorithm 3' :Triangularization over $\mathbb{R}[\mathbf{s}]$ Iterative Sylvester matrix approach

Only steps 0 and 3, taken from Algorithm 1', differ from Algorithm 3

## 4.2 Interpolation Approach

Since interpolated matrix  $\mathbf{L}_d$  and permuted Sylvester matrix  $\mathbf{R}_d$  are equivalent up to a change of basis, the interpolation approach to triangularization over  $\mathbb{R}[\mathbf{s}]$  naturally follows.

**Lemma 6** *Given interpolated values  $A(\mathbf{s}_0), A(\mathbf{s}_1), \dots, A(\mathbf{s}_d)$  of a full column rank polynomial matrix  $A(\mathbf{s})$ , suppose that triangularization Algorithm 2 resulted in a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ . Then, for any  $i = 1, \dots, m$  the minimal degree polynomial of ideal  $\mathcal{J}_d^{\sigma_i} \mathbb{R}[\mathbf{s}]$  belongs to submodule  $\mathcal{J}_d^{\sigma_i}$ .*



**Proof** : Similar to that of Lemma 4. □

Lemma 6 provides the guarantee that it is of no use to work with a higher number of interpolation points once a triangular form over  $\mathbb{R}(\mathbf{s})$  is obtained. Therefore one can extract the minimal degree polynomial in  $\mathcal{J}_d^{\sigma_i}$  by working directly on the matrix  $\mathbf{T}$  derived by triangularization of  $\mathbf{L}_d$ . Let the sets  $\mathcal{C}_{\sigma_i}$  be defined as in (11). Define  $\mathbf{L}_d^{\sigma_i} = \{\text{col}_j \mathbf{L}_d \cap \text{row}_k \mathbf{L}_d \text{ for all } j \in \mathcal{C}_{\sigma_i} \text{ and all } k \in [\sigma_i(d+1) - d, \sigma_i(d+1)]\}$ . Submodule  $\mathcal{J}_d^{\sigma_i}$  is generated by matrix  $\mathbf{L}_d^{\sigma_i}$  in the basis  $\{\mathbf{s}^d, \dots, \mathbf{s}, 1\} M_d^{-1}$ . Thus an easy way to minimize the degree of a polynomial in  $\mathcal{J}_d^{\sigma_i}$  could be to triangularize matrix  $M_d^{-1} \mathbf{L}_d^{\sigma_i}$ . The minimal degree polynomial then corresponds to the rightmost column of the computed triangular form. However, in doing so one does not take advantage of the special structure of interpolated matrix  $\mathbf{L}_d^{\sigma_i}$ . Moreover, Vandermonde matrix  $M_d$  may be ill-conditioned. Therefore, the use of a special Vandermonde linear system solver is recommended. Only a simplified version of the two-step algorithm exposed in [6] is required for degree minimization. It is described in the sequel.

For a given column index  $i$ , suppose that one aims at extracting the minimal degree polynomial in the submodule generated by a matrix  $\mathbf{I} = \mathbf{L}_d^{\sigma_i}$  in the basis  $\{\mathbf{s}^d, \dots, \mathbf{s}, 1\} M_d^{-1}$ . Entries of  $\mathbf{I}$  are denoted as follows

$$\mathbf{I} = \begin{bmatrix} t_1(\mathbf{s}_d) & \cdots & t_p(\mathbf{s}_d) \\ \vdots & & \vdots \\ t_1(\mathbf{s}_1) & \cdots & t_p(\mathbf{s}_1) \\ t_1(\mathbf{s}_0) & \cdots & t_p(\mathbf{s}_0) \end{bmatrix} \in \mathbb{R}^{(d+1) \times p}.$$

