Approximate Least Common Multiple of Several Polynomials Using the ERES Division Algorithm

Dimitrios Christou*

Department of Science and Mathematics, School of Liberal Arts and Sciences, Deree - The American College of Greece, 6 Gravias St., GR-15342, Athens, Greece.

Nicos Karcanias

Systems and Control Centre, School of Mathematics, Computer Science, and Engineering, City University London, Northampton Square, EC1V 0HB, London, United Kingdom.

Marilena Mitrouli**

Department of Mathematics, National and Kapodistrian University of Athens, Panepistemiopolis GR-15773, Athens, Greece.

Abstract

In this paper a numerical method for the computation of the approximate least common multiple (ALCM) of a set of several univariate real polynomials is presented. The most important characteristic of the proposed method is that it avoids root finding procedures and computations involving the greatest common divisor (GCD). Conversely, it is based on the algebraic construction of a special matrix which contains key data from the original set of polynomials and leads to the formulation of a linear system which provides the degree and the coefficients of the ALCM using low-rank approximation techniques and numerical optimization tools particularly in the presence of inaccurate data. The numerical stability and complexity of the method is analysed and a comparison with other methods is provided.

Keywords: Greatest common divisor, linear systems, shifting operation, numerical errors, least squares
2010 MSC: 15A23, 65F05, 65F30, 65Y20

1. Introduction

The area of *approximate algebraic computations* is a fast growing area which has attracted the interest of many researchers in recent years. Two well known problems of algebraic computations are the computation of the Greatest Common Divisor (GCD) and the computation of the Least Common Multiple (LCM) of sets of polynomials. Both of them have widespread applications in several branches of

*Corresponding author
**Principal corresponding author
Email addresses: dchristou@acg.edu (Dimitrios Christou), N.Karcanias@city.ac.uk (Nicos Karcanias), mmitroul@math.uoa.gr (Marilena Mitrouli)

mathematics and engineering. The problem of computing an approximate LCM of many polynomials is an integral part of the algebraic synthesis methods in Linear Control [1, 2, 3]. This problem is linked to the computation of polynomial matrix fraction descriptions of rational matrices. In fact, the computation of the LCM is central in the derivation of minimal polynomial representations and this has several applications, such as the representation of vector transfer functions problem [1], the squaring- down problem [4], and the pole assignment by dynamic precompensation [1].

In the case of two polynomials $\mathcal{P} = \{p_1(s), p_2(s)\}$ we have the standard identity that $p_1(s) \cdot p_2(s) = \gcd\{\mathcal{P}\} \cdot \mathrm{lcm}\{\mathcal{P}\}$, which indicates the natural linking of the two problems. For a given set of polynomials with randomly selected coefficients (generic values) the existence of a non-trivial (different than 1) GCD is a non-generic property [5]. This implies that the set of polynomials with a non-trivial GCD belongs to an algebraic variety of Lebesgue measure zero. Use of the genericity property of polynomials implies that for any generic set of polynomials the LCM is the product of the set of polynomials, since any subset of them will be coprime. For non-generic sets of polynomials, is known that the LCM exists and divides the product of the polynomials. This suggests that there are fundamental differences between the two computational problems.

Existing LCM methods rely on GCD algorithms and numerical factorization of polynomials [6], and the computation of a minimal basis of special polynomial matrices [7, 8, 9] and use of algebraic identities. In [6], the above standard algebraic identity of the LCM is generalized and this provides a symbolic procedure for the LCM computation, as well as the basis for a robust numerical LCM algorithm that avoids any computation of the roots of the corresponding polynomials, and also leads to an approximate solution when the data are given inexactly or there are computational errors. The associativity property of the LCM [6] is fundamental for this computation and the developed methodology depends on the proper transformation of the LCM computations to real matrix computations.

An alternative approach is presented in [10]. This approach is based on standard system theory concepts and also avoids root finding as well as GCD computations. The results in [10] led to a robust procedure for the LCM of several polynomials and enabled the computation of approximate values when the original data have numerical inaccuracies. However, the complexity of the developed algorithm in [10] dramatically increases when the size of $\mathcal{P}$ is increased.

Another approach to finding approximate LCMs is also presented in [11] and the proposed algorithm, developed for two univariate or multivariate polynomials, uses the geometrical notion of principal angles between vector spaces and the interrelation of the LCM with the GCD.

In this paper the aim is to develop a numerical method for the approximate computation of the LCM of sets of several real univariate polynomials by completely avoiding root finding and GCD computations. The developed method relies on the properties of polynomial long division and the use of algebraic procedures in the context of the ERES methodology [12, 13, 14]. Specifically, it involves the formulation of a linear system which provides the degree and the coefficients of the LCM when it is solved. Depending on the nature of the data, the solution of this system may be sought either using direct algebraic methods, such as LU factorization, or using optimization methods, such as linear least-squares, for approximate solutions. The paper is structured as follows.

In Section 2, we provide a matrix representation for the LCM of sets of many polynomials based on the theoretical properties of the LCM and the Euclidean identity. This representation leads to a definition for the approximate LCM of sets of several univariate polynomials. In Section 3, we develop the *ERES division algorithm* which is an integral part of current LCM method, and in Section 4 we discuss and analyse the theoretical concepts of the computation of the LCM of a polynomial set without using the GCD. In Section 5, the obtained theoretical results formulate the Hybrid LCM method and we discuss the issues of its implementation. The numerical properties of this method are analysed, and in Section 6 various examples with comparison to other LCM methods are given for the demonstration of the developed procedures.

Notation. In the following, $\mathbb{N}$ denotes the set of natural numbers and $\mathbb{R}$ denotes the field of real numbers. $\mathbb{R}[s]$ denotes the ring of polynomials in one variable over $\mathbb{R}$. Capital letters denote matrices, i.e. $A \in \mathbb{R}^{m \times n}$ is a $m \times n$ matrix with elements from $\mathbb{R}$. Small underlined letters are used for vectors, i.e. $\underline{v} \in \mathbb{R}^m$ is a column vector with m elements from $\mathbb{R}$. The transpose vector of $\underline{v}$ (row vector) is denoted by $\underline{v}^t$. The greatest common divisor and the least common multiple of a set of polynomials $\mathcal{P}$ are denoted by $\gcd\{\mathcal{P}\}$ and $\operatorname{lcm}\{\mathcal{P}\}$, respectively. Throughout the paper a polynomial in $\mathbb{R}[s]$ is denoted by $p(s)$ and its degree will be $\deg\{p\}$, $\rho(\cdot)$ denotes the rank of a matrix, $\|\cdot\|_F$ denotes the Frobenius matrix norm and $\|\cdot\|_2$ denotes the Euclidean vector norm, and $\operatorname{diag}_m\{\underline{v}\}$ stands for a diagonal matrix with m rows created from a vector $\underline{v}$. The mathematical operators $\triangleq$ and $\approx$ indicate equality by definition and approximate equality, respectively. Finally, we use the Maple notation for the range of numbers that an index can take on, i.e. $i = 1..h$ means that the index i can take on all the integer numbers from 1 to h.

2. The matrix representation of the LCM of several polynomials and its approximation

The next set of real polynomials in one variable will be considered in the following:

$$\mathcal{P}_{h,n} = \Big\{ p_i(s) \in \mathbb{R}[s],\ i = 1, 2, \dots, h \text{ with } d_i = \deg\{p_i\} > 0 \text{ and } n = \max_i(d_i) \Big\} \tag{1}$$

The polynomials $p_i(s)$ are represented with respect to the maximum degree n by

$$p_i(s) = \sum_{j=0}^{n} c_j^{(i)}\, s^j = \Big[0, \dots, 0, c_{d_i}^{(i)}, \dots, c_0^{(i)}\Big] \begin{bmatrix} s^n \\ \vdots \\ s \\ 1 \end{bmatrix} \tag{2}$$

where $c_{d_i}^{(i)} \neq 0$ for every $i = 1..h$ and $c_n^{(i)} \neq 0$ for at least one $i \in \{1..h\}$.

We also consider a real univariate polynomial of degree $d > n$ represented as

$$l(s) = \sum_{j=0}^{d} a_j\, s^j = [a_d, \dots, a_0] \begin{bmatrix} s^d \\ \vdots \\ s \\ 1 \end{bmatrix} \text{ and } d = \sum_{i=1}^{h} d_i \tag{3}$$

Since $d_i \leq n < d$ for all $i = 1..h$, according to the Euclidean identity there exist real polynomials $q_i(s)$ (quotients) and $r_i(s)$ (remainders) such that

$$l(s) = q_i(s)\, p_i(s) + r_i(s) \tag{4}$$

with $\deg\{q_i\} = d - d_i$ and $\deg\{r_i\} \leq d_i - 1$.

For each one of the polynomials $p_i(s)$, $q_i(s)$, $r_i(s)$, and $l(s)$ we denote the corresponding vector of coefficients by

$$\underline{p}_i^t = \left[c_{d_i}^{(i)}, \ldots, c_0^{(i)}\right], \quad \underline{q}_i^t = \left[q_{d-d_i}^{(i)}, \ldots, q_0^{(i)}\right], \quad \underline{r}_i^t = \left[r_{d_i-1}^{(i)}, \ldots, r_0^{(i)}\right],$$

and $\underline{a}^t = [a_d, \ldots, a_0]$, respectively. Then every polynomial in the set $\mathcal{P}_{h,n}$ with degree $d_i = n$ can be associated with an upper trapezoidal matrix of the form:

$$\begin{aligned} P_i &\triangleq \operatorname{diag}_{(d+1)}\{\underline{p}_i^t\} = \\ &= \begin{bmatrix} c_{d_i}^{(i)} & c_{d_i-1}^{(i)} & \ldots & c_0^{(i)} & 0 & \ldots & 0 \\ 0 & c_{d_i}^{(i)} & c_{d_i-1}^{(i)} & \ldots & c_0^{(i)} & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & c_{d_i}^{(i)} & c_{d_i-1}^{(i)} & \ldots & c_0^{(i)} \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+n+1)} \end{aligned} \tag{5}$$

If $d_i < n$, then $P_i \triangleq \left[O_{n-d_i} \,|\, \operatorname{diag}\{\underline{p}_i^t\}\right]$ where O_{n-d_i} is a $(d+1)\times(n-d_i)$ zero matrix. Similarly, the matrices

$$Q_i \triangleq \operatorname{diag}_{(d_i+1)}\{\underline{q}_i^t\} \in \mathbb{R}^{(d_i+1)\times(d+1)} \tag{6}$$

$$R_i \triangleq \left[O_{d+n-d_i} \,|\, \operatorname{diag}_{(d_i+1)}\{\underline{r}_i^t\}\right] \in \mathbb{R}^{(d_i+1)\times(d+n+1)} \tag{7}$$

$$L_i \triangleq \left[O_{n-d_i} \,|\, \operatorname{diag}_{(d_i+1)}\{\underline{a}^t\}\right] \in \mathbb{R}^{(d_i+1)\times(d+n+1)} \tag{8}$$

can be associated with the polynomials $q_i(s)$, $r_i(s)$, and $l(s)$, respectively. The extended matrix

$$\widehat{P} = \begin{bmatrix} P_1 \\ \vdots \\ P_h \end{bmatrix} \in \mathbb{R}^{(hd+h)\times(d+n+1)} \tag{9}$$

is a representative matrix for the set $\mathcal{P}_{h,n}$ and the Euclidean identity (4) can be expressed in the matrix form as

$$\widehat{L} = \widehat{Q} \cdot \widehat{P} + \widehat{R} \tag{10}$$

$$\widehat{L} = \begin{bmatrix} L_1 \\ \vdots \\ L_h \end{bmatrix} \in \mathbb{R}^{(d+h)\times(d+n+1)}, \quad \widehat{R} = \begin{bmatrix} R_1 \\ \vdots \\ R_h \end{bmatrix} \in \mathbb{R}^{(d+h)\times(d+n+1)} \tag{11}$$

where $\widehat{L}$ is the extended matrix associated with the polynomial $l(s)$ as defined in (3), $\widehat{R}$ is the extended matrix associated with the remainders after division of the polynomial $l(s)$ by the polynomials of $\mathcal{P}_{h,n}$, and

$$\widehat{Q} = \operatorname{diag}\{Q_1, Q_2, \ldots, Q_h\} \in \mathbb{R}^{(d+h)\times(hd+h)} \tag{12}$$

is the block-diagonal matrix associated with the quotients after division of the polynomial $l(s)$ by the polynomials of $\mathcal{P}_{h,n}$.

Therefore, if $l(s)$ in (4) represents the LCM of $\mathcal{P}_{h,n}$, the following basic definition can be provided.

