The ERES Method for Computing the Approximate GCD of Several Polynomials [1]

D. Christou and N. Karcanias

Control Engineering Research Centre, School of Engineering and Mathematical Sciences, City University, Northampton Square, EC1V 0HB, London, U.K.

M. Mitrouli

Department of Mathematics, University of Athens, Panepistemiopolis 15784, Athens, Greece.

Abstract

The computation of the greatest common divisor (GCD) of a set of polynomials has interested the mathematicians for a long time and has attracted a lot of attention in recent years. A challenging problem that arises from several applications, such as control or image and signal processing, is to develop a numerical GCD method that inherently has the potential to work efficiently with sets of several polynomials with inexactly known coefficients. The presented work focuses on : (i) the use of the basic principles of the ERES methodology for calculating the GCD of a set of several polynomials and defining approximate solutions by developing the *hybrid* implementation of this methodology. (ii) the use of the developed framework for defining the *approximate notions* for the GCD as a distance problem in a projective space to develop an optimization algorithm for evaluating the strength of different ad-hoc approximations derived from different algorithms. The presented new implementation of ERES is based on the effective combination of symbolic-numeric arithmetic (hybrid arithmetic) and shows interesting computational properties for the approximate GCD problem. Additionally, an efficient implementation of the *strength* of an approximate GCD is given by exploiting some of the special aspects of the respective distance problem. Finally, the overall performance of the ERES algorithm for computing approximate solutions is discussed.

Key words: Approximate Greatest Common Divisor, Gaussian elimination, partial SVD, symbolic-numeric computations

Email addresses: `dchrist@math.uoa.gr` (D. Christou), `n.karcanias@city.ac.uk` (N. Karcanias), `mmitroul@math.uoa.gr` (M. Mitrouli).

[1] This research was financially supported by PENED 03ED 740 of the Greek Gen-

1 Introduction

The computation of the Greatest Common Divisor (GCD) of polynomials is a fundamental problem in many mathematical areas such as linear systems, control theory, network theory and communications. The GCD is central to the development of algebraic synthesis methods and its robust computation is a key issue. Finding the GCD of a set of m real univariate polynomials of maximal degree n, is a classical problem that has been considered before {[2,3,9,17,20,25,7,22,27,28] and reference therein} and can be seen in several engineering problems and applications such as image processing, signal processing, robotics and others. However, engineering models are not exact and they are always characterised by parameter uncertainty. This introduces some considerable problems with any framework based on exact symbolic tools, given that the underlined models are always characterised by parameter uncertainty. The central challenge is the transformation of algebraic notions to an appropriate analytic setup within which meaningful *approximate* solutions to exact algebraic problems may be sought. This motivates the need for transforming the algebraic problems into equivalent linear algebra problems and then develop approximate algebraic computations, which are appropriate for the case of computations on models characterised by parameter uncertainty.

A number of important invariants for linear systems rely on the notion of GCD of many polynomials and, in fact, the GCD is instrumental in defining system notions such as zeros, decoupling zeros, zeros at infinity, notions of minimality of system representations etc: On the other hand, systems and control methods provide concepts and tools, which enable the development of new computational procedures for the GCD. The computational methods for specifying the GCD or the *approximate* GCD of polynomials are separated in two main categories: The *Euclidean type methods* [4,5,26], which rely on pairwise computations and the *Matrix based methods*, which are based on processing a matrix formed directly from the coefficients of the given polynomials. According to the way that the matrix is processed, the matrix based methods are separated in those which *a)* form and work with a matrix that corresponds to the whole set of polynomials [2,3,9,20,25,28] and *b)* form a matrix for two polynomials and work on pairwise computations [10,22,26,33].

A major challenge for the control theoretic applications of the GCD is that frequently we have to deal with a very large number of polynomials. It is this requirement that makes the pairwise type approaches for GCD [4,5,22,26,33] not suitable for such applications. The GCD related work described in this paper goes back to the attempt to introduce the notion of almost zero of a set of polynomials [19] and study the properties of such zeros from the feedback

eral Secretariat for Research and Technology.

viewpoint. This work was subsequently developed to a methodology for computing the approximate GCD of polynomials using numerical linear algebra methods, such as the ERES [25] and matrix pencil methods [20]. The results in this area of computations are important in the development of meaningful solutions to algebraic system theory problems for models characterised by parameter uncertainty and they are linked to a large range of related problems such as :

(i) Almost non-coprimeness and solutions of polynomial Diofantine Equations.
(ii) Characterisation of Almost uncontrollability and Almost unobservability.
(iii) Approximate factorisation of rational transfer function models.