When  $j = 1, \dots, p$ , each column of  $\mathbf{I}$

$$\text{col}_j \mathbf{I} = \begin{bmatrix} t_j(\mathbf{s}_d) \\ \vdots \\ t_j(\mathbf{s}_1) \\ t_j(\mathbf{s}_0) \end{bmatrix} = M_d \begin{bmatrix} t_{d,j} \\ \vdots \\ t_{1,j} \\ t_{0,j} \end{bmatrix}$$

corresponds to a polynomial  $t_j(\mathbf{s}) = t_{0,j} + \mathbf{s} t_{1,j} + \dots + \mathbf{s}^d t_{d,j}$  which Newton representation reads

$$t_j(\mathbf{s}) = \sum_{k=0}^d c_{k,j} \prod_{l=0}^{k-1} (\mathbf{s} - \mathbf{s}_l).$$

Newton coefficients  $c_{k,j}$  are divided differences and may be generated in an iterative way from entries of matrix  $\mathbf{I}$  [6] to yield the Newton coefficient matrix

$$\mathbf{C} = \begin{bmatrix} c_{d,1} & \cdots & c_{d,p} \\ \vdots & & \vdots \\ c_{1,1} & \cdots & c_{1,p} \\ c_{0,1} & \cdots & c_{0,p} \end{bmatrix}.$$

Triangularization of  $\mathbf{C}$  reads

$$\mathbf{F} = \mathbf{C}\mathbf{V} = \begin{bmatrix} f_{d,1} & 0 & \cdots & 0 \\ f_{d-1,1} & f_{d-1,2} & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ f_{q,1} & f_{q,2} & & f_{q,p} \\ \vdots & \vdots & & \vdots \\ f_{1,1} & f_{1,2} & & f_{1,p} \\ f_{0,1} & f_{0,2} & \cdots & f_{0,p} \end{bmatrix}. \quad (18)$$

The last column of  $\mathbf{F}$  thus corresponds to the polynomial

$$t_{\min}(\mathbf{s}) = \sum_{k=0}^q f_{k,p} \prod_{l=0}^{k-1} (\mathbf{s} - \mathbf{s}_l)$$

featuring the greatest number of zero leading coefficients, that is to say, to the expected minimal degree polynomial. The procedure is as follows.

### Procedure 1 : Degree minimization over interpolated polynomial submodule

*Input* Interpolation points  $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_d$  and a full column rank matrix  $\mathbf{I}$  whose  $p$  columns generate a submodule of  $\mathbb{R}[\mathbf{s}]_d$  in the basis  $\{\mathbf{s}^d, \dots, \mathbf{s}, 1\}M_d^{-1}$ .

*Output* The vector  $v$  such that  $\mathbf{I}v$  is a representation w.r.t. the same basis of the minimal degree polynomial in this submodule.

*Step 1* Build Newton matrix  $\mathbf{C}$  using the iteration

$$\begin{aligned} \mathbf{C} &= \mathbf{I} \\ \text{for } k &= 1, \dots, d \\ \text{for } i &= 1, \dots, d+1-k \\ \text{row}_i \mathbf{C} &= (\text{row}_i \mathbf{C} - \text{row}_{i+1} \mathbf{C}) / (\mathbf{s}_{d+1-i} - \mathbf{s}_{d+1-i-k}). \end{aligned}$$

*Step 2* Triangularize matrix  $\mathbf{C}$  over  $\mathbb{R}$  as in (18). Vector  $v$  is the rightmost column in  $\mathbf{V}$ .

If Procedure 1 is successively applied to matrices  $\mathbf{L}_d^{\sigma_i}$  corresponding to submodules  $\mathcal{J}_d^{\sigma_i}$  for  $i = 1, \dots, m$ , one gets a reduction matrix  $U(\mathbf{s})$  such that degrees of polynomials  $\text{ent}_{\sigma_i, i} A(\mathbf{s})U(\mathbf{s})$  are minimized. According to Theorem 4,  $U(\mathbf{s})$  triangularizes  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ . The following theorem has been shown.