Definition 1. Given a set of several polynomials $\mathcal{P}_{h,n}$, the *exact least common multiple* of the polynomials of the set is a polynomial of the smallest possible degree $\ell \leq hn$, such that:

$$\widehat{L} = \widehat{Q} \cdot \widehat{P} \tag{13}$$

Equivalently, using an appropriate matrix norm [15], denoted by $\| \cdot \|$, the next equality is satisfied:

$$\|\widehat{L} - \widehat{Q} \cdot \widehat{P}\| = 0 \tag{14}$$

Remark 1. Throughout the paper by LCM we mean the exact least common multiple of $\mathcal{P}_{h,n}$.

The following lemma provides the theoretical foundation of the proposed LCM method and it is based on the generalization of a characteristic LCM property from the number case to the polynomial one.

Lemma 1. *Given a set of polynomials $\mathcal{P}_{h,n}$, the LCM of $\mathcal{P}_{h,n}$ is a real polynomial with the least possible degree $\ell \leq \sum_{i=1}^{h} \deg\{p_i\}$ and every polynomial $p_i(s)$ divides evenly into LCM.*

Proof. If we denote by $l(s)$ the LCM of $\mathcal{P}_{h,n}$ with degree ℓ and $d_i = \deg\{p_i\}$ for every $i = 1, 2, \ldots, h$, then two different cases must be considered:

1. If the polynomials $p_i(s) \in \mathcal{P}_{h,n}$ cannot be factored into polynomials in $\mathbb{R}[s]$, the LCM is given by the product of all $p_i(s) \in \mathcal{P}_{h,n}$, such that

$$l(s) = \prod_{i=1}^{h} p_i(s) \quad \text{and} \quad \ell = \sum_{i=1}^{h} d_i$$

   Obviously, in this case every $p_i(s)$ divides evenly into $l(s)$.

2. Assuming that the polynomials $p_i(s) \in \mathcal{P}_{h,n}$ can be factored into polynomials $t_{i,j_i}(s) \in \mathbb{R}[s]$ irreducible over $\mathbb{R}$, i.e. $t_{i,j_i}(s)$ is non-constant and cannot be represented as the product of two or more non-constant polynomials from $\mathbb{R}[s]$, then

$$p_i(s) = t_{i,1}^{k_{i,1}}(s) \cdot t_{i,2}^{k_{i,3}}(s) \cdots t_{i,j_i}^{k_{i,j_i}}(s) \tag{15}$$

   with powers k_{i,j_i} such that $k_{i,1} + \ldots + k_{i,j_i} = d_i$ for any positive integer $j_i \leq d_i$ and $i = 1..h$.
   The set containing the factors $t_{i,j}(s)$ of every polynomial $p_i(s)$ raised to a power $k_{i,j}$ can be defined by

$$\mathcal{T}_f = \left\{ t_{i,j}^{k_{i,j}}(s) \in \mathbb{R}[s] \text{ for } i = 1..h \text{ and } j = 1, 2, \ldots, j_i \leq d_i \right\}$$

   and we consider the subsets of $\mathcal{T}_f$:

$$\mathcal{T}_{cf} = \left\{ t_{i,j}^{k_{i,j}}(s) : t_{i,j}(s)\text{=common factor raised to the highest power } k_{i,j} \right\}$$

$$\mathcal{T}_{ncf} = \left\{ t_{i,j}^{k_{i,j}}(s) : t_{i,j}(s)\text{=non-common factor} \right\}$$

Then, the LCM is given by

$$l(s) = \prod_{i,j} t_{i,j}^{k_{i,j}}(s), \quad \text{where} \;\; t_{i,j}^{k_{i,j}}(s) \in \mathcal{T}_{cf} \cup \mathcal{T}_{ncf}.$$

Therefore, considering (15), every $p_i(s)$ divides evenly into $l(s)$. Moreover, since $\mathcal{T}_{cf} \cup \mathcal{T}_{ncf} \subseteq \mathcal{T}_f$, it follows that

$$\ell = \sum_{i,j} k_{i,j} \leq \sum_{i=1}^{h} d_i$$

which implies that the LCM is a real polynomial with maximum degree equal to the sum of the degrees of all $p_i(s) \in \mathcal{P}_{h,n}$.

□

The above Lemma 1 implies that:

$$r_i(s) = 0, \quad \forall\, i = 1..h \tag{16}$$

Hence, when $l(s)$ represents the LCM, the matrix $\widehat{R}$ in (10) is actually a zero matrix. Thus, from the algebraic viewpoint the LCM of $\mathcal{P}_{h,n}$ satisfies the equation:

$$\|\widehat{R}\| = 0 \tag{17}$$

However, when the given data contain numerical inaccuracies an approximate LCM must be sought. Therefore, based on the developed matrix representation of the LCM, the following definition of the approximate LCM can be provided.

Definition 2. Given a set of several polynomials $\mathcal{P}_{h,n}$ and a specified small tolerance ε, the *approximate least common multiple* of the set, denoted by ALCM, is a polynomial of the smallest possible degree $\ell \leq hn$, such that the next inequalities are satisfied:

$$\|\widehat{L} - \widehat{Q} \cdot \widehat{P}\| \leq \varepsilon \;\Leftrightarrow\; \|\widehat{R}\| \leq \varepsilon \tag{18}$$

3. The ERES representation of the polynomial Euclidean division

The *ERES method* [12, 13, 14, 16] was originally developed for the computation of the approximate GCD of sets of many polynomials using Gaussian transformation and shifting. In this section the algebraic relationship between ERES and the Euclidean algorithm for two univariate polynomials will be analysed. Specifically, it will be shown how the remainder of the Euclidean division of two polynomials can be represented as a matrix product where the matrices correspond to the applied ERES operations.

3.1. Definition of the ERES operations and background results

The following describe the basic notions of the ERES methodology.

Definition 3 (ERES operations [12]). For any set $\mathcal{P}_{h,n}$ as defined in (1), a vector representative $\underline{p}(s)$ and an associated matrix $P \in \mathbb{R}^{h\times(n+1)}$ are defined by

$$\underline{p}(s) = [\, p_1(s), \ldots, p_h(s)\,]^t = [\, \underline{p}_1, \ldots, \underline{p}_{h-1}, \underline{p}_h \,]^t \cdot \underline{e}_n(s) = P \cdot \underline{e}_n(s)$$

where $\underline{e}_n(s) = [s^n, s^{n-1}, \ldots, s, 1]^t$ and $\underline{p}_i \in \mathbb{R}^{n+1}$ for all $i = 1..h$. Then, the following operations are defined:

a) Elementary row operations with scalars from $\mathbb{R}$ on P.
b) Addition or elimination of zero rows (or columns) on P.
c) If $\underline{c}^{(i)} = [0, \ldots, 0, c_l^{(i)}, \ldots, c_{n+1}^{(i)}]^t \in \mathbb{R}^{n+1}$ with $c_l^{(i)} \neq 0$ is the i^{th} row of P, then we define the *shifting operation* as the next transformation:

$$\mathit{shf} : \mathrm{shf}(\underline{c}^{(i)}) = [c_l^{(i)}, \ldots, c_{n+1}^{(i)}, 0, \ldots, 0]^t \in \mathbb{R}^{n+1} \tag{19}$$

Type (a), (b) and (c) operations are referred to as *Extended-Row-Equivalence and Shifting (ERES) operations*. The ERES operations without using the shifting operation are referred to as *ERE operations*.

Remark 2. The matrix P is formed directly from the coefficients of the polynomials of the set $\mathcal{P}_{h,n}$ and typically its size is $h \times (n+1)$. However, for the LCM computation, if all the polynomials $p_i(s)$ have a factor of the form s^{k_i}, the shifting operation implies that

$$\mathrm{lcm}\,\{\mathcal{P}_{h,n}\} = s^k\,\mathrm{lcm}\,\{\mathrm{shf}(\mathcal{P}_{h,n})\}$$

where $k = \min_{1\leq i\leq h}\{k_i\}$. The factor s^{k_i} of any polynomial in the given set, if it exists, can be defined by the maximum number of consecutive coefficients which are zero, starting from the constant. Therefore, in this case we may consider a reduced matrix P of size $h \times (n-k+1)$ for the set $\mathcal{P}_{h,n}$.

Remark 3. The matrix $\widehat{P}$ as defined in (9) can be regarded as the *extended form of* P constructed for the matrix representation of the LCM of sets of several polynomials.

3.2. The shifting operation for upper trapezoidal matrices

Given a matrix $A = [\underline{a}_1^t, \underline{a}_2^t, \ldots, \underline{a}_m^t]^t \in \mathbb{R}^{m\times n}$, where $\underline{a}_i \in \mathbb{R}^n$ for $i = 1, 2, \ldots, m$ are the row-vectors of A, the *matrix-shifting* is defined as the application of shifting to every row of A. The shifted form of A will be denoted by

$$\mathrm{shf}(A) \triangleq A^* = [\mathrm{shf}(\underline{a}_1^t), \mathrm{shf}(\underline{a}_2^t), \ldots, \mathrm{shf}(\underline{a}_m^t)]^t \in \mathbb{R}^{m\times n} \tag{20}$$

Furthermore, if $A \in \mathbb{R}^{m\times n}$ with $m < n$ is a non-singular upper trapezoidal matrix with rank $\rho(A) = m$, then in [14] it is proven that there exists a square matrix $S \in \mathbb{R}^{n\times n}$ such that:

$$\mathrm{shf}(A) = A \cdot S \tag{21}$$

The matrix S is referred to as the *shifting matrix* of A and is given by the formula:

$$S = \sum_{i=1}^{m} \left(I_n - A^\dagger A + A^\dagger A_i\right) J_i \tag{22}$$

where, for every $i = 1, 2, \ldots, m$, A_i denotes the $m \times n$ matrix derived from A if we keep only the i^{th} row and zero all the others, $J_i \in \mathbb{R}^{n \times n}$ is a permutation matrix which gives the appropriate shifting to each A_i, respectively, and $A^\dagger \in \mathbb{R}^{n \times m}$ is the pseudo-inverse of A, such that $A\,A^\dagger = I_m$. The shifting transformation of a nonsingular upper trapezoidal matrix is a reversible process, unless the shifted matrix is rank deficient.

3.3. The ERES division algorithm

Using the ERES operations, the Euclidean division of two arbitrary polynomials will be presented in matrix form. The developed algorithmic procedure will provide an elegant and quick way to compute the quotient and the remainder of the division $l(s)/p_i(s)$ which is essential for the computation of the ALCM in the current study.

For the development of a division algorithm based on the ERES operations we consider two arbitrary polynomials in one variable with degrees m and n such that $m > n$:

$$a\,(s) = \sum_{i=0}^{m} a_i\, s^i,\ a_m \neq 0 \quad \text{and} \quad b\,(s) = \sum_{i=0}^{n} b_i\, s^i,\ b_n \neq 0, \quad m, n \in \mathbb{N} \tag{23}$$

The set of all the pairs of polynomials $\big(a(s), b(s)\big)$ is defined by

$$\mathcal{D}_{m,n} = \Big\{(a(s), b(s)) : a(s), b(s) \in \mathbb{R}[s],\ m = \deg\{a(s)\} \geq \deg\{b(s)\} = n\Big\}$$

For any pair $\mathcal{P} = \big(a(s), b(s)\big) \in \mathcal{D}_{m,n}$, we define a vector representative $\underline{p}(s)$ and an associated matrix $P \in \mathbb{R}^{2 \times (m+1)}$ such that

$$\underline{p}(s) = \begin{bmatrix} a(s) \\ b(s) \end{bmatrix} = \begin{bmatrix} \underline{a}^t \\ \underline{b}^t \end{bmatrix} \cdot \underline{e}_m(s) = P \cdot \underline{e}_m(s)$$

where $\underline{a}^t = [a_m, \ldots, a_0] \in \mathbb{R}^{m+1}$, $\underline{b}^t = [\underbrace{0, \ldots, 0}_{m-n}, b_n, \ldots, b_0\,] \in \mathbb{R}^{m+1}$ are the coefficient vectors of $a(s)$ and $b(s)$, respectively, and $\underline{e}_m(s) = [s^m, s^{m-1}, \ldots, s, 1]^t$ is the power basis vector of s.