The existence of certain types and/or values of invariants and system properties may be classified as generic or nongeneric [15,23,32,16] on a family of linear models. Computing, or evaluating nongeneric types, or values of invariants and thus associated system properties on models with numerical inaccuracies is crucial for applications. For such cases, symbolic tools fail, since *almost always* lead to a generic solution, which does not represent the *approximate presence* of the property on the set of models under considerations. The formulation of a methodology for robust computation of nongeneric algebraic invariants, or nongeneric values of generic ones [21], has as prerequisites :

(a) The development of a numerical linear algebra characterisation of the invariants, which may allow the measurement of degree of presence of the property on every point of the parameter set.
(b) The development of special numerical tools, which avoid the introduction of additional errors.
(c) The formulation of appropriate criteria, which allow the termination of algorithms at certain steps and the definition of meaningful approximate solutions to the algebraic computation problem.

It is clear that the formulation of the algebraic problem as an equivalent numerical linear algebra problem, is essential in transforming concepts of algebraic nature to equivalent concepts of analytic character and thus setup the right framework for approximations.

This paper focuses on the development of a hybrid implementation of the method known as ERES (Extended Row Equivalence and Shifting operations) [25] for the computation of the GCD of many polynomials, which inherently has the potential to define approximate solutions to the GCD problem and the development of a numerical method for evaluating the strength of such approximations [18]. The basic principle of the methodology is that the GCD is a property of the row space of the basis matrix of the set of polynomials, and this property is also invariant under the symbolic operation of shifting. The

ERES method is thus a matrix based method based on a property of invariance of the GCD [17], that is row transformations and shifting operations on the basis matrix of the set of polynomials, which is formed from the coefficients of the polynomials of the set. The present method has the advantage that :

- It can handle many polynomials simultaneously, without resorting to the successive two at a time computations of the Euclidean or other pairwise based approaches [4,5,22,26,33].
- It invokes a numerical termination criterion that allows the derivation of *approximate* solutions to the GCD computation problem.
- It allows the combination of symbolic-numeric operations performed effectively in a mixture of numerical and symbolical steps.

In this paper the definition of the *approximate GCD* is considered as a distance problem in a projective space. The evaluation of the *strength* of approximation for approximate GCD computations has been an important drawback until recently for all matrix based methods dealing simultaneously with many polynomials. A rigorous definition of the approximate GCD has been given recently [18] that allows the computation of the strength of approximation and sets up a framework for computing the *optimal approximate GCD*. This approach is based on recent results [11] on the representation of the GCD of many polynomials in terms of the factorisation of the generalised resultant and a Toeplitz matrix representation of the GCD. These results allow the parameterisation of all perturbations, which are required to make a selected approximate GCD, an exact GCD of the perturbed set of polynomials. The evaluation of the strength of approximation is equivalent to an evaluation of a distance problem in a projective space and it is thus reduced to an optimization problem.

This paper aims:

- To use the basic principles of the ERES methodology [17,25] for defining approximate solutions to the GCD problem by developing the hybrid implementation of this methodology.
- To use the recently developed framework for defining the *approximate notions* for the GCD as a distance problem in a projective space [18] to develop an optimization algorithm for evaluating the strength of different ad-hoc approximations derived from different algorithms.

The new present implementation of ERES combines in an optimal setup the symbolical application of rows transformations and shifting, and the numerical computation of an appropriate termination criterion, which can provide the required approximate solutions. This combination highlights the hybridity of the ERES method. Generally, symbolic processing is used to improve on the conditioning of the input data, or to handle a numerically ill-conditioned subproblem, and numeric tools are used in accelerating certain parts of an algorithm, or in computing approximate outputs. The effective combination of symbolic and numerical operations depends on the nature of an algebraic method and the proper handling of the input data either as rational or floating-point numbers. Symbolic-numeric implementation is possible in software programming environments with symbolic-numeric arithmetic capabilities such as Maple, Mathematica, Matlab and others, which involve the efficient combination of exact (rational-symbolic) and numerical (floating-point) operations. This combination gives a different perspective in the way to implement an algorithm and uses the notion of *hybrid computations.*