**Theorem 6** *Triangularizing over  $\mathbb{R}[\mathbf{s}]$  a full column rank polynomial matrix  $A(\mathbf{s})$  amounts to triangularizing over  $\mathbb{R}$  the interpolated matrix  $\mathbf{L}_d$ , followed by  $m$  degree minimizations over interpolated polynomial submodules as described in Procedure 1, where  $m$  is the number of columns of  $A(\mathbf{s})$ .*

**Algorithm 4 : Triangularization over  $\mathbb{R}[\mathbf{s}]$   
Interpolation approach**

*Input* Values  $A(\mathbf{s}_i) \in \mathbb{R}^{n \times m}$  of a full column rank polynomial matrix  $A(\mathbf{s}) \in \mathbb{R}^{n \times m}[\mathbf{s}]$ , interpolated at distinct points  $\mathbf{s}_i \in \mathbb{R}$ ,  $i = 0, 1, 2, \dots$

*Output* A unimodular reduction matrix  $U(\mathbf{s}) \in \mathbb{R}^{m \times m}[\mathbf{s}]$  such that  $A(\mathbf{s})U(\mathbf{s}) = T(\mathbf{s})$  is a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ .

Only Step 4 differs from Algorithm 2

*Step 4.a* Extract sub-matrices  $\mathbf{L}_d^{\sigma_i} = \{\text{col}_j \mathbf{L}_d \cap \text{row}_k \mathbf{L}_d, \text{ for all } j \in \mathcal{C}_{\sigma_i} \text{ and all } k \in [\sigma_i(d+1) - d, \sigma_i(d+1)]\}$ . Procedure 1 with inputs  $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_d$  and  $\mathbf{L}_d^{\sigma_i}$  generates outputs  $v_i$  for  $i = 1, \dots, m$ .

*Step 4.b* Extract sub-matrices  $\mathbf{U}^{\sigma_i} = \{\text{col}_j \mathbf{U}^{\sigma_i} \text{ for all } j \in \mathcal{C}_{\sigma_i}\}$ . Build matrix  $\bar{U}$  such that  $\text{col}_i \bar{U} = \mathbf{U}^{\sigma_i} v_i$  for  $i = 1, \dots, m$ .

**Algorithm 4' :Triangularization over  $\mathbb{R}[\mathbf{s}]$   
Iterative Interpolation approach**

See Algorithm 4, except for Steps 0 and 3, see Algorithm 2'.

### 4.3 Numerical Considerations

Obviously, the derivation of Algorithm 3 from Algorithm 1 does not introduce any potential numerical instability. Things are quite different in Algorithm 4 since the column selection scheme in Step 4 hinges on Procedure 1. Hopefully, the computation of Newton coefficients in Step 1 of Procedure 1 generally produces surprisingly accurate solutions even when Vandermonde matrix  $M_d$  is ill-conditioned – see [6] or [8, §21.3]. Furthermore, the triangularization in Step 2 of Procedure 1 can be performed via powerful numerical methods such as column Householder transformations [6].

The column selection schemes in Step 4 of Algorithms 3 and 4 ensure that the reduction matrix  $U(\mathbf{s})$  is unimodular. In [3], the authors underlined the fact that their Gaussian elimination algorithm, in order to exploit the special structure of the Sylvester matrix, was designed to preserve unimodularity to the detriment of numerical stability. By contrast, the procedures described here hinge on orthogonal matrix operations while preserving unimodularity of the reduction matrix. One can naturally argue that the procedures do not take advantage of the special structure of the Sylvester or interpolated matrices, with the notable exception of Procedure 1. This seems to be the price one has to pay for numerical reliability, but does not appear to be a fundamental limitation in view of the availability of powerful computer tools.

Finally, if one is interested in getting the Hermite form defined in Theorem 1 rather than an arbitrary triangular form over  $\mathbb{R}[\mathbf{s}]$ , this can straightforwardly be done using elementary

column operations. Each leading entry must be made monic and entries at the left of it must be reduced via the Euclidean algorithm. Naturally, this may be achieved at the expense of numerical reliability. However, the number of potentially unstable operations needed for computing the Hermite form of a triangular form over  $\mathbb{R}[\mathbf{s}]$  is certainly far less than the number of operations needed for computing the Hermite form of a polynomial matrix in arbitrary form.