The matrix P is formed directly from the coefficients of the given polynomials $a(s)$ and $b(s)$ and it has the following form:

$$P = \begin{bmatrix} a_m & \ldots & a_{n+1} & a_n & \ldots & a_0 \\ 0 & \ldots & 0 & b_n & \ldots & b_0 \end{bmatrix} \tag{24}$$

Since P is a non-singular matrix, using (21) and (22) there exists a shifting matrix $S \in \mathbb{R}^{(m+1)\times(m+1)}$ such that:

$$P^* = P \cdot S \tag{25}$$

The matrix $P^* \in \mathbb{R}^{2 \times (m+1)}$ is the shifted form of P such that

$$P^* = \begin{bmatrix} a_m & \ldots & a_{n+1} & a_n & \ldots & a_0 \\ b_n & \ldots & b_0 & 0 & \ldots & 0 \end{bmatrix} \tag{26}$$

and it is the associated matrix of the pair

$$\mathcal{P}^* = \big(a(s), s^{m-n}\, b(s)\big) \in \mathcal{D}_{m,m}$$

Now, given a pair of polynomials $\mathcal{P}$, if $a(s)$ is divided by $b(s)$, then the next identity occurs:

$$\frac{a(s)}{b(s)} = \frac{a_m}{b_n}\, s^{m-n} + \frac{r_1(s)}{b(s)} \quad \text{or} \quad a(s) = \frac{a_m}{b_n}\, s^{m-n}\, b(s) + r_1(s) \tag{27}$$

This is the first and most basic step of the Euclidean algorithm. The polynomial $r_1(s) \in \mathbb{R}[s]$ is a partial remainder of the division $a(s)/b(s)$ and it is given by

$$r_1(s) = \sum_{i=m-n}^{m-1} \left(a_i - \frac{a_m}{b_n}\, b_{i-(m-n)} \right) s^i + \sum_{i=0}^{m-n-1} a_i\, s^i \tag{28}$$

The remainder $r_1(s)$ can be computed by applying ERES operations to the matrix P of the pair $\mathcal{P}$.

Lemma 2. *Given a pair of polynomials $\mathcal{P} = \big(a(s), b(s)\big) \in \mathcal{D}_{m,n}$, there exists a polynomial $r_1(s) \in \mathbb{R}[s]$ with $deg\{r_1(s)\} < m$, such that*

$$a(s) = \frac{a_m}{b_n}\, s^{m-n}\, b(s) + r_1(s) \tag{29}$$

with

$$r_1(s) = \underline{v}^t \cdot R_1 \cdot \underline{e}_m(s) \tag{30}$$

where $R_1 \in \mathbb{R}^{2\times(m+1)}$ is the matrix which occurs after applying the ERES operations to the matrix P of the pair $\mathcal{P}$, and $\underline{v}^t = [0,\ 1]$.

Proof. Considering the division $a(s)/b(s)$ and according to Euclid's algorithm, there is a polynomial $r_1(s)$ with degree $0 \leq deg\{r_1(s)\} < m$ such that

$$r_1(s) = a(s) - \frac{a_m}{b_n}\, s^{m-n}\, b(s) \tag{31}$$

Then, the above equation can be written as a matrix product in the following form:

$$r_1(s) = [0,\ 1] \begin{bmatrix} 0 & 1 \\ 1 & -\dfrac{a_m}{b_n} \end{bmatrix} \begin{bmatrix} a(s) \\ s^{m-n}\, b(s) \end{bmatrix} = [0,\ 1] \begin{bmatrix} 0 & 1 \\ 1 & -\dfrac{a_m}{b_n} \end{bmatrix} P^*\, \underline{e}_m(s) \tag{32}$$

If we denote by

$$Q = \begin{bmatrix} 0 & 1 \\ 1 & -\dfrac{a_m}{b_n} \end{bmatrix} \tag{33}$$

and $\underline{v}^t = [0,\ 1]$, then (25) and (33) provide the next matrix representation for the remainder polynomial:

$$r_1(s) = \underline{v}^t \cdot Q \cdot P \cdot S \cdot \underline{e}_m(s) \tag{34}$$

Therefore, there exists a matrix $R_1 \in \mathbb{R}^{2\times(m+1)}$ such that

$$R_1 = Q \cdot P \cdot S \quad \text{and} \quad r_1(s) = \underline{v}^t \cdot R_1 \cdot \underline{e}_m(s) \tag{35}$$

However, the same matrix R_1 can be obtained by applying the ERES operations to the matrix P according to the next three-step procedure:

1. Apply shifting to the rows of P. Let $S \in \mathbb{R}^{(m+1)\times(m+1)}$ be the appropriate shifting matrix.

$$P^{(1)} = P \cdot S = \begin{bmatrix} a_m & a_{m-1} & \dots & a_{m-n} & a_{m-n-1} & \dots & a_0 \\ b_n & b_{n-1} & \dots & b_0 & 0 & \dots & 0 \end{bmatrix}$$

2. Reorder the rows of the matrix $P^{(1)}$.

$$P^{(2)} = J \cdot P^{(1)} = \begin{bmatrix} b_n & \dots & b_0 & 0 & \dots & 0 \\ a_m & \dots & a_{m-n} & a_{m-n-1} & \dots & a_0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

3. Apply LU factorization [17, 15] to $P^{(2)}$. Then,

$$P^{(3)} = L^{-1} \cdot P^{(2)} =$$
$$= \begin{bmatrix} b_n & \dots & b_0 & 0 & \dots & 0 \\ 0 & \dots & a_{m-n} - b_0 \dfrac{a_m}{b_n} & a_{m-n-1} & \dots & a_0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ \dfrac{a_m}{b_n} & 1 \end{bmatrix}$$

The above procedure can be described by the equation

$$P^{(3)} = L^{-1} \cdot J \cdot P \cdot S \tag{36}$$

which represents all the ERES transformations. Naturally, we have that

$$L^{-1} \cdot J = \begin{bmatrix} 1 & 0 \\ -\dfrac{a_m}{b_n} & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -\dfrac{a_m}{b_n} \end{bmatrix} = Q$$

and thus, considering (35), it holds:

$$P^{(3)} = Q \cdot P \cdot S = R_1 \tag{37}$$

□

The following theorem refers to the matrix representation of the remainder of the Euclidean division of two real polynomials and establishes the algebraic relationship between the ERES method and Euclid's division algorithm.

Theorem 1. *Applying the Euclidean algorithm to a pair of real polynomials $\mathcal{P} = (a(s), b(s)) \in \mathcal{D}_{m,n}$, there exist unique real polynomials $q(s)$, $r(s)$ with degrees $\deg\{q(s)\} = m - n$ and $\deg\{r(s)\} < n$, respectively, such that*

$$a(s) = q(s) \cdot b(s) + r(s)$$

and the remainder $r(s)$ can be expressed in the matrix form

$$r(s) = \underline{v}^t \cdot R_\eta \cdot \underline{e}_n(s) \tag{38}$$

where $R_\eta \in \mathbb{R}^{2\times(n+1)}$ is the matrix which is obtained from the iterative application of the ERES operations to the matrix P of the pair $\mathcal{P}$, and $\underline{v}^t = [0,\ 1]$, $\underline{e}_n(s) = [s^n, s^{n-1}, \dots, s, 1]^t$. Furthermore, the quotient $q(s)$ is given by

$$q(s) = \sum_{i=0}^{m-n} q_i\, s^i = \sum_{i=0}^{m-n} l_{m-n+1-i}\, s^i \tag{39}$$

where the coefficients $q_i = l_{m-n+1-i}$ are implicitly computed during the construction of R_η. In addition, the matrix R_η satisfies the following relation:

$$R_\eta = Q_\eta \cdot P \cdot S_\eta \tag{40}$$

where $Q_\eta \in \mathbb{R}^{2\times 2}$ is the matrix which is obtained from the iterative applications of the ERE operations to the matrix P, and $S_\eta \in \mathbb{R}^{(n+1)\times(n+1)}$ is the matrix which accounts for all the iterative applications of the shifting operations to the matrix P.

Proof. The Euclidean division $a(s)/b(s)$ includes the following steps:

$$\begin{aligned} a(s) &= l_1\, s^{m-n}\, b(s) + r_1(s) \\ r_1(s) &= l_2\, s^{k_1-n}\, b(s) + r_2(s) \\ &\vdots \\ r_i(s) &= l_{i+1}\, s^{k_i-n}\, b(s) + r_{i+1}(s) \\ &\vdots \\ r_{\eta-1}(s) &= l_\eta\, s^{k_{\eta-1}-n}\, b(s) + r_\eta(s) \end{aligned}$$

where $r_i(s) \in \mathbb{R}[s]$ is a polynomial with degree $k_i = \deg\{r_i(s)\}$, $i = 1, 2, \ldots, \eta$ and η is the total number of steps in Euclid's algorithm for which $\eta = m - n + 1$. Normally, $k_i > n$ for $i = 1, 2, \ldots, \eta - 2$ and $k_{\eta-1} = n$, whereas $k_\eta < n$.

Since $P \in \mathbb{R}^{2\times(m+1)}$ is the matrix of the pair $\mathcal{P} = (a(s), b(s))$, then using Lemma 2, the remainder $r_1(s)$ in the first iteration is given by

$$r_1(s) = \underline{v}^t \cdot R_1 \cdot \underline{e}_m(s) \quad \text{and} \quad R_1 = L_1^{-1} \cdot J \cdot P \cdot S_1$$

where

$$L_1 = \begin{bmatrix} 1 & 0 \\ l_1 & 1 \end{bmatrix},$$

the matrix S_1 is the appropriate shifting matrix, and J is an elementary permutation matrix. Similarly, in the second iteration of the process, if L_2 applies ERE transformations to R_1 and S_{R_1} shifts its rows, the next matrix is obtained:

$$R_2 = L_2^{-1} \cdot R_1 \cdot S_{R_1} \cdot Z_{k_1} \tag{41}$$

The matrix $Z_{k_1} \in \mathbb{R}^{(m+1)\times(k_1+1)}$ is used in order to reduce the column dimension of R_1 by deleting the last $m - k_1$ zero columns. Hence, R_1 is linked together with R_2. If the same steps are followed, the next generalized form can be obtained:

$$r_i(s) = \underline{v}^t \cdot R_i \cdot \underline{e}_{k_{i-1}}(s), \quad i = 2, 3, \ldots, \eta \tag{42}$$

with

$$R_i = L_i^{-1} \cdot R_{i-1} \cdot S_{R_{i-1}} \cdot Z_{k_{i-1}}, \quad i = 2, 3, \ldots, \eta \tag{43}$$

The final matrix R_η is associated with the remainder $r(s)$ of the Euclidean division $a(s)/b(s)$, such that

$$r(s) \triangleq r_\eta(s) = \underline{v}^t \cdot R_\eta \cdot \underline{e}_n(s)$$

□

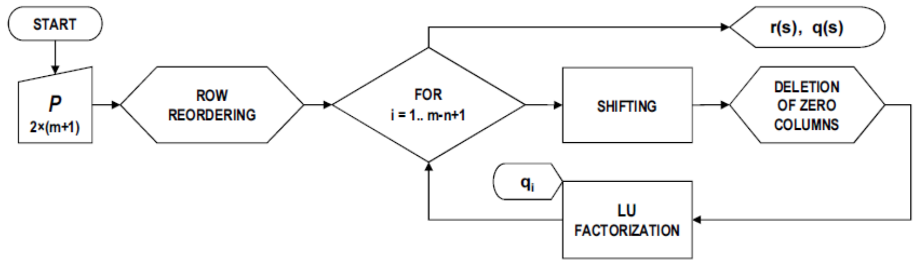

Figure 1: The ERES Division Algorithm

Remark 4. The developed procedure as described in the proof of Theorem 1 requires $m - n + 1$ iterations (for-loop) to be completed, which are referred to as *ERES iterations*. This procedure forms the *ERES division algorithm*. A flowchart of this algorithm is presented in Figure 1.

Definition 4. Given two polynomials $a(s),\ b(s) \in \mathbb{R}[s]$ with degrees m and n, respectively, and $m > n$, the transformation

$$\begin{bmatrix} a(s) \\ b(s) \end{bmatrix} \overset{ERES}{\Longrightarrow} \begin{bmatrix} b(s) \\ r(s) \end{bmatrix} \tag{44}$$

represents the Euclidean division of two polynomials using the ERES operations and is referred to as the *ERES division*.