In the following, $\mathbb{R}$ and $\mathbb{Z}$ denotes the real and integer numbers respectively and $\mathbb{R}[s]$ denotes the polynomial ring of univariate polynomials with coefficients from the reals. Capital letters denote matrices and small underlined letters denote vectors. If $A \in \mathbb{R}^{\mu \times \nu}$ is an $\mu \times \nu$ matrix with elements from $\mathbb{R}$, then its rank will be denoted by $\rho(A)$, the respective transposed matrix will be denoted by A^t and by $\|A\|_F$ or simply $\|A\|$ we denote the Frobenius matrix norm of A, [8,14].

1.1 Notation and preliminary definitions

Let us consider the set of univariate polynomials

$$\mathcal{P}_{h+1,n} = \Big\{ a(s), b_i(s) \in \mathbb{R}[s],\ i = 1,2,\ldots,h \text{ with} \\ n = \deg\{a(s)\},\ p = \max_{1 \le i \le h}\Big\{\deg\{b_i(s)\}\Big\} \le n \Big\} \tag{1}$$

We represent the polynomials $a(s), b_i(s)$ with respect to the highest degrees (n,p) as :

$$\begin{aligned} a(s) &= a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0\ ,\ a_n \neq 0 \\ b_i(s) &= b_{i,p}s^p + \ldots + b_{i,1}s + b_{i,0}\ ,\ i = 1,2,\ldots,h \end{aligned} \tag{2}$$

The set $\mathcal{P}_{h+1,n}$ will be called an (n,p)-ordered polynomial set and whenever we want to denote the number of elements and the maximal degree of a polynomial set we shall use this notation, otherwise the set of polynomials will be abbreviated as $\mathcal{P}$. Also, we shall denote by $\Pi(n,p;h+1)$ the family of polynomial sets $\mathcal{P}_{h+1,n}$ having $h+1$ elements and highest degrees (n,p), $n \ge p$; i.e. if the degrees of the polynomials in the set are denoted by $d_i,\ i = 0,\ldots,h$, then $d_0 \ge d_1 \ge d_2 \ge \ldots \ge d_h$ and $d_0 = n,\ d_1 = p$.

Definition 1 *For any $\mathcal{P}_{h+1,n}$ set, we define a vector representative (vr) $\underline{p}_{h+1}(s)$ and a basis matrix P_{h+1} represented as :*

$$\underline{p}_{h+1}(s) = [a(s), b_1(s), \ldots, b_h(s)]^t = [\underline{p}_0, \underline{p}_1, \ldots, \underline{p}_{n-1}, \underline{p}_n] \cdot \underline{e}_n(s) = P_{h+1}\, \underline{e}_n(s)$$

where $P_{h+1} \in \mathbb{R}^{(h+1)\times(n+1)}$, $\underline{e}_n(s) = [1, s, \ldots, s^{n-1}, s^n]^t$.

If c is the integer for which $\underline{p}_0 = \ldots = \underline{p}_{c-1} = \underline{0}$, $\underline{p}_c \neq 0$, then $c = w(\mathcal{P}_{h+1,n})$ is called the order of $\mathcal{P}_{h+1,n}$ and s^c is an elementary divisor of the GCD. The set $\mathcal{P}_{h+1,n}$ is considered to be a c-order set and will be called proper *if $c = 0$, and* non-proper *if $c \geq 1$. Clearly,*

$$\gcd(\mathcal{P}_{h+1,n}) = s^c \cdot \gcd(\mathcal{P}_{h+1,n-c})$$

In the following and without loss of generality, we assume that $\mathcal{P}_{h+1,n}$ is proper.

2 The ERES methodology

Given a set $\mathcal{P}_{h+1,n}$ of many polynomials with a basis matrix P_{h+1} the following operations are defined [17]:

(i) Elementary row operations with scalars from $\mathbb{R}$ on P_{h+1}.
(ii) Addition or elimination of zero rows on P_{h+1}.
(iii) If $\underline{a}^t = [0, \ldots, 0, a_l, \ldots, a_{n+1}] \in \mathbb{R}^{n+1}$, $a_l \neq 0$ is a row of P_{h+1} then we define as the *Shifting* operation

$$shf : \text{shf}(\underline{a}^t) = [a_l, \ldots, a_{n+1}, 0, \ldots, 0] \in \mathbb{R}^{n+1}$$

By $\text{shf}(\mathcal{P}_{h+1,n})$, we shall denote the set obtained from $\mathcal{P}_{h+1,n}$ by applying shifting on every row of P_{h+1}. Type (i), (ii) and (iii) operations are referred to as *Extended-Row-Equivalence and Shifting* (ERES) operations. The following theorem describes the properties characterizing the GCD of any given $\mathcal{P}_{h+1,n}$ [17].