## 5 Numerical Example

Both Sylvester matrix and interpolation approaches are now illustrated on a very simple example: reduction to lower left triangular form of polynomial matrix

$$A(\mathbf{s}) = \begin{bmatrix} \mathbf{s} - 1 & \mathbf{s}^2 - 1 \\ 2 & 2\mathbf{s} + 2 \\ 0 & 3 \end{bmatrix}.$$

### 5.1 Sylvester matrix approach

First, Algorithm 1' will be used to derive a minimal degree reduction matrix. Let  $d_U = 0$ ,  $d = d_A + d_U = 2$  and built  $\mathbf{R}_2$  as in (6). Triangularization (10) reads

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & -1 \\ \hline 0 & 0 \\ 0 & 2 \\ 2 & 2 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 3 \end{bmatrix}}_{\mathbf{R}_2} \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathbf{U}} = \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & -1 \\ \hline 0 & 0 \\ -2 & 0 \\ -2 & 2 \\ \hline 0 & 0 \\ 0 & 0 \\ -3 & 0 \end{bmatrix}}_{\mathbf{T}}$$

Triangular matrix  $\mathbf{T}$  has shape  $\bar{\Sigma} = \{\bar{\sigma}_1, \bar{\sigma}_2\} = \{1, 2\}$ . The sets  $\mathcal{C}_i$  defined in (11) are  $\mathcal{C}_1 = \{1, 2\}$ ,  $\mathcal{C}_2 = \emptyset$  and  $\mathcal{C}_3 = \emptyset$ . Since  $\mathcal{C}_2$  and  $\mathcal{C}_3$  are empty, no reduction matrix of degree

$d_U = 0$  has been found. Letting  $d_U = d_U + 1$ , next triangularization (10) reads

$$\underbrace{\left[ \begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{array} \right]}_{R_3} \underbrace{\left[ \begin{array}{ccc|c} 0 & -0.707 & 0.408 & -0.577 \\ -1 & 0 & 0 & 0 \\ \hline 0 & 0 & -0.817 & -0.577 \\ 0 & -0.707 & -0.408 & 0.577 \end{array} \right]}_U = \underbrace{\left[ \begin{array}{ccc|c} -1 & 0 & 0 & 0 \\ 0 & -1.414 & 0 & 0 \\ 1 & 0.707 & -1.225 & 0 \\ 0 & 0.707 & 1.225 & 0 \\ \hline 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -2 & -2.828 & 0 & 0 \\ 0 & -1.414 & -2.450 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 0 & -2.121 & -1.225 & 1.732 \end{array} \right]}_T.$$

Now, the shape of triangular matrix  $\mathbf{T}$  is  $\bar{\Sigma} = \{1, 2, 3, 12\}$ . Column index sets are  $\mathcal{C}_1 = \{1, 2, 3\}$ ,  $\mathcal{C}_2 = \emptyset$  and  $\mathcal{C}_3 = \{12\}$  whence triangular shape  $\Sigma = \{\sigma_1, \sigma_2\} = \{1, 3\}$ . A possible choice for matrix  $\bar{U}$  is for example  $\bar{U} = [\text{col}_1 \mathbf{U} \text{ col}_4 \mathbf{U}]$ . The corresponding polynomial matrix is

$$U(\mathbf{s}) = \begin{bmatrix} 0 & -0.577\mathbf{s} - 0.577 \\ -\mathbf{s} & 0.577 \end{bmatrix}.$$

One can check that  $U(\mathbf{s})$  is not unimodular, so that

$$T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s}) = \begin{bmatrix} -\mathbf{s}^3 + \mathbf{s} & 0 \\ -2\mathbf{s}^2 - 2\mathbf{s} & 0 \\ -3\mathbf{s} & 1.732 \end{bmatrix}$$

is only a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}(\mathbf{s})$ . Now suppose that one is seeking a triangular form over  $\mathbb{R}[\mathbf{s}]$ . From Algorithm 3',  $\hat{k}_1 = \max\{k \in \mathcal{C}_1\} = 3$  and  $\hat{k}_2 = \max\{k \in \mathcal{C}_3\} = 12$  so that matrix  $\bar{U} = [\text{col}_3 \mathbf{U} \text{ col}_4 \mathbf{U}]$  yields a unimodular matrix

$$U(\mathbf{s}) = \begin{bmatrix} -0.817 + 0.408\mathbf{s} & -0.577\mathbf{s} - 0.577 \\ -0.408 & 0.577 \end{bmatrix}$$

and a triangular form

$$T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s}) = \begin{bmatrix} -1.225\mathbf{s} + 1.225 & 0 \\ -2.450 & 0 \\ -1.225 & 1.732 \end{bmatrix}$$

of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ .

## 5.2 Interpolation Approach

Now assume that polynomial matrix  $A(\mathbf{s})$  is only given through interpolated values  $A(\mathbf{s}_0), \dots, A(\mathbf{s}_d)$  and that Algorithm 4 is used to compute a reduction matrix over  $\mathbb{R}[\mathbf{s}]$ .

Therefore, let  $d = \hat{d} = 4$  as in (12). By example, choose interpolation points  $\mathbf{s}_0 = -2, \mathbf{s}_1 = -1, \mathbf{s}_2 = 0, \mathbf{s}_3 = 1, \mathbf{s}_4 = 2$ . Algorithm 4 directly leads to triangularization

$$\left[ \begin{array}{cc|cc|cc} -12 & 12 & 6 & -6 & -3 & 3 \\ -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 12 & 2 & 6 & 1 & 3 \\ \hline 8 & -8 & -4 & 4 & 2 & -2 \\ 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 4 & 2 & 4 & 2 & 4 \\ 8 & 24 & 4 & 12 & 2 & 6 \\ \hline 0 & 12 & 0 & -6 & 0 & 3 \\ 0 & 3 & 0 & -3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 0 & 12 & 0 & 6 & 0 & 3 \end{array} \right] \left[ \begin{array}{cccc|cc} -0.617 & -0.246 & 0.151 & -0.401 & 0.568 & -0.229 \\ 0.617 & -0.492 & 0.302 & -0.535 & 0 & 0 \\ \hline 0.309 & 0.492 & -0.302 & -0.267 & 0.162 & -0.688 \\ -0.309 & 0.246 & -0.151 & -0.668 & -0.568 & 0.229 \\ \hline -0.154 & -0.616 & -0.452 & 0.134 & -0.406 & -0.459 \\ 0.154 & -0.123 & -0.754 & -0.134 & 0.406 & 0.459 \end{array} \right] =$$

$$\mathbf{L}_4 \underbrace{[\mathbf{U}^{\sigma_1} \mid \mathbf{U}^{\sigma_2}]}_{\mathbf{U}} =$$

$$\left[ \begin{array}{cccc|cc} 19.44 & 0 & 0 & 0 & 0 & 0 \\ 2.160 & 2.708 & 0 & 0 & 0 & 0 \\ 0 & 0.739 & 1.206 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 4.012 & -5.416 & 0 & -12.83 & 0 & 0 \\ \hline -12.96 & 0 & 0 & 0 & 0 & 0 \\ -2.160 & -2.708 & 0 & 0 & 0 & 0 \\ 0 & -1.477 & -2.412 & 0 & 0 & 0 \\ 0.926 & -2.216 & -3.618 & -6.414 & 0 & 0 \\ 8.024 & -10.83 & 0 & -25.66 & 0 & 0 \\ \hline 9.721 & -7.755 & 2.261 & -2.806 & 4.623 & 0 \\ 3.240 & -2.585 & -0.905 & 0 & 2.920 & 0.688 \\ 0.463 & -0.369 & -2.261 & -0.401 & 1.217 & 1.377 \\ 1.389 & -1.108 & -1.809 & -4.009 & -0.487 & 2.065 \\ 6.018 & -4.801 & 0.452 & -10.82 & -2.190 & 2.753 \end{array} \right] = \mathbf{T}.$$