Example 1. We consider the division $a(s)/b(s)$ of the polynomials:

$$\begin{aligned} a(s) &= 2\,s^3 + 3\,s^2 - 7\,s - 32, \qquad && \deg\{a(s)\} = m = 3 \\ b(s) &= s^2 + 4\,s + 5, && \deg\{b(s)\} = n = 2 \end{aligned}$$

with a corresponding matrix

$$P = \begin{bmatrix} 2 & 3 & -7 & -32 \\ 0 & 1 & 4 & 5 \end{bmatrix} \in \mathbb{R}^{2\times 4}$$

which is structured using the power basis vector $\left[s^3, s^2, s, 1\right]^t$. According to the Euclidean division it is

$$\frac{a(s)}{b(s)} = \frac{2\,s^3 + 3\,s^2 - 7\,s - 32}{s^2 + 4\,s + 5} = (2\,s - 5) + \frac{3\,s - 7}{s^2 + 4\,s + 5}$$

and the remainder is $r(s) = 3\,s - 7$. The ERES division is represented by the equation

$$R_\eta = Q_\eta \cdot P \cdot S_\eta = \begin{bmatrix} 1 & 4 & 5 \\ 0 & 3 & -7 \end{bmatrix}$$

where

$$Q_\eta = \begin{bmatrix} 0 & 1 \\ 1 & -(2-5) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad S_\eta = \begin{bmatrix} \frac{13}{2} & \frac{45}{2} & \frac{57}{10} \\ -3 & -5 & \frac{23}{5} \\ 1 & 1 & -\frac{12}{5} \\ 0 & 1 & 2 \end{bmatrix}$$

The remainder $r(s)$ of the division $a(s)/b(s)$ is given by

$$r(s) = [0,\ 1] \cdot R_\eta \cdot \begin{bmatrix} s^2 \\ s \\ 1 \end{bmatrix} = 3\,s - 7$$

The coefficients of the quotient $q(s) = 2\,s - 5$ are implicitly obtained during the construction process of the matrix Q_η. □

4. Computation of the LCM without using the GCD

In this section the theoretical concepts of the current approach for the ALCM computation are analysed.

4.1. The symbolic-rational formulation of the remainder sequence

Let the LCM of a given set $\mathcal{P}_{h,n}$ be a polynomial $l(s)$ with arbitrary (symbolic) coefficients in its generic form (3). The ERES division algorithm provides the means to compute the vectors $\underline{r}_i = \left[r^{(i)}_{d_i-1}, \ldots, r^{(i)}_0\right]^t$ of the remainders of the division $l(s)/p_i(s)$ in rational-symbolic form. Considering the overall matrix representation of the ERES division given in (40), each vector $\underline{r}_i$ is obtained from the second row of a matrix R_η and is used in order to construct the matrix $\widehat{R}$ as described in Section 2. However, this matrix is relatively large for effective use in a computational method.

A careful study of the elements of the obtained vectors $\underline{r}_i$ reveals that they are linear combinations of the arbitrary coefficients a_j of $l(s)$. More specifically,

$$\underline{r}_i = \begin{bmatrix} r^{(i)}_{d_i-1} \\ r^{(i)}_{d_i-2} \\ \vdots \\ r^{(i)}_0 \end{bmatrix} = \begin{bmatrix} f^{(i)}_{d_i-1,d}\, a_d + \ldots + f^{(i)}_{d_i-1,1}\, a_1 + f^{(i)}_{d_i-1,0}\, a_0 \\ f^{(i)}_{d_i-2,d}\, a_d + \ldots + f^{(i)}_{d_i-2,1}\, a_1 + f^{(i)}_{d_i-2,0}\, a_0 \\ \vdots \\ f^{(i)}_{0,d}\, a_d + \ldots + f^{(i)}_{0,1}\, a_1 + f^{(i)}_{0,0}\, a_0 \end{bmatrix} \tag{45}$$

where all $f^{(i)}_{\mu,\nu}$ for $\mu = 0, 1, \ldots, d_i - 1$ and $\nu = 0, 1, \ldots, d$ are real numbers which follow the analytic formula

$$f^{(i)}_{\mu,\nu} = \sum_{0<k<d} \lambda_k \frac{\left(c^{(i)}_0\right)^{m_0} \left(c^{(i)}_1\right)^{m_1} \cdots \left(c^{(i)}_{d_i}\right)^{m_{d_i}}}{\left(c^{(i)}_{d_i}\right)^{d-n+1}} \tag{46}$$

where $\lambda_k \in \mathbb{R}$ and $\sum_{j=0}^{d_i} m_j = d - n + 1$ for every $i = 1..h$.

Example 2. Let the two arbitrary polynomials:

$$l(s) = a_5\, s^5 + a_4\, s^4 + a_3\, s^3 + a_2\, s^2 + a_1\, s + a_0\ , \qquad \deg\{l(s)\} = d = 5$$
$$p(s) = c_3\, s^3 + c_2\, s^2 + c_1\, s + c_0\ , \qquad \deg\{p(s)\} = n = 3$$

The ERES division algorithm provides the following vector of coefficients $\underline{r}$ of the remainder of the division $l(s)/p(s)$.

$$\underline{r} = \begin{bmatrix} a_2 - \frac{c_2}{c_3}\,a_3 + \frac{\left(-c_1{c_3}^2+{c_2}^2c_3\right)}{{c_3}^3}\,a_4 + \frac{\left(-c_0{c_3}^2+2\,c_1c_2c_3-{c_2}^3\right)}{{c_3}^3}\,a_5 \\ a_1 - \frac{c_1}{c_3}\,a_3 + \frac{\left(-c_0{c_3}^2+c_1c_2c_3\right)}{{c_3}^3}\,a_4 + \frac{\left(c_0c_2c_3+{c_1}^2c_3-c_1{c_2}^2\right)}{{c_3}^3}\,a_5 \\ a_0 - \frac{c_0}{c_3}\,a_3 + \frac{c_0c_2}{{c_3}^2}\,a_4 + \frac{\left(c_0c_1c_3-c_0{c_2}^2\right)}{{c_3}^3}\,a_5 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-c_0{c_3}^2+2\,c_1c_2c_3-{c_2}^3}{{c_3}^3} & \frac{-c_1{c_3}^2+{c_2}^2c_3}{{c_3}^3} & -\frac{c_2}{c_3} & 1 & 0 & 0 \\ \frac{c_0c_2c_3+{c_1}^2c_3-c_1{c_2}^2}{{c_3}^3} & \frac{-c_0{c_3}^2+c_1c_2c_3}{{c_3}^3} & -\frac{c_1}{c_3} & 0 & 1 & 0 \\ \frac{c_0c_1c_3-c_0{c_2}^2}{{c_3}^3} & \frac{c_0c_2}{{c_3}^2} & -\frac{c_0}{c_3} & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

□

4.2. The real matrix representation of the remainder sequence

We may associate every vector $\underline{r}_i$ with a real matrix $F_i \in \mathbb{R}^{d_i \times (d+1)}$, such that

$$\underline{r}_i = F_i \cdot \underline{a} \tag{47}$$

where $\underline{a} = [a_d, \ldots, a_0]^t$ is the vector of coefficients of the polynomial $l(s)$ and every matrix F_i has the following structure:

$$F_i = \left[\widetilde{F}_i | I_{d_i}\right] = \begin{array}{c} \begin{array}{cccccc} a_d & \cdots & a_{d_i} & a_{d_i-1} & \cdots & a_0 \\ \downarrow & & \downarrow & \downarrow & & \downarrow \end{array} \\ \left[\begin{array}{ccc|ccc} f^{(i)}_{d_i-1,d} & \cdots & f^{(i)}_{d_i-1,d_i} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f^{(i)}_{0,d} & \cdots & f^{(i)}_{0,d_i} & 0 & \cdots & 1 \end{array}\right] \end{array} \tag{48}$$

where I_{d_i} is the $d_i \times d_i$ identity matrix and $i = 1..h$. Then, an extended matrix $F_{\mathcal{P}} \in \mathbb{R}^{d \times (d+1)}$ can be formed, such that

$$F_{\mathcal{P}} \cdot \underline{a} = \begin{bmatrix} F_1 \\ \vdots \\ F_h \end{bmatrix} \underline{a} = \begin{bmatrix} \underline{r}_1 \\ \vdots \\ \underline{r}_h \end{bmatrix} \tag{49}$$

The new matrix $F_{\mathcal{P}}$ is smaller than $\widehat{R}$ in (11) and free of arbitrary parameters, but the two matrices are linked together as explained in the following proposition.

Proposition 1. *For the matrices $\widehat{R}$ and $F_{\mathcal{P}}$ it holds:*

$$\|\widehat{R}\|_F \leq \sqrt{n+1}\, \|F_{\mathcal{P}}\, \underline{a}\|_2 \tag{50}$$

Proof. Considering the matrix $\widehat{R}$, we can obtain the following result:

$$\|\widehat{R}\|_F = \sqrt{\sum_{i=1}^{h} (d_i+1)\left(\|\underline{r}^t_i\|_2\right)^2} \leq \sqrt{(n+1)\sum_{i=1}^{h}\left(\|\underline{r}_i\|_2\right)^2} = \sqrt{n+1}\, \|F_{\mathcal{P}}\, \underline{a}\|_2$$

□

Therefore, if $\|F_{\mathcal{P}}\,\underline{a}\|_2 = 0$, we expect that $\|\widehat{R}\|_F = 0$. This result marks the transition from a symbolic-rational state of the problem to a numerical state where robust numerical methods can be employed for the computation of an approximate LCM. Solving the linear system

$$F_{\mathcal{P}} \cdot \underline{a} = \underline{0} \tag{51}$$

introduces a new algorithmic procedure for the numerical computation of the LCM of several polynomials and its approximation. The following result reveals a special property of the matrix $F_{\mathcal{P}}$.

Proposition 2. *The rank of $F_{\mathcal{P}}$ is equal to the degree of the LCM of the set $\mathcal{P}_{h,n}$.*

$$\rho(F_{\mathcal{P}}) = \deg\{\mathrm{lcm}\{\mathcal{P}_{h,n}\}\} \tag{52}$$

Proof. We consider again the LCM of the set $\mathcal{P}_{h,n}$ in its generic form $l(s)$ as defined in (3). The linear system (51) has d equations and $d+1$ variables. Hence, it is an underdetermined homogeneous linear system. If $n(F_{\mathcal{P}})$ denotes the nullity of $F_{\mathcal{P}}$ and $\rho(F_{\mathcal{P}})$ its rank, then $n(F_{\mathcal{P}}) \geq 1$ which implies that there are infinite solutions.

a) If $n(F_{\mathcal{P}}) = 1$ only one solution (up to scalar multiples) is obtained and this is actually the generic solution:

$$l(s) = p_1(s) \cdot p_2(s) \cdots p_h(s)$$

Then,

$$\left.\begin{array}{l} \deg\{l(s)\} = \sum_{i=1}^{h} d_i = d \\ \rho(F_{\mathcal{P}}) = d + 1 - n(F_{\mathcal{P}}) = d \end{array}\right\} \Rightarrow \rho(F_{\mathcal{P}}) = \deg\{l(s)\} = d$$

b) If $n(F_{\mathcal{P}}) = \nu > 1$, we can set exactly ν free variables. However, a careful examination of the structure of $F_{\mathcal{P}}$ (eq. 48) reveals that each column corresponds to a coefficient a_j of the polynomial $l(s)$ which has a fixed position according to the power basis vector $\underline{e}_d = [s^d, \ldots, s, 1]^t$. Thus, the last $d+1-\nu$ columns, which correspond to a_j for $j = 0, 1, \ldots, d-\nu$, should lead to the trivial solution. Assuming that $l(s)$ is a monic polynomial, the least degree solution will be obtained if we set

$$a_{d-\nu+1} = 1 \;\text{ and }\; a_{d-\nu+2} = \ldots = a_d = 0$$

Then,

$$\left.\begin{array}{l} \deg\{l(s)\} = d - \nu + 1 \\ \rho(F_{\mathcal{P}}) = d + 1 - n(F_{\mathcal{P}}) = d + 1 - \nu \end{array}\right\} \Rightarrow \rho(F_{\mathcal{P}}) = \deg\{l(s)\} = d + 1 - \nu$$

□

Remark 5. Given a small tolerance $\varepsilon > 0$, the numerical ε-rank of $F_{\mathcal{P}}$ determines the degree of the ALCM of the set $\mathcal{P}_{h,n}$.

The preceding analysis leads to the following result.

Theorem 2. *Given a set of several polynomials $\mathcal{P}_{h,n}$, the corresponding LCM denoted by $l(s) = \sum_{j=0}^{d} a_j\, s^j$ with $d = \sum_{i=1}^{h} \deg\{p_i(s)\}$ is given by the least degree solution of the underdetermined linear system*

$$F_{\mathcal{P}} \cdot \underline{a} = \underline{0} \tag{53}$$

where $F_{\mathcal{P}} \in \mathbb{R}^{d \times (d+1)}$ is associated with the remainders $r_i(s)$ which derive from the application of the ERES division algorithm to all the pairs $(l(s),\, p_i(s))$, and $\underline{a} = [a_d, \ldots, a_0]^t$ is the vector of coefficients of the LCM.