Theorem 2 *For any set $\mathcal{P}_{h+1,n}$, with a basis matrix P_{h+1}, $\rho(P_{h+1}) = r$ and $\gcd\{\mathcal{P}_{h+1,n}\} = \phi(s)$ we have the following properties :*

(i) If $\mathcal{R}_P$ is the row space of P_{h+1}, then $\phi(s)$ is an invariant of $\mathcal{R}_P$ (e.g. $\phi(s)$ remains invariant after the execution of elementary row operations on P_{h+1}). Furthermore if $r = \dim(\mathcal{R}_P) = n + 1$, then $\phi(s) = 1$.
(ii) If $w(\mathcal{P}_{h+1,n}) = c \geq 1$, $\text{shf}(\mathcal{P}_{h+1,n})$, then

$$\phi(s) = \gcd\{\mathcal{P}_{h+1,n}\} = s^c \cdot \gcd\{\text{shf}(\mathcal{P}_{h+1,n})\}$$

(iii) If $\mathcal{P}_{h+1,n}$ is proper, then $\phi(s)$ is invariant under the combined ERES set of operations.

Remark 3 *The GCD of any set of polynomials is a property of the row space of the basis matrix of the set. Thus, the computation of the GCD requires selection of a basis that is best suited for such computations. The issue of selecting the best possible base from all those provided by the rows of the basis matrix without transforming the original data is critical for such computations and this problem is referred to as selection of the best* uncorrupted base *[25]. It is this property, which indicated that not all polynomials are required for the computation of the GCD.*

From Theorem (2) it is evident that ERES operations preserve the GCD of any $\mathcal{P}_{h+1,n}$ and thus can be easily applied in order to obtain a modified basis matrix with much simpler structure. The successive application of these operations on a basis matrix of a set of polynomials leads to the formulation of the ERES methodology for computing the GCD of a set of polynomials [25]. After successive applications of ERES operations on an initial basis matrix, the maximal degree of the resulting set of polynomials is reduced and after a finite number of steps the resulting basis matrix has rank one. In that stage, any row of the matrix specifies the coefficients of the required GCD of the set. Provided that operations are performed exactly, the computation of the GCD by using the ERES method is straightforward.

The development of the ERES algorithm requires the treatment of the following problems :

P1 Selection of the best uncorrupted base of the given set of polynomials.
P2 Application of the ERES operations.
P3 The development of a proper termination criterion.

The above requirements are actually the most essential parts of the ERES algorithm and their proper implementation determines the overall behavior of the algorithm. In the context of symbolic-numeric implementation, the problems P1-P3 can be handled as follows:

P1: A method for the selection of the best uncorrupted base [25] relies on the properties of the Gram matrix and uses tools from the theory of compound matrices [24]. However, this method seems to become inefficient due to its high complexity for sets of many polynomials. Alternatively, it is simpler to get a base of the original set of polynomials by applying symbolically (rational operations) a triangularisation method, such as Gaussian elimination, to the original basis matrix P_{h+1}. Although the original data are transformed, the introduction of round-off errors is avoided and the computation of the GCD remains unaffected.

P2: The most reliable tool for applying row operations is Gaussian Elimination with partial pivoting. Due to the iterative nature of the ERES method this process is preferable to be treated symbolically in order to avoid the propagation of errors during the iterations. Additionally, the first application of the Gaussian elimination to the basis matrix P_{h+1} can give us a base of the rowspace of P_{h+1}. The Shifting transformation is by definition a symbolic operation and together with the Gaussian Elimination with partial pivoting they represent the *main procedure* of the ERES algorithm.