Matrix  $\mathbf{T}$  has shape  $\bar{\Sigma} = \{1, 2, 3, 5, 11, 12\}$  hence  $\mathcal{C}_1 = \{1, 2, 3, 5\}$ ,  $\mathcal{C}_2 = \emptyset$ ,  $\mathcal{C}_3 = \{11, 12\}$  and  $\Sigma = \{\sigma_1, \sigma_2\} = \{1, 3\}$ . In order to get a triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$ , two degree minimizations must be performed on matrices

$$\mathbf{L}^{\sigma_1} = \begin{bmatrix} 19.44 & 0 & 0 & 0 \\ 2.160 & 2.708 & 0 & 0 \\ 0 & 0.739 & 1.206 & 0 \\ 0 & 0 & 0 & 0 \\ 4.012 & -5.416 & 0 & -12.83 \end{bmatrix} \text{ and } \mathbf{L}^{\sigma_2} = \begin{bmatrix} 4.623 & 0 \\ 2.920 & 0.688 \\ 1.217 & 1.377 \\ -0.487 & 2.065 \\ -2.190 & 2.753 \end{bmatrix}.$$

Using Procedure 1, one gets  $v_1 = [-0.195 \ -0.779 \ -0.572 \ 0.169]'$  and  $v_2 = [0.375 \ 0.927]'$ . Finally, Algorithm 3 yields

$$\bar{U} = [U^{\sigma_1} v_1 \ U^{\sigma_2} v_2] = \begin{bmatrix} 0.158 & 0 \\ 0 & 0 \\ \hline -0.316 & -0.577 \\ -0.158 & 0 \\ \hline 0.791 & -0.577 \\ 0.474 & 0.577 \end{bmatrix}$$

and the unimodular reduction matrix

$$U(\mathbf{s}) = \begin{bmatrix} 0.158\mathbf{s}^2 - 0.316\mathbf{s} + 0.791 & -0.577\mathbf{s} - 0.577 \\ -0.158\mathbf{s} + 0.474 & 0.577 \end{bmatrix}.$$

The corresponding triangular form of  $A(\mathbf{s})$  over  $\mathbb{R}[\mathbf{s}]$  is

$$T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s}) = \begin{bmatrix} 1.265\mathbf{s} - 1.265 & 0 \\ 2.530 & 0 \\ -0.474\mathbf{s} + 1.423 & 1.732 \end{bmatrix}.$$

Note that the reduction matrix obtained here has degree two, whereas a unimodular reduction matrix of degree one has been found using iterative Algorithm 3' in Section 5.1.

## 6 Comparison with other triangularization methods

Numerical tests have been carried out to compare the performances of our algorithms with other triangularization methods. Only triangularization over  $\mathbb{R}[\mathbf{s}]$  has been considered. In addition to the algorithms presented in this paper, namely

**Algorithm A** – Sylvester matrix approach (Algorithm 3)

**Algorithm B** – Interpolation approach (Algorithm 4)

the other triangularization algorithms that were tested are

**Algorithm C** – Labhalla's Gaussian elimination algorithm [15, Algorithm HERMIPOL]

**Algorithm D** – Standard elementary polynomial operations (see Theorem 1)

**Algorithm E** – MAPLE<sup>2</sup> built-in symbolic routine `linalg[hermite]`.

Some remarks concerning implementation are listed below.

---

<sup>2</sup>MAPLE is a registered trademark of Waterloo Maple Software.

- The routines have been tested on square polynomial matrices  $A(\mathbf{s})$  of dimension  $n$ , degree  $d$  and random integer coefficients with absolute value strictly less than 10.
- The minimal possible degree of a matrix  $U(\mathbf{s})$  that triangularizes a matrix  $A(\mathbf{s})$  as above is generically equal to the upper bound  $\hat{d}_U$  given in (3). For this reason, tests were not conducted on iterative Algorithms 3' and 4'. Their execution times would necessarily be greater than those of Algorithms 3 and 4, respectively. Note however that this may not be the case for triangularization problems arising from practical control issues. See also the example in Section 5.
- At the exception of Algorithm E, the algorithms have been implemented in MATLAB 5.1 with floating point accuracy  $2.22 \cdot 10^{-16}$  on a Sun SparcStation 4.
- Algorithms A and B rely upon a triangularization over  $\mathbb{R}$  that is performed by column Householder transformations. They are entirely written in interpreted MATLAB code and no external routine call is made. In particular, we did not use the MATLAB built-in routine `qr`.
- Algorithm C relies upon a triangularization over  $\mathbb{R}$  that is performed by successive applications of a modified Gaussian elimination with only partial pivoting, as described in [15]. It is also entirely written in interpreted MATLAB code.
- Interpolation points in Algorithm B are equispaced within the real interval  $[-1, +1]$ .
- The value  $\varepsilon = n \cdot d \cdot 10^{-10}$  is used to extract the  $\varepsilon$ -shape (see Definition 4) in Algorithms A and B.
- The criterion that has been used to decide whether the triangularization has properly been achieved is twofold.