Proof. Let ρ denotes the rank of $F_{\mathcal{P}}$. From Proposition 2 we have:

$$\rho = \rho(F_{\mathcal{P}}) = \deg\{l(s)\}$$

Therefore, since the columns of $F_{\mathcal{P}}$ correspond to the coefficients of $l(s)$ in a fixed order, the $d \times (d+1)$ linear system (53) can be reduced to a $d \times \rho$ linear system, such that

$$\check{F}_{\mathcal{P}} \cdot \underline{\check{a}} + \underline{f}_{d-\rho+1} \cdot a_\rho + \widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} = \underline{0} \quad \Leftrightarrow \quad \widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} = -a_\rho\, \underline{f}_{d-\rho+1} \tag{54}$$

where

- the matrix $\check{F}_{\mathcal{P}}$ is constructed from the first $d - \rho$ columns of $F_{\mathcal{P}}$,
- the matrix $\widehat{F}_{\mathcal{P}}$ is constructed from the last ρ columns of $F_{\mathcal{P}}$,
- $\underline{f}_{d-\rho+1}$ is the $d - \rho + 1$ column of $F_{\mathcal{P}}$ (counting from left to right),
- $\underline{\check{a}} = [a_d, \ldots, a_{\rho+1}]^t$ is the vector of the coefficients of $l(s)$ which can be omitted (or set equal to 0),
- $\underline{\hat{a}} = [a_{\rho-1}, \ldots, a_0]^t$ is the vector of the r coefficients of $l(s)$ which form the solution, and
- $a_\rho \neq 0$ which corresponds to the leading coefficient of the LCM. If $c_{d_i}^{(i)}$ denotes the nonzero leading coefficients of the polynomials $p_i(s)$, as defined in (2), then a_ρ may take on the following values:

  $$a_\rho = \begin{cases} 1, \text{ if all the polynomials } p_i(s) \text{ are monic, i.e. } c_{d_i}^{(i)} = 1,\ i = 1..h \\ \operatorname{lcm}\left\{ c_{d_i}^{(i)}, i = 1..h \right\}, \text{ if } c_{d_i}^{(i)} \text{ are all integer numbers.} \\ \prod_{i=1}^{h} c_{d_i}^{(i)}, \text{ if } c_{d_i}^{(i)} \text{ are real numbers.} \end{cases} \tag{55}$$

  The last form of the coefficient a_ρ as a product of the leading coefficients of the given polynomials is particularly used for computing the ALCM.

Consequently, the full-rank linear system (54) has a unique solution which is the least degree solution of the system (53) and provides the LCM of the given set of polynomials $\mathcal{P}_{h,n}$. □

The next example illustrates the procedure for the algebraic computation of the LCM of sets of more than two polynomials which derives from Theorem 2.

Example 3. We consider the set of $h = 3$ polynomials:

$$\mathcal{P}_{h,n} = \Big\{ p_i(s) \in \mathbb{R}[s],\ i = 1, 2, 3\ ,\ d_i = \{3, 3, 2\},\ n = 3 \Big\}$$

with

$$\begin{array}{l} p_1(s) = (2\,s - 3)(s + 1)^2 \\ p_2(s) = (3\,s + 3)(s - 2)^2 \\ p_3(s) = (s + 1)(s - 2) \end{array} \quad \text{and} \quad P = \begin{bmatrix} 2 & 1 & -4 & -3 \\ 3 & -9 & 0 & 12 \\ 0 & 1 & -1 & -2 \end{bmatrix}$$

The current procedure for the LCM computation requires the following steps:

1. Assuming that the actual degree of the LCM is unknown, we represent the LCM in terms of its generic degree $d = 3 + 3 + 2 = 8$ using arbitrary coefficients $a_j,\ j = 0, 1, \dots, 8$.

$$l(s) = a_8\, s^8 + a_7\, s^7 + a_6\, s^6 + a_5\, s^5 + a_4\, s^4 + a_3\, s^3 + a_2\, s^2 + a_1\, s^1 + a_0\, s^0$$

2. We can apply the ERES division algorithm to every pair $(l(s), p_i(s))$ and form a sequence of remainder vectors $\underline{r}_i$ in symbolic-rational format.

3. Using the obtained remainder vectors, we create the matrices F_i with respect to the vector $\underline{a} = [a_8, \dots, a_1, a_0]^t$ as described in (45)–(48). Finally, we construct the 8×9 matrix:

$$F_{\mathcal{P}} = \begin{bmatrix} \frac{457}{64} & \frac{3}{32} & \frac{65}{16} & -\frac{5}{8} & \frac{9}{4} & -\frac{1}{2} & 1 & 0 & 0 \\ \frac{201}{32} & \frac{115}{16} & \frac{17}{8} & \frac{15}{4} & \frac{1}{2} & 2 & 0 & 1 & 0 \\ \frac{9}{64} & \frac{195}{32} & -\frac{15}{16} & \frac{27}{8} & -\frac{3}{4} & \frac{3}{2} & 0 & 0 & 1 \\ 313 & 135 & 57 & 23 & 9 & 3 & 1 & 0 & 0 \\ -228 & -92 & -36 & -12 & -4 & 0 & 0 & 1 & 0 \\ -540 & -228 & -92 & -36 & -12 & -4 & 0 & 0 & 1 \\ 85 & 43 & 21 & 11 & 5 & 3 & 1 & 1 & 0 \\ 86 & 42 & 22 & 10 & 6 & 2 & 2 & 0 & 1 \end{bmatrix}$$

which forms the homogeneous linear system (53).

4. Since $\rho = \rho(F_{\mathcal{P}}) = 5$, the degree of the LCM will be equal to 5 and we obtain the next reduced linear system:

$$\widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} = -a_5\, \underline{f}_4$$

where the matrix $\widehat{F}_{\mathcal{P}} \in \mathbb{R}^{8 \times 5}$ is constructed from the last 5 columns of $F_{\mathcal{P}}$, $\underline{f}_4$ is the 4^{th} column of $F_{\mathcal{P}}$, and $\underline{\hat{a}} = [a_4, , \dots, a_0]^t$. Moreover, $a_j = 0$ for $j = 6, 7, 8$ and, since not all the polynomials of the given set $\mathcal{P}_{h,n}$ are monic and their leading coefficients are prime numbers, the leading coefficient of the LCM will be $a_5 = 2 \cdot 3 \cdot 1 = 6$. Finally, we end up with the overdetermined linear system:

$$\begin{bmatrix} \frac{9}{4} & -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 2 & 0 & 1 & 0 \\ -\frac{3}{4} & \frac{3}{2} & 0 & 0 & 1 \\ 9 & 3 & 1 & 0 & 0 \\ -4 & 0 & 0 & 1 & 0 \\ -12 & -4 & 0 & 0 & 1 \\ 5 & 3 & 1 & 1 & 0 \\ 6 & 2 & 2 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} -\frac{5}{8} \\ \frac{15}{4} \\ \frac{27}{8} \\ 23 \\ -12 \\ -36 \\ 11 \\ 10 \end{bmatrix}$$

Since the matrix $\widehat{F}_{\mathcal{P}}$ has full rank, the above linear system has a unique solution which can be computed using standard algebraic methods, such as LU decomposition or Gaussian elimination [17, 15]. The obtained solution is

$$\underline{\hat{a}} = [-21, 0, 51, -12, -36]^t$$

and corresponds to the exact LCM of the given set $\mathcal{P}_{h,n}$, which is:

$$\text{lcm}\{\mathcal{P}_{h,n}\} = l(s) = 6\,s^5 - 21\,s^4 + 51\,s^2 - 12\,s - 36$$

5. The H-LCM method and its hybrid implementation

The results derived from Theorem 2 lead to the formulation of a computational method for the ALCM which is free of GCD computations or polynomial factorization. In order to obtain the "best" possible solution in a sense of a meaningful approximation, exact symbolic and numerical computations are combined together (hybrid computations) in order to formulate the *Hybrid LCM method* (H-LCM) for the approximate computation of the least common multiple of set of several univariate polynomials.

5.1. The development of the H-LCM method

The H-LCM method requires careful consideration especially when numerical inaccuracies are present and an approximate solution is desired. The method has two main parts with the following tasks:

P1 Divide the generic LCM by the given polynomials and create a sequence of remainders. Then process the remainder vectors so as to obtain a linear system for the computation of the ALCM.

P2 Solve the linear system to compute the ALCM.

In the context of a symbolic-numeric implementation, the above problems can be handled as follows.

P1: Assuming the generic degree, the ALCM is expressed as a polynomial with symbolic coefficients. Then, the ERES division algorithm can be employed in order to provide the general algebraic (symbolic) form of the remainder vectors $\underline{r}_i$ which correspond to the division of the generic ALCM by the given polynomials. The implementation of this part of the algorithm using symbolic-rational operations provides the means to form the special matrix $F_{\mathcal{P}}$ which will be used to estimate the degree and the coefficients of the ALCM.

P2: The second part of the H-LCM algorithm involves: i) the construction of the matrix $F_{\mathcal{P}}$ from the remainder vectors $\underline{r}_i$, ii) the determination of the rank of $F_{\mathcal{P}}$, and iii) the solution of the linear system (54). In the approximate case the solution of the linear system (54) can be computed using a low-rank approximation technique and numerical optimization tools.

5.2. The H-LCM algorithm

Input : $\mathcal{P}_{h,n} = \Big\{ p_i(s) \in \mathbb{R}[s],\ i = 1..h \text{ with } d_i = \deg\{p_i\} > 0 \Big\}$

Step 1: Set $d = \sum_{i=1}^{h} d_i$ and $l(s) = \sum_{j=0}^{d} a_j\, s^j$ using coefficients a_j in symbolic form.

Step 2: For $i = 1..h$, apply the ERES division algorithm to the pairs of polynomials $\big(l(s), p_i(s)\big)$ to obtain the remainder vectors $\underline{r}_i$ in symbolic-rational form.

Step 3: For $i = 1..h$, construct the matrix $F_{\mathcal{P}}$ using the vectors $\underline{r}_i$ as described in (7) – (51) and compute its rank $\rho = \rho(F_{\mathcal{P}})$.

Step 4: Form the $d \times \rho$ linear system $\widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} = -a_\rho\, \underline{f}_{d-\rho+1}$ and solve this system using numerical computations.

Output: $\mathrm{lcm}\{\mathcal{P}\} \triangleq l(s) = a_\rho\, s^\rho + \underline{\hat{a}}^t \cdot \underline{e}_{\rho-1}(s)$

5.3. The numerical computation of an ALCM using the H-LCM algorithm

The critical step in the second part of the H-LCM algorithm is the computation of the numerical rank of $F_{\mathcal{P}}$ which determines the degree of the ALCM (Remark 5). Considering a small tolerance $\varepsilon > 0$, the numerical ε-rank of the matrix $F_{\mathcal{P}}$ can be computed using singular value decomposition (SVD) [17].

Therefore, using Theorem 2 in the approximate case, we aim to solve the $d \times (d - \rho)$ linear system

$$\widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} \approx -a_\rho\, \underline{f}_{d-\rho+1} \tag{56}$$

where now r is the ε-rank of the matrix $\widehat{F}_{\mathcal{P}}$ which is also equal to the ε-rank of the matrix $F_{\mathcal{P}}$. We actually seek a solution, such that:

$$\left\| \widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} + a_\rho\, \underline{f}_{d-\rho+1} \right\| = \text{minimum}$$

If we use the Euclidean norm $\|\cdot\|_2$, the latter implies a *least-squares* solution [17] for the linear system (56) and, since it is a full-rank overdetermined linear system, it has a unique least-squares solution [17] which can be represented as

$$\underline{\hat{a}} = \widehat{F}_{\mathcal{P}}^{\dagger} \cdot (-a_\rho\, \underline{f}_{d-\rho+1}) \tag{57}$$

where $\widehat{F}_{\mathcal{P}}^{\dagger}$ is the pseudo-inverse of $\widehat{F}_{\mathcal{P}}$.

The main result for the numerical computation of the ALCM is summarized in the next theorem.

Theorem 3. *Let $\mathcal{P}_{h,n}$ a set of real univariate polynomials, as defined by (1) and a small specified tolerance $\varepsilon > 0$. An ALCM of the set $\mathcal{P}_{h,n}$ is given by the solution of the least-squares problem*

$$\mathcal{L} \triangleq \min_{\underline{\hat{a}}} \left\| \widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} - \big(- a_{\rho} \, \underline{f}_{d-\rho+1} \big) \right\|_2 \tag{58}$$

where ρ is the numerical ε-rank of the matrix $\widehat{F}_{\mathcal{P}}$, a_{ρ} is the leading coefficient of the ALCM, and $\underline{\hat{a}}$ is the vector of the remaining $\rho - 1$ coefficients of the ALCM.