P3: The algorithm's termination criterion relies on the proper detection of the final unity rank matrix and it can be based on the numerical computation of the singular values of a normalized matrix, obtained at the end of each iteration of the main procedure. We shall refer to it as the *rank-1 procedure.*

Therefore, the new approach for the implementation of the ERES algorithm involves the use of exact (rational) operations for its main procedure and numerical (floating-point) operations for the rank-1 procedure. The effects from this combination on the ERES algorithm will be discussed next and the algorithm in its new formulation is referred to as *Hybrid ERES algorithm.* The numerical implementation of the ERES algorithm in finite precision arithmetic has been analyzed in [25].

3 Implementation of the Hybrid ERES Algorithm

3.1 The hybrid structure of the algorithm

The construction of the algorithm of the ERES method is based on stable algebraic processes, which are applied iteratively on the initial basis matrix $P_{h+1} \in \mathbb{R}^{(h+1)\times(n+1)}$. The main target of the ERES method is to reduce the number of the rows of P_{h+1} and finally to end up to a unity rank matrix, which contains the coefficients of the GCD. An important key element in the implementation of the Hybrid ERES algorithm is the selection of the appropriate data structures to represent the input data. In a symbolic-numeric software programming environment the type of data structures suggests the type of arithmetic operations. Arithmetic operations with integers or fractions of integers (rational operations) are performed in infinite accuracy and this is an important feature to take into advantage.

Having a set $\mathcal{P}_{h+1,n}$ and its basis matrix P_{h+1}, a necessary preliminary step is to convert the given floating-point data to rational format (fractions of integers) and the next steps are implemented symbolically, using rational operations :

- Reorder the rows of P_{h+1} such that its first row corresponds to the polynomial of the lowest degree.
- Scale the first row of the P_{h+1} so as to have the maximum pivot.
- Apply Gaussian elimination with partial pivoting to P_{h+1}.
- Apply Shifting on every row of P_{h+1} .
- Delete the zero rows and columns and form a new basis matrix $P_{h+1}^{(\cdot)}$ with reduced dimensions.

These steps underlie the main procedure of the Hybrid ERES algorithm. We shall denote by $P_{h+1}^{(\kappa)}$ the matrix, which occurs after the κ^{th} iteration of the main procedure of the algorithm ($\kappa = 1, 2, \ldots$). If the produced matrix $P_{h+1}^{(\kappa)}$ has zero elements in its last column (i.e. the polynomials, which correspond to the rows of the matrix, have different degrees) the above steps go over $P_{h+1}^{(\kappa)}$.

If the matrix $P_{h+1}^{(\kappa)}$ in the κ^{th} iteration has no zero elements in its last column (i.e. the polynomials, which correspond to the rows of the matrix, have the same degree), then a numerical copy of it is made and the next steps are implemented numerically, using finite precision floating-point operations :

- Normalization of the rows of the matrix $P_{h+1}^{(\kappa)}$ using the Euclidean norm.
- Computation of the singular values of $P_{h+1}^{(\kappa)}$.