1. All coefficients in the strictly upper triangular part of the triangular form  $T(\mathbf{s}) = A(\mathbf{s})U(\mathbf{s})$  must have absolute value less than  $10^{-8}$ .
2. The inequalities

$$0.9 < \frac{|\det U(\mathbf{s})|}{|\det U(0)|} < 1.1 \quad (19)$$

must hold for any  $\mathbf{s} \in [-1, +1]$ . This is aimed at testing unimodularity of  $U(\mathbf{s})$  without computing  $\det U(\mathbf{s})$ , a numerically difficult task for high values of  $n$  and  $d$ .

Average execution times in seconds are reported in Table 1. A star ( $\star$ ) means that the triangularization failed because the above criteria were not verified. A double star ( $\star\star$ ) means that the execution time exceeded 1500 seconds.



d	n	Algorithm				
		A	B	C	D	E
3	3	0.21	1.3	0.62	16	0.65
5	3	0.51	2.4	1.1	26	2.7
7	3	0.87	4.0	1.6	39	12
9	3	1.5	6.1	2.3	54	30
3	4	0.46	2.4	1.1	29	8.8
5	4	2.0	5.6	2.5	59	111
7	4	4.1	10	3.7	*	603
9	4	7.8	*	5.5	*	**
3	5	2.1	4.6	2.3	*	109
5	5	7.3	12	4.6	*	**
7	5	14.4	*	8.0	*	**
9	5	29	*	12	*	**
3	9	48	59	17	*	**
5	9	220	*	*	*	**
7	9	580	*	*	*	**
9	9	1200	*	*	*	**

Table 1 : Comparative execution times in seconds.

Some comments are in order.

- As expected, Algorithm D is the less reliable triangularization method. Even for relatively low values of  $d$  and  $n$ , Algorithm D returns a result that is generally extremely far from the correct result.
- Algorithm E is based on symbolic computation. This approach is known to be subject to intermediate expression swell and turns out to be impractical even for medium-sized problems [19, Chapter 3].
- Algorithm B behaves badly for relatively high values of  $d$  and  $n$ . This is due to an ill-conditioning of the interpolated matrix, in turn due to the choice of interpolation points. For a discussion on that matter, see Section 3.3.
- For low values of  $n$ , Algorithm A is faster than Algorithm C. The tendency is reversed when  $n$  is higher. In the latter case, Algorithm A seems to be more reliable than Algorithm C, to the expense of computational load.

## 7 Conclusion

Numerical procedures were proposed for polynomial matrix triangularization. Two algorithms were first described for triangularization over the field of polynomial fractions.

They may also be used for polynomial rank evaluation or polynomial null-space extraction. The first algorithm is based on Sylvester matrices whereas the second algorithm makes extensive use of interpolation theory. Once combined with polynomial degree minimization, they gave rise to two other algorithms for triangularization over the ring of polynomials that can alternatively be used for greatest common divisor computation.

Contrary to previously available algorithms, the procedures presented here only rely upon numerically reliable and well-worked linear algebra computer tools. From the point of view of numerical stability, the advantage of using these new polynomial triangularization techniques over using the standard Euclidean division algorithm may be compared to the advantage of using orthogonal reductions over using the standard Gaussian elimination algorithm.

Some aspects of this work need further investigation. As mentioned above, the proposed procedures do not exploit the special structure of Sylvester or interpolated matrices. One may wonder whether this could be done in a numerically stable way, so as to further reduce the computational load, both in terms of operation count and memory requirement. Another point that deserves attention is the use of complex interpolation points, and more generally the possibility of triangularizing complex polynomial matrices. Finally, an interesting question is whether the same kind of numerical method could be used for getting the Smith form or some less restrictive but equivalent diagonal form.