The residual from the solution of the linear least-squares problem (58) characterises the numerical quality of the obtained ALCM and the unique solution of the least-squares problem (58) can be computed using either the QR, or the SVD least-squares methods [15]. In fact, several mathematical software packages, such as Matlab and Maple, include efficient built-in routines for the linear least-squares problem where QR decomposition or SVD based algorithms are used.

5.4. Error analysis of the H-LCM algorithm

In order to study the overall stability of the proposed H-LCM method according to the backward error analysis concepts we first prove that the matrix formed by the end of the process P1 is actually a slight variation of $F_{\mathcal{P}}$, that is $F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$, where the difference $\Delta F_{\mathcal{P}}$ is appropriately bounded. Then, we apply standard error analysis results for the corresponding least-squares problem.

5.4.1. Backward error analysis for the formulation of the matrix $F_{\mathcal{P}}$

Given a set $\mathcal{P}_{h,n}$ with numeric coefficients (floating-point numbers), in the first part of the algorithm, P1, we may assume an error-free conversion from the numeric to a rational form (fractions of integers). The application of the ERES division algorithm to the polynomials of the original set $\mathcal{P}_{h,n}$ involves symbolic-rational operations which are used to construct the initial matrix $F_{\mathcal{P}}$ for the ALCM computation and do not introduce numerical errors during the processing of the data. The general form of the elements of this matrix is described in (46). Conversely, in the beginning of the second part of the algorithm, P2, the necessary conversion from the rational to a numeric form may lead to a small loss of accuracy which depends on the number of digits of the variable floating-point arithmetic controlled by the user.

Hence, assuming that the input data may contain uncertainties due to measurements, previous computations, or errors committed in storing numbers on the computer, we shall consider a perturbed set of polynomials $\widetilde{\mathcal{P}}_{h,n}$, which also follows the general backward error concept [15, 17]. Therefore, for each polynomial $p_i(s)$, we regard a perturbed polynomial $\tilde{p}_i(s)$ as a slight variation of the initial $p_i(s)$, such that

$$\tilde{p}_i(s) = \sum_{j=0}^{d_i} \left(c_j^{(i)} + \epsilon_i \right) s^j \ , \ \forall \ i = 1..h \tag{59}$$

where $\epsilon_i = O(\mathbf{u})$ and $\mathbf{u}$ denotes the machine-precision ($\mathbf{u} = 2^{-52}$ in 16-digits arithmetic precision). In the following, we assume a uniform perturbation $\epsilon = \max_{i=1..h}(\epsilon_i)$.

As described in Section 4, the construction of the initial matrix $F_{\mathcal{P}}$ involves the computation of h smaller matrices F_i which are associated with the remainder vectors $\underline{r}_i$ in (47), and thus, with the polynomials $p_i(s)$.

Definition 5. For every polynomial $p_i(s)$ in $\mathcal{P}_{h,n}$ a parameter $\gamma_i > 0$, such that

$$\gamma_i = \frac{\max_{j=0..d_i}\left(|c_j^{(i)}|\right)}{|c_{d_i}^{(i)}|} \ , \ \forall\, i = 1..h \tag{60}$$

characterizes the elements of the matrix F_i for every $i = 1..h$. Then, a uniform parameter for all F_i is defined by

$$\gamma = \max_{i=1..h}(\gamma_i) \tag{61}$$

Remark 6. The norm of F_i computed without taking into account the perturbation ϵ is:

$$\|F_i\|_F = \sqrt{d_i}\, O\left(\gamma_i^{d-d_i+1}\right)$$

For instance, considering the polynomials in Example 2 with $d = 5$ and $d_i = 3$ we get:

$$\|F_i\|_F \leq \left(3\,{\gamma_i}^6 - 10\,{\gamma_i}^5 + 14\,{\gamma_i}^4 - 8\,{\gamma_i}^3 + 6\,{\gamma_i}^2 + 3\right)^{\frac{1}{2}} = \sqrt{3}\, O\left(\gamma_i^3\right)$$

Assuming the same degree n for all the h polynomials of the given set $\mathcal{P}_{h,n}$, we have $d = h\,n$. Consequently,

$$\|F_{\mathcal{P}}\|_F \leq h\,\sqrt{n}\, O\left(\gamma^{d-n+1}\right) \tag{62}$$

Proposition 3. *The application of the ERES division to the perturbed set $\widetilde{\mathcal{P}}_{h,n}$ leads to a matrix of the form $F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$, represented as:*

$$ERESdiv\Big(l(s), \widetilde{\mathcal{P}}_{h,n}\Big) = F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$$

The matrix $\Delta F_{\mathcal{P}}$ represents the total accumulated error and it holds:

$$\|\Delta F_{\mathcal{P}}\|_F \leq (hn - n + 1)\, h\,\sqrt{n}\,\epsilon\, O\left(\gamma^{hn-n}\right) + O(\epsilon^2) \tag{63}$$

Proof. Let $F_i + \Delta F_i$ the computed matrix associated with each remainder vector $\underline{r}_i$, where ΔF_i represents the accumulated error after the application of the ERES division. This process can be described as follows:

$$ERESdiv\Big(l(s), \tilde{p}_i(s)\Big) = F_i + \Delta F_i$$

Considering (46), (48), and (60), we obtain

$$\|\Delta F_i\|_F \leq (d - d_i + 1)\,\sqrt{d_i}\,\frac{\epsilon}{\gamma_i}\, O\left(\gamma_i^{d-d_i+1}\right) + O(\epsilon^2) \tag{64}$$

which can be extended to (63) using (49), (61), and (62). □

5.4.2. Backward error analysis for the least-squares solution

In order to compute the rank of the matrix $F_{\mathcal{P}}$ and determine the degree of the ALCM more accurately, it is preferable to apply the SVD algorithm to a normalized copy of $F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$, such that all its elements be less than 1 in absolute value [17]. This process is an elementary row transformation which provides better numerical stability and does not affect the properties of the system (53) and the final solution. The normalization of the rows using the Euclidean norm in floating-point arithmetic with unit round-off $\mathbf{u}$ satisfies [15] the relations

$$\bar{F}_{\mathcal{P}} = N \cdot (F_{\mathcal{P}} + \Delta F_{\mathcal{P}}) + E_N$$
$$\|E_N\|_2 \leq d\,\mathbf{u}\,\|F_{\mathcal{P}} + \Delta F_{\mathcal{P}}\|_2 + O(\mathbf{u}^2) \tag{65}$$

where $N \in \mathbb{R}^{d\times d}$ is a diagonal matrix accounting for the performed transformations with $\|N\|_2 \leq \sqrt{d}$, and $E_N \in \mathbb{R}^{d\times(d+1)}$ is the obtained error.

Therefore, if ρ is the numerical rank of $\bar{F}_{\mathcal{P}}$, then the matrix $\widehat{F}_{\mathcal{P}}$ in the least-squares problem (58) is actually formed from the last ρ columns of $F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$ and the above error in (65) is not added to the system. We only have that

$$\rho \triangleq \rho(\widehat{F}_{\mathcal{P}}) = \rho(\bar{F}_{\mathcal{P}})$$

and, if we denote by

$$\widetilde{F}_{\mathcal{P}} = \widehat{F}_{\mathcal{P}} + \Delta\widehat{F}_{\mathcal{P}} \ \text{ and } \ \underline{\tilde{f}}_{d-\rho+1} = \underline{f}_{d-\rho+1} + \Delta\underline{f}_{d-\rho+1}$$

the $d \times \rho$ submatrix and the $d - \rho + 1$ column of the matrix $F_{\mathcal{P}} + \Delta F_{\mathcal{P}}$, respectively, which are necessary for the ALCM computation, then, due to the specific ordered structure of the matrix $F_{\mathcal{P}}$ and considering the results obtained from the preceding analysis, it holds that:

$$\left\|\Delta\widehat{F}_{\mathcal{P}}\right\|_F \leq (\rho - n + 1)\,\epsilon\,\left\|\widehat{F}_{\mathcal{P}}\right\|_F \ \text{ and } \ \left\|\Delta\underline{f}_{d-\rho+1}\right\|_2 \leq \epsilon\,\left\|\underline{f}_{d-\rho+1}\right\|_2 \tag{66}$$

where

$$\left\|\widehat{F}_{\mathcal{P}}\right\|_F \leq h\sqrt{n}\,O\left(\gamma^{\rho-n}\right) \ \text{ and } \ \left\|\underline{f}_{d-\rho+1}\right\|_2 \leq h\sqrt{n}\,O\left(\gamma^{\rho}\right) \tag{67}$$

Remark 7. The errors $\Delta F_{\mathcal{P}}$ and $\Delta\widetilde{F}_{\mathcal{P}}$ involve the parameter γ, as defined by (61), which refers to the magnitude of the coefficients of the given polynomials $p_i(s)$. In general, we expect a small error when $\gamma \leq 1$. Otherwise, we may consider an initial scaling for the polynomials, so that $\|\underline{p}_i\|_2 \leq \sqrt{d_i}$, $i = 1..h$.

Different methods lead to a variety of estimates for the optimal size of backward errors for least-squares problems [15, 17, 18]. Based on the QR method for the full-rank least-squares problem (Golub-Householder method [17]), the computed solution $\underline{\hat{a}}$ is such that it satisfies the condition:

$$\left\|\left(\widetilde{F}_{\mathcal{P}} + \Delta F\right)\underline{\hat{a}} - \left(-a_\rho\,\underline{\tilde{f}}_{d-\rho+1} + \Delta f\right)\right\|_2 = \text{minimum}$$

where $(\Delta F, \Delta f)$ is the inherited rounding error perturbation. It has been shown in [19] that the normwise backward error of this kind of problem is small. More specifically,

$$\|\Delta F\|_F \leq (6h - 3\rho + 41)\,\rho\,\mathbf{u}\,\|\widetilde{F}_{\mathcal{P}}\|_F + O(\mathbf{u}^2) \ \text{ and } \tag{68}$$

$$\|\Delta f\|_2 \leq (6h - 3\rho + 41)\,\rho\,\mathbf{u}\,|a_\rho|\,\left\|\underline{\tilde{f}}_{d-\rho+1}\right\|_2 + O(\mathbf{u}^2) \tag{69}$$

The preceding analysis leads to the following theorem which provides a backward error for the ALCM computation.

Theorem 4. *The computed ALCM solution* $\hat{\underline{a}} = [0, \ldots, 0, a_\rho, a_{\rho-1}, \ldots, a_0]^t$ *obtained from the H-LCM algorithm, is the exact solution of the nearby problem*

$$(F_{\mathcal{P}} + E)\,\underline{a} = \underline{0} \tag{70}$$

where

$$\|E\|_F \leq \left\|\Delta \widehat{F}_{\mathcal{P}}\right\|_F + \|\Delta F\|_F + |a_\rho| \left(\left\|\Delta \underline{f}_{\,d-\rho+1}\right\|_2 + \|\Delta f\|_2 \right) \tag{71}$$

5.5. Computational complexity of the H-LCM algorithm

Assuming that all the polynomials $p_i(s) \in \mathcal{P}_{h,n}$ have the same degree n (i.e. $d_i = n$ for all $i = 1..h$), then $d = h\,n$. The number of operations required for the ERES division algorithm is

$$fl_{ED}(h, n) = h(n+2)(d-n+1) \leq (h\,n)^3 \tag{72}$$

It can easily be proven that the above inequality holds for every n, $h \geq 2$. However, since the ERES division involves operations not only with rational numbers but also with symbolic parameters, the overall complexity mostly depends on how these data are stored and processed by the software. The current procedures have been implemented using Maple 18 which is a mathematical software optimized for symbolic-rational computations. Using appropriate interpolation techniques, several tests with various sets $\mathcal{P}_{h,n}$ have shown that the time complexity for the first part, P1, of the H-LCM algorithm, which involves h calls of the ERES division algorithm, can be approximately given by the function

$$T(\kappa) = O\left(\kappa^3 \log(\kappa)\right), \quad \kappa = hn - n + 1 \tag{73}$$

where κ represents the number of the *ERES iterations* of the main procedure of the ERES division algorithm for two polynomials of degrees $m = hn$ and n (see Remark 4 and Figure 1).

As an example of the performance of the ERES division algorithm, Figure 2 illustrates the time complexity of the algorithm for $h \geq 2$ polynomials with randomly selected 2-digit integer coefficients and uniform degree $n \geq 2$. The diagram on the left shows the increase of the processing time for sets $\mathcal{P}_{h,n}$ with fixed number of polynomials $h = 4$ when $n = 2, 3, \ldots, 100$. The diagram on the right shows the increase of the processing time for sets $\mathcal{P}_{h,n}$ with fixed polynomial degree $n = 4$ when $h = 2, 3, \ldots, 100$.