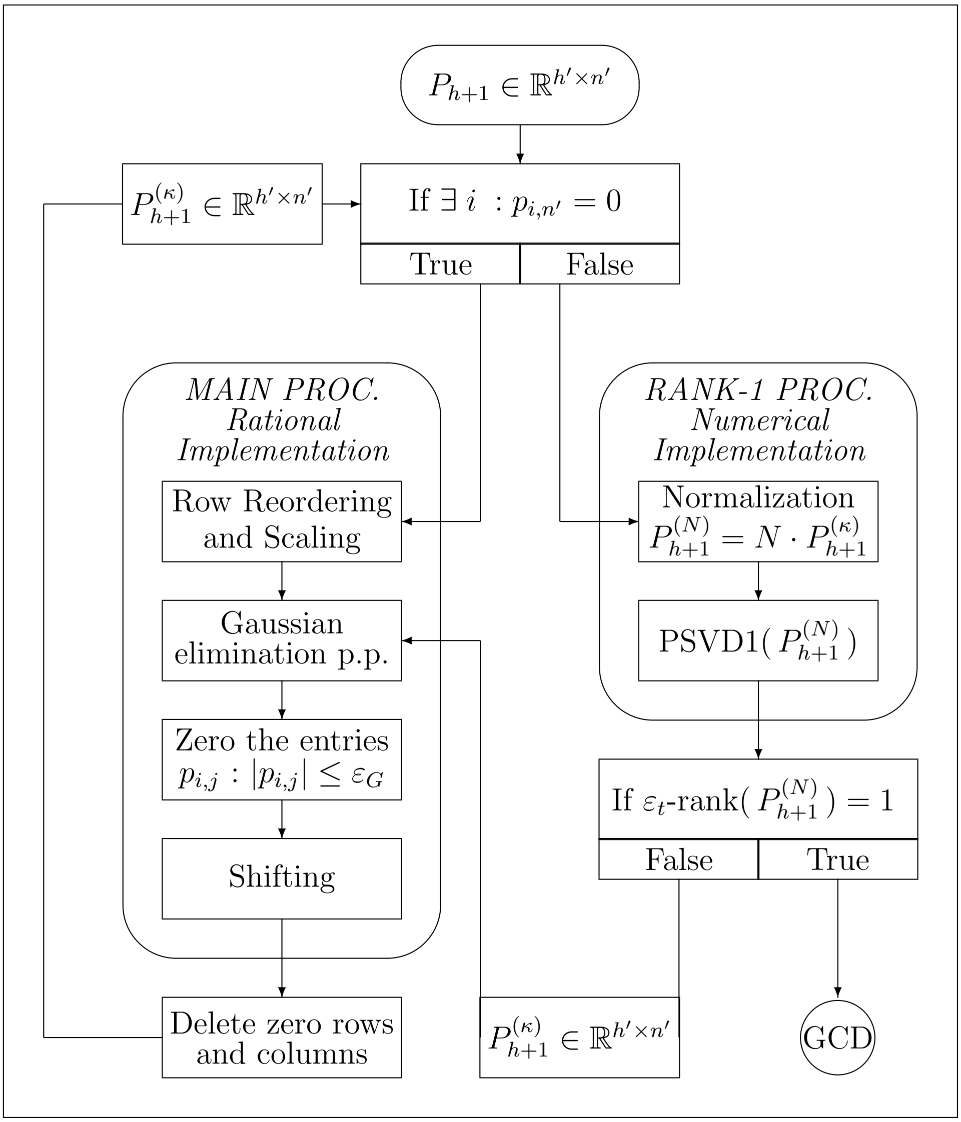
These steps underlie the rank-1 procedure of the Hybrid ERES algorithm. If the matrix $P_{h+1}^{(\kappa)}$ has numerical ε_t-rank equal to 1 according to a small specified accuracy $\varepsilon_t > 0$, the algorithm stops and gives an appropriate solution. Otherwise, the algorithm continues with the main procedure using the rational matrix $P_{h+1}^{(\kappa)}$. All the above steps are illustrated in Figure 1.

3.2 Computation of the GCD

The computation of the GCD with the Hybrid ERES algorithm depends on the accuracy of the input data and the performed operations. If the coefficients of the polynomials of the given set are exactly known and the GCD exists, then the successive symbolic application of the processes of the main procedure of the algorithm to the basis matrix of the set will lead to a final matrix with rank equal to one exactly. Thus, any row of this matrix give the coefficients of the GCD.

Otherwise, we must search for an approximate numerical solution, which is actually provided by the rank-1 procedure of the algorithm. The numerical computation of the singular values of $P_{h+1}^{(\kappa)}$ is a typical process to estimate the rank of a matrix and provides the ERES algorithm with a termination criterion. This criterion is applied when the polynomials, which correspond

Fig. 1. The Hybrid ERES Algorithm


$P_{h+1} \in \mathbb{R}^{h' \times n'}$
$P_{h+1}^{(\kappa)} \in \mathbb{R}^{h' \times n'}$
If $\exists\, i \ : p_{i,n'} = 0$
True
False
MAIN PROC.
Rational
Implementation
Row Reordering and Scaling
Gaussian elimination p.p.
Zero the entries $p_{i,j} : |p_{i,j}| \leq \varepsilon_G$
Shifting
RANK-1 PROC.
Numerical
Implementation
Normalization $P_{h+1}^{(N)} = N \cdot P_{h+1}^{(\kappa)}$
PSVD1($P_{h+1}^{(N)}$)
If ε_t-rank($P_{h+1}^{(N)}$) $= 1$
False
True
Delete zero rows and columns
$P_{h+1}^{(\kappa)} \in \mathbb{R}^{h' \times n'}$
GCD