## Acknowledgment

We are particularly grateful to the reviewers for their thorough work and constructive comments.

## References

- [1] P. J. Antsaklis and Z. Gao “Polynomial and Rational Matrix Interpolation : Theory and Control Applications”, *International Journal of Control*, Vol. 58, No. 2, pp. 349–404, 1993.
- [2] T. G. J. Beelen and G. W. Veltkamp “Numerical Computation of a Coprime Factorization of a Transfer Function Matrix”, *Systems and Control Letters*, Vol. 9, pp. 281–288, 1987.
- [3] R. R. Bitmead, S. Y. Kung, B. D. O. Anderson and T. Kailath “Greatest Common Divisor via Generalized Sylvester and Bézout Matrices”, *IEEE Transactions on Automatic Control*, Vol. 23, No. 6, pp. 1043–1047, 1978.
- [4] F. M. Callier and C. A. Desoer “Multivariable Feedback Systems”, *Springer Verlag*, 1982.

- [5] C. T. Chen “Linear System Theory and Design”, *CBS College Publishing*, 1984.
- [6] G. H. Golub and C. F. Van Loan “Matrix Computations - Second Edition”, *The Johns Hopkins University Press*, 1989.
- [7] D. Henrion and M. Šebek “Symmetric Matrix Polynomial Equation : Interpolation Results”, to appear in *Automatica*, Vol. 34, 1998.
- [8] N. J. Higham “Accuracy and Stability of Numerical Algorithms”, *SIAM*, 1996.
- [9] T. Kailath “Linear Systems”, *Prentice Hall*, 1980.
- [10] E. Kaltofen, M. S. Krishnamoorthy and B. D. Saunders “Fast Parallel Computations of Hermite and Smith Forms of Polynomial Matrices”, *SIAM Journal of Algebraic and Discrete Methods*, Vol. 8, pp. 683–690, 1987.
- [11] V. Kučera “Analysis and Design of Discrete Linear Control Systems”, *Prentice Hall International Series in Systems and Control Engineering*, 1991.
- [12] V. Kučera “Diophantine Equations in Control – A Survey”, *Automatica*, Vol. 29, No. 6, pp. 1361–1375, 1993.
- [13] H. Kwakernaak and M. Šebek “Polynomial  $J$ -Spectral Factorization”, *IEEE Transactions on Automatic Control*, Vol. 39, No. 2, pp. 315–328, 1994.
- [14] H. Kwakernaak and M. Šebek “The Polynomial Toolbox for MATLAB”, *Version 1.5*, 1997. Available as freeware at <http://www.math.utwente.nl/polbox>.
- [15] S. Labhalla, H. Lombardi and R. Marlin “Algorithmes de calcul de la réduction de Hermite d’une matrice à coefficients polynomiaux”, *Theoretical Computer Science*, Vol. 161, No. 1-2, pp. 69–92, 1996.
- [16] P. Lancaster and M. Tismenetsky “The Theory of Matrices - Second Edition with Applications”, *Academic Press*, 1985.
- [17] M. Šebek, S. Pejchová, R. C. W. Strijbos and D. Henrion “Polynomial Matrix Operations in MATLAB”, *International Conference on Computer Aided Systems Theory and Technology*, Las Palmas de Gran Canaria, Canary Islands, Spain, February 1997.
- [18] M. Šebek, S. Pejchová, D. Henrion and H. Kwakernaak “Numerical Methods for Zeros and Determinant of Polynomial Matrix”, *IEEE Mediterranean Symposium on New Directions in Control and Automation*, pp. 473–477, Chania, Crete, Greece, June 1996.
- [19] A. Storjohann “Computation of Hermite and Smith Normal Forms of Matrices”, *Master’s Thesis*, University of Waterloo, Ontario, Canada, 1994.
- [20] G. Villard “Computing Popov and Hermite Forms of Polynomial Matrices”, *International Symposium on Symbolic and Algebraic Computation*, Zurich, Switzerland, ACM Press, pp. 250–258, July 1996.