In the second part, P2, of the H-LCM algorithm, the numerical rank of the $d \times (d+1)$ matrix $F_{\mathcal{P}}$ can be determined by the magnitude of its singular values. Thus, additional operations are required for the normalization and the estimation of the numerical rank of $F_{\mathcal{P}}$. Conclusively, if the Golub-Householder method [17] is used for the full-rank least-squares problem, the total amount of operations required for the H-LCM algorithm will be

$$O\left(\frac{4}{3}\,(n\,h)^3 + \frac{3}{2}\,(n\,h)\rho^2 + 5\,\rho^3\right) \tag{74}$$

In the worst case scenario where the LCM has the maximum degree $d = n\,h = \rho$, the algorithm performs less than $8\,d^3$ operations for the final result.

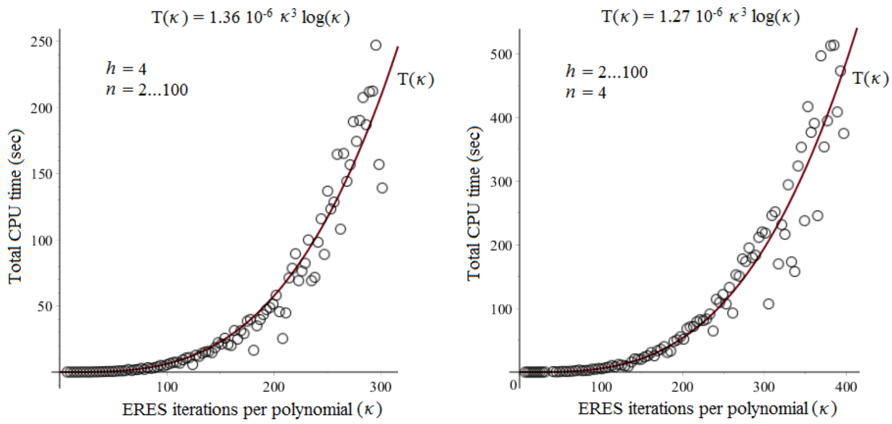

Figure 2: H-LCM time complexity: Construction of the matrix $F_{\mathcal{P}}$

6. Numerical examples and comparison with other methods

The LCM problem has also been addressed in [6] and for comparison reasons the provided method was reformulated as described next.

6.1. The symbolic-rational LCM algorithm

The approach followed in [6] was based on the reduction of the computation of the LCM to an equivalent problem where the computation of GCD is an integral part. The essence of this procedure is that if $p(s)$ denotes the product of the polynomials of the original set and $g(s)$ the GCD of a special set of polynomials derived from the original set, then the LCM, $l(s)$, can be computed using the factorisation $p(s) = g(s)\, l(s)$. The use of algorithms for computing the GCD are important for the particular LCM method. Naturally, for approximate values of the GCD the order of approximation is defined as a factor of $p(s)$ and the computation of the approximate LCM is then seen as the best way of completing the approximate factorisation, which is defined as the *optimal completion problem* [6].

The following algorithm is developed in the context of symbolic-rational computations for the estimation of the ALCM of a set of polynomials $\mathcal{P}$ and it is based on the results presented in [6]. The next *Symbolic-Rational (SR) LCM algorithm* is actually a variation of the numerical LCM algorithm which has been developed in [6].

The SR-LCM Algorithm.

Input : $\mathcal{P} = \left\{p_i(s) \in \mathbb{R}[s],\ i = 1..h\right\}$

Step 1 : Compute $p(s) = p_1(s)\, p_2(s) \cdots p_h(s)$.

Step 2 : Find the set $\mathcal{T} = \left\{p_{\omega_i}(s) :\ p_{\omega_i}(s) = \prod_{k=1}^{h-1} p_{i_k}(s),\ i = 1, \ldots, h\right\}$ for all $\omega_i = (i_1, i_2, \ldots, i_{h-1}) \in Q_{h-1,h}$.

Step 3 : Compute $g(s) = \gcd\{\mathcal{T}\}$.

Step 4 : Compute $l(s) = \dfrac{p(s)}{g(s)}$ by applying the ERES division algorithm to the pair $(p(s), g(s))$.

Output : $\operatorname{lcm}\{\mathcal{P}\} = l(s)$

Notation 1. Let $Q_{\mu,\nu}$ be the ordered set of lexicographically ordered sequences of μ integers from ν. We shall denote by $\omega = (i_1, i_2, \ldots, i_{h-1}) \in Q_{h-1,h}$ and $\widehat{\omega} = (j)$ is the index from $(1..h)$ which is complementary to the set of indices in ω.

The above SR-LCM algorithm is very effective when the polynomials have integer coefficients and are processed by using exact rational operations. However, the amount of the required arithmetic operations for the initial polynomial $p(s)$ and the polynomials of the set $\mathcal{T}$ can be prohibitively high. Specifically, for h polynomials with average degree $\bar{d} \geq 2$ the algorithm performs $O\left((\bar{d}+1)^h\right)$ operations. More operations are required for the GCD computation and the ERES division. Several tests have shown that the SR-LCM algorithm can be computationally efficient only for moderate sets of polynomials.

Regarding its numerical efficiency, if the original data are inaccurate, the construction of the polynomials of the set $\mathcal{T}$ and the ERES division algorithm can be implemented using symbolic-rational operations in order to minimize the risk of obtaining erroneous results due to the excessive accumulation of rounding errors. Therefore, the computation of an ALCM with the SR-LCM algorithm relies on the approximate computation of the GCD. There are several effective methods for the computation of an approximate GCD for univariate polynomials [20, 21, 22, 23, 24, 25] which can be integrated with the SR-LCM algorithm, but in the following tests we used the H-ERES algorithm [16].

6.2. Computational examples

The following examples illustrate the basic characteristics of the H-LCM algorithm and a comparison with the methods developed in [6] and [10] is provided.

Example 4. Consider the set of three integer polynomials:

$$\mathcal{P} = \left\{ \begin{array}{l} p_1(s) = (s+1)(s+2+\epsilon)^2 \\ p_2(s) = (s+2)(s+3)(s+4+\epsilon) \\ p_3(s) = (s+4)^2(s+5) \end{array} \right\} \tag{75}$$

Two of the three polynomials of the given set $\mathcal{P}$ contain a small perturbation ϵ. If $\epsilon = 0$, the exact LCM is

$$\begin{aligned} \operatorname{lcm}\{\mathcal{P}\} &= (s+1)(s+2)^2(s+3)(s+4)^2(s+5) \\ &= s^7 + 21\,s^6 + 183\,s^5 + 855\,s^4 + 2304\,s^3 + 3564\,s^2 + 2912\,s + 960 \end{aligned}$$

The maximum theoretical degree of the LCM is $d = 9$ and, using the H-LCM algorithm, a 9×10 matrix $F_{\mathcal{P}}$ is constructed for the main linear system (53):

$$F_{\mathcal{P}} = \begin{bmatrix} -1793 & 769 & -321 & 129 & -49 & 17 & -5 & 1 & 0 & 0 \\ -4868 & 2052 & -836 & 324 & -116 & 36 & -8 & 0 & 1 & 0 \\ -3076 & 1284 & -516 & 196 & -68 & 20 & -4 & 0 & 0 & 1 \\ -111645 & 26335 & -6069 & 1351 & -285 & 55 & -9 & 1 & 0 & 0 \\ -539054 & 125370 & -28286 & 6090 & -1214 & 210 & -26 & 0 & 1 & 0 \\ -632040 & 145656 & -32424 & 6840 & -1320 & 216 & -24 & 0 & 0 & 1 \\ -1101157 & 194017 & -33069 & 5385 & -821 & 113 & -13 & 1 & 0 & 0 \\ -8219432 & 1421064 & -235880 & 36936 & -5288 & 648 & -56 & 0 & 1 & 0 \\ -15521360 & 2645520 & -430800 & 65680 & -9040 & 1040 & -80 & 0 & 0 & 1 \end{bmatrix}$$

Then, the rows of $F_{\mathcal{P}}$ are normalized by using the Euclidean norm and its numerical rank is computed. If $\varepsilon = \mathbf{u}\,\|F_{\mathcal{P}}\|_F \approx 10^{-16}$, the diagram of the singular values of the normalized $F_{\mathcal{P}}$, denoted by $\bar{F}_{\mathcal{P}}$, in Figure 3a shows a sudden drop from the 7^{th} to the 8^{th} singular value, and $\sigma_8 < \varepsilon$. Therefore, in regular double precision arithmetic (16-digits) the numerical ε-rank of the matrix $\bar{F}_{\mathcal{P}}$ is $\rho = 7$. Hence, the aim is to find an ALCM with degree equal to 7.

If $\widehat{F}_{\mathcal{P}}$ is the 9×7 matrix which derives from $F_{\mathcal{P}}$ by deleting its first two columns, $a_\rho = a_7 = 1$ since the original polynomials are monic, and $\underline{f}_{d-\rho+1} = \underline{f}_3$ is the 3^{rd} column of $F_{\mathcal{P}}$, then the least-squares problem (58) provides the solution. We applied three different least-squares methods based on: a) the QR factorization (LS-QR), b) the singular value decomposition (LS-SVD), and c) the pseudo-inverse of $\widehat{F}_{\mathcal{P}}$ (LS-PInv), [15]. The quality of the obtained solution is measured regarding the magnitude of the residual and the relative error,

$$\|u\|_2 = \left\| \widehat{F}_{\mathcal{P}} \cdot \underline{\hat{a}} + \underline{f}_3 \right\|_2 \quad \text{and} \quad \text{Rel} = \frac{\|\underline{\hat{a}} - \underline{a}\|_2}{\|\underline{a}\|_2}$$

respectively, when the exact solution $\underline{a}$ is known. The results, which are presented in Table 1, show that the QR-Least-Squares method produces a more accurate solution compared to the other two least-squares methods.

	LS-QR	LS-SVD	LS-PInv
Residual	$6.255761 \cdot 10^{-15}$	$1.533661 \cdot 10^{-11}$	$8.876500 \cdot 10^{-12}$
Relative error	$4.641785 \cdot 10^{-13}$	$1.535942 \cdot 10^{-11}$	$1.405633 \cdot 10^{-11}$

Table 1: Numerical results for the LCM of the set (75) with $\epsilon = 0$ given by different least-squares methods.

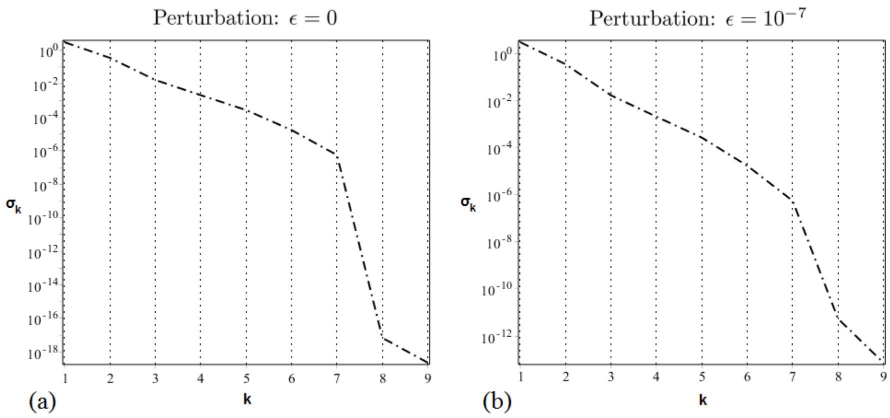

Figure 3: Singular values diagram for the matrix $\bar{F}_{\mathcal{P}}$ associated with the set (75).

The set $\mathcal{P}$ in (75) is considered again while adding a small perturbation $\epsilon = 10^{-7}$. Considering this perturbation in the data, the diagram in Figure 3b illustrates the magnitude of the singular values of the matrix $\bar{F}_{\mathcal{P}}$. The numerical rank of the 9×10 input matrix $\bar{F}_{\mathcal{P}}$, computed in regular double precision, is

9 as expected ($\sigma_9 > 10^{-16}$). In this case the degree of the exact LCM of the set $\mathcal{P}$ becomes equal to 9 which is the generic degree. However, if a numerical tolerance $\varepsilon = 10^{-8} = 0.1\,\epsilon$ is set, then the rank drops to 7. Thus, the degree of the produced approximate ε-LCM is equal to 7.