(*) Initial values: $h' = h + 1, \ n' = n + 1$ (**) $p_{i,n'}$ denotes the elements of $P_{h+1}^{(\kappa)}$ in the last column.

to the rows of the matrix, have the same degree and it is described in the following theorem [25] :

Theorem 4 *Let $A = [\underline{a}_1, \ldots, \underline{a}_\mu]^t \in \mathbb{R}^{\mu \times \nu}$, $\mu \leq \nu$, $\underline{a}_1 \neq 0$, $i = 1, \ldots, \mu$. Then for an appropriate accuracy $\varepsilon_t > 0$ the numerical ε_t-rank of A equals to one ($\rho_{\varepsilon_t}(A) = 1$) if and only if the singular values $\sigma_\mu \leq \sigma_{\mu-1} \leq \cdots \leq \sigma_1$ of the normalization $A_N = [\underline{u}_1, \ldots, \underline{u}_\mu]^t \in \mathbb{R}^{\mu \times \nu}$, $\underline{u}_i = \underline{a}_i / \|\underline{a}_i\|_2$ of A satisfy the conditions*

$$|\sigma_1 - \sqrt{\mu}| \leq \varepsilon_t \text{ and } \sigma_i \leq \varepsilon_t, \ i = 2, 3, \ldots, \mu \ .$$

Every time when the algorithm reaches that stage, there is a potential ε_t-rank 1 matrix for a specific tolerance ε_t and if we accept values of $\varepsilon_t \leq 10^{-1}$, we can obtain a series of matrices, that yield an ε_t-GCD. The computation of the ε_t-GCD can be done according to the following proposition [25] :

Proposition 5 *Let $A = U \cdot \Sigma \cdot W^t$ be the singular value decomposition of a given matrix $A \in \mathbb{R}^{\mu \times \nu}$, $\rho(A) = 1$. Then a "best" rank one approximation to A in the Frobenius norm is given by $A_1 = \sigma_1 \cdot \underline{u} \cdot \underline{w}^t$, where σ_1 is the largest singular value of A and $\underline{u}$ and $\underline{w}$ are the first columns of the orthogonal matrices U and W of the singular value decomposition of A respectively. The vector $\underline{w}$ is then the "best" representative of the rows of matrix A in the sense of the rank one approximation.*

Therefore, the polynomial that comes from the first row of the right singular matrix of $P_{h+1}^{(N)}$, can be considered as the numerical output of the Hybrid ERES algorithm.

Of course, the singular value decomposition is undoubtedly a robust numerical procedure and, since we seek a unity rank matrix to stop the algorithm, the only essential information we need is concerned with the first two singular values of the matrix $P_{h+1}^{(N)}$. Thus, it is not necessary to perform the whole singular value decomposition. The development of a partial singular value decomposition algorithm is presented in [30,29]. The outline of a variation of the classical singular value decomposition method, especially developed for the efficient computation of the unique singular value and its right singular vector of an approximate ε_t-rank 1 matrix, is presented next and we shall refer to it as the *PSVD1 method.*

3.3 The partial SVD algorithm for approximate rank-1 matrices

Let us have a matrix A with dimensions $m \times n$ and let $\sigma_1 \leq \ldots \leq \sigma_{k-1} \leq \sigma_k$, $k = \min\{m, n\}$ be its singular values. The following algorithm is established :

Algorithm 1 ***The PSVD1 Algorithm***

```
INPUT:    – a matrix $A \in \mathbb{R}^{m \times n}$
          – a bound $\varepsilon_t > 0$ such that $\sigma_1 \leq \ldots \leq \sigma_{k-1} \leq \varepsilon_t \leq \sigma_k$,
            $k = \min\{m, n\}$.

STEP 1:   Bidiagonalization phase.

          If $m \geq n$ then
                transform $A$ into upper bidiagonal form $B_u$ by Householder
                transformations: $A = U_u \cdot B_u \cdot V_u^t$
          else
          transform $A$ into lower bidiagonal form $B_l$ by Householder
          transformations : $A = U_l \cdot B_l \cdot V_l^t$
     end if

STEP 2:   Rank 1 detection phase.
```

Having a bidiagonal matrix B_u (or B_l), we only need to partition the bidiagonal into unreduced subbidiagonals such that only one singular value is greater than ε_t. In order to detect such a property, we construct a $2n \times 2n$ symmetric tridiagonal matrix