If we try to compute an ALCM using the SR-LCM algorithm, we can just verify that the LCM of the set $\mathcal{P}$ is a polynomial with degree 9, which implies that the associated GCD, $g(s)$, is equal to 1. However, for a numerical tolerance $\varepsilon = 10^{-4}$ the H-ERES algorithm for GCDs [16] is able to produce a non trivial GCD of degree 2. If we denote by $l_1(s)$ the approximate solution given by the H-LCM algorithm and $l_2(s)$ the approximate solution given by the SR-LCM algorithm, then the distance between those two approximations is

$$\|l_1(s) - l_2(s)\|_2 = 0.4057382668$$

which is relatively large. Therefore, judging from the tolerance ε, the relative error, and the residual in Table 2, the approximation given by the H-LCM algorithm is far better than the one given by the SR-LCM algorithm.

	ε	Degree	Rel. Error / Residual
SR-LCM	10^{-4}	7	$1.296138 \cdot 10^{-2}$
H-LCM	10^{-8}	7	$1.488148 \cdot 10^{-10}$

Table 2: Numerical results for the approximate LCM of the set (75) with $\epsilon = 10^{-7}$ given by different LCM methods.

Example 5. The following five real polynomials with approximately equal root clusters are given in [6]:

$$\mathcal{P} = \left\{ \begin{array}{l} p_1(s) = (s-0.5)\,(s-0.502)\,(s+1)\,(s-2)\,(s-1.5) \\ p_2(s) = (s-0.501)\,(s-0.503)\,(s-1)\,(s+2)\,(s+1.5) \\ p_3(s) = (s-0.50066)\,(s-0.502393)\,(s+1.09553)\,(s-1.09568) \\ p_4(s) = (s-0.494572)\,(s-0.501611)\,(s-0.00833993) \\ p_5(s) = (s-0.499717)\,(s-0.50192) \end{array} \right\} \tag{76}$$

An approximate LCM of degree 14 was computed by both the SR-LCM and the H-LCM methods using standard double precision and different tolerances ε. The results are presented in Table 3. In quadruple precision both methods provided the trivial LCM of degree 19, but it is worth noting that the H-LCM algorthm was about 125 times faster than the SR-LCM algorithm. Generally, the H-LCM algorithm is significantly faster than the SR-LCM algorithm.

	ε	Degree	Rel. Error / Residual
SR-LCM	10^{-7}	14	$O\left(10^{-14}\right)$
H-LCM	10^{-15}	14	$O\left(10^{-17}\right)$

Table 3: Numerical results for the approximate LCM of the set (76) given by different LCM methods.

Example 6. The following two real polynomials are given [10]:

$$p_1(s) = (s + 1.5)\left(s^2 + 3.1\,s + 5.2\right) \text{ and } p_2(s) = (s + 1.5)(s + 3.7) \tag{77}$$

The exact LCM of the given polynomials is:

$$\text{lcm}\{p_1(s), p_2(s)\} = s^4 + 8.3\,s^3 + 26.87\,s^2 + 44.25\,s + 28.86$$

For $\varepsilon = 10^{-16}$ the H-LCM algorithm gives the following result:

$$l(s) = s^4 + 8.300000\,s^3 + 26.870000\,s^2 + 44.244999\,s + 28.860000$$

The relative error using the H-LCM method is $6.51 \cdot 10^{-16}$ which is better than the relative error $4.59 \cdot 10^{-14}$ produced by the LCM method developed in [10].

Example 7. The following polynomial set contains three real polynomials:

$$\mathcal{P} = \left\{ \begin{array}{l} p_1(s) = s^2 - 5\,s + 6 \\ p_2(s) = s^2 - (5 - \varepsilon_1)\,s + 6 \\ p_3(s) = s - (2 - \varepsilon_2) \end{array} \right\} \tag{78}$$

The coefficients of the polynomials of the above set $\mathcal{P}$ are perturbed by the parameters ε_1, ε_2, which are small positive numbers taking on values from 10^{-5} to 10^{-15}.

Considering the exact coefficients of the polynomials when $\varepsilon_1 = \varepsilon_2 = 0$, the LCM of the set is:

$$\text{lcm}\{\mathcal{P}\} = s^2 - 5\,s + 6$$

However, if the coefficients of the polynomials become inexact ($\varepsilon_1 \neq \varepsilon_2 > 0$), several ALCMs can be obtained with degrees varying from 2 to 4. Those ALCMs depend strongly on the selection of the numerical tolerances ε_1 and ε_2. The next test provides a picture of the sensitivity of the ALCM to small perturbations in the coefficients of the polynomials of the given set.

For each value of ε_1 and ε_2 from 10^{-5} to 10^{-15}, the H-LCM algorithm was applied to the given set (78). The numerical ε-rank for the determination of the degree of the ALCM is selected to be $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$. For all the pairs $(\varepsilon_1, \varepsilon_2)$ 121 ALCMs were obtained and their degrees are shown in Table 4. Every entry (i, j) in Table 4 represents the degree of the ALCM of the set $\mathcal{P}$ for $(\varepsilon_1, \varepsilon_2) = (10^i, 10^j)$ when $i, j = -5, -6, \ldots, -15$. For example, the degree of the ALCM for $(\varepsilon_1, \varepsilon_2) = (10^{-5}, 10^{-5})$ is 2. Additionally, Table 5 presents some selected results for the above ALCM computations.

7. Conclusions

In this paper the definition of the approximate LCM of several polynomials in $\mathbb{R}[s]$ is provided in matrix form and with the aid of the ERES division algorithm a new LCM computational method is derived, referred to as the Hybrid LCM algorithm (H-LCM). The developed method combines pure symbolic and numerical finite precision computations in order to form and solve a linear system which provides the degree and the coefficients of an approximate LCM of a given set of polynomials. The overall performance of the H-LCM algorithm shows that this method can be effectively used for sets of several polynomials with inexactly known coefficients. The implementation of the H-LCM algorithm in Maple is available upon request.

(i,j)	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15
-5	2	2	2	3	3	4	4	4	4	4	4
-6	2	2	2	2	3	3	4	4	4	4	4
-7	2	2	2	2	2	3	3	4	4	4	4
-8	2	2	2	2	2	2	3	3	4	4	4
-9	3	2	2	2	2	2	2	3	3	4	4
-10	3	3	2	2	2	2	2	2	3	3	4
-11	3	3	3	2	2	2	2	2	2	3	3
-12	3	3	3	3	2	2	2	2	2	2	3
-13	3	3	3	3	3	2	2	2	2	2	2
-14	3	3	3	3	3	3	2	2	2	2	2
-15	3	3	3	3	3	3	3	2	2	2	2

Table 4: LCM degrees of the set (78) for $(\varepsilon_1, \varepsilon_2) = (10^i, 10^j)$.

Degree	=	2, Tolerance $(\varepsilon_1, \varepsilon_2) = \left(10^{-15}, 10^{-15}\right)$
Residual	=	$1.100559 \cdot 10^{-17}$
ALCM	=	$s^2 - 5.0\,s + 6.0$
Degree	=	2, Tolerance $(\varepsilon_1, \varepsilon_2) = \left(10^{-14}, 10^{-11}\right)$
Residual	=	$9.880299 \cdot 10^{-15}$
ALCM	=	$s^2 - 5.00000000000286\,s + 5.99999999999857$
Degree	=	2, Tolerance $(\varepsilon_1, \varepsilon_2) = \left(10^{-5}, 10^{-5}\right)$
Residual	=	$4.249130 \cdot 10^{-18}$
ALCM	=	$s^2 - 5.0\,s + 6.0$
Degree	=	3, Tolerance $(\varepsilon_1, \varepsilon_2) = \left(10^{-15}, 10^{-11}\right)$
Residual	=	$4.336809 \cdot 10^{-17}$
ALCM	=	$s^3 - 6.99841030078348\,s^2 + 15.9920515039174\,s$ -11.9904618047009
Degree	=	4, Tolerance $(\varepsilon_1, \varepsilon_2) = \left(10^{-5}, 10^{-10}\right)$
Residual	=	$3.018419 \cdot 10^{-16}$
ALCM	=	$s^4 - 10.0000000007172\,s^3 + 37.0000000039680\,s^2$ $-60.0000000062129\,s + 36.0000000022916$

Table 5: Some ALCMs of the set (78).

Acknowledgements

The authors are thankful to the reviewers of this paper for their crucial comments which significantly improved this work.

References

[1] T. Kailath, Linear Systems, Prentice Hall, Inc., Englewood Cliffs, NJ, 1980.

[2] V. Kučera, Analysis and Design of Discrete Linear Control Systems, Prentice Hall, Inc., London, 1991.

[3] M. Vidyasagar, Nonlinear Systems Analysis, Prentice Hall, Inc., Englewood Cliffs, NJ, 2nd edn., 1993.

[4] N. Karcanias, C. Giannakopoulos, Necessary and sufficient conditions for zero assignment by constant squaring down, Linear Algebra and its Applications 122–124 (1989) 415–446.

[5] M. Hirsch, S. Smale, Differential Equations, Dynamical Systems and Linear Algebra, Academic Press, New York, 1974.

[6] N. Karcanias, M. Mitrouli, Numerical computation of the Least Common Multiple of a set of polynomials, Reliable computing 6 (4) (2000) 439–457.

[7] G. D. Forney, Minimal bases of rational vector spaces, with applications to multivariable linear systems, SIAM Journal on Control & Optimization 13 (3) (1975) 493–520.

[8] N. Karcanias, Geometric properties, invariants and the Toeplitz structure of minimal bases of rational vector spaces, IMA J. Control Theory Information 13 (1996) 1–11.

[9] N. Karcanias, M. Mitrouli, Minimal bases of matrix pencils and coprime matrix fraction descriptions, IMA J. Mathematical Control Theory and Information 19 (2002) 245–278.

[10] N. Karcanias, M. Mitrouli, System Theoretic Based Characterisation and Computation of LCM of a Set of Polynomials, Linear Algebra and its Applications 381 (2004) 1–23.

[11] K. Batselier, P. Dreesen, B. De Moor, A geometrical approach to finding multivariate approximate LCMs and GCDs, Linear Algebra and its Applications 438 (2013) 3618–3628.

[12] N. Karcanias, Invariance properties and characterisation of the greatest common divisor of a set of polynomials, Int. J. Control 46 (1987) 1751–1760.

[13] M. Mitrouli, N. Karcanias, Computation of the GCD of polynomials using Gaussian transformation and shifting, Int. J. Control 58 (1993) 211–228.

[14] D. Christou, N. Karcanias, M.Mitrouli, Matrix representation of the shifting operation and numerical properties of the ERES method for computing the greatest common divisor of sets of many polynomials, J. Comp. Appl. Math. 260 (2014) 54–67.

[15] B. N. Datta, Numerical Linear Algebra and Applications, SIAM, Philadelphia, USA, 2nd edn., 2010.

[16] D. Christou, N. Karcanias, M.Mitrouli, The ERES method for computing the approximate GCD of several polynomials, Applied Numerical Mathematics 60 (2010) 94–114.

[17] G. H. Golub, C. F. Van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, London, 2nd edn., 1989.

[18] B. Waldén, R. Karlson, J. G. Sun, Optimal backward perturbation bounds for the linear Least Squares problem, Numerical Linear Algebra with Applications 2 (3) (1995) 271–286.

[19] C. L. Lawson, R. J. Hanson, Solving Least-Squares Problems, Classics in Applied Mathematics, SIAM, 1995.

[20] P. Chin, R. M. Corless, G. F. Corliss, Optimization Strategies for the Approximate GCD Problem, in: Proc. ISSAC'98, Rostock, Germany, 228–235, 1998.

[21] D. A. Bini, P. Boito, Structured Matrix-Based Methods for Polynomial ϵ-gcd:Analysis and Comparison, in: Proc. ISSAC'07, Waterloo, Ontario, Canada, 9–16, 2007.

[22] J. R. Winkler, J. Allan, Structured total least norm and approximate GCDs of inexact polynomials, J. Comp. Appl. Math. 215 (2008) 1–13.

[23] I. Z. Emiris, A. Galligo, H. Lombardi, Certified approximate univariate GCDs, J. Pure and Applied Algebra 117 & 118 (1997) 229–251.

[24] E. Kaltofen, Z. Yang, L. Zhi, Structured Low Rank Approximation of a Sylvester Matrix, in: D. Wang, L. Zhi (Eds.), Symbolic-Numeric Computation, Birkhäuser, 69–83, 2007.

[25] M. Mitrouli, N. Karcanias, C. Koukouvinos, Numerical performance of the matrix pencil algorithm computing the greatest common divisor of polynomials and comparison with other matrix-based methodologies, J. Comp. Appl. Math. 76 (1996) 89–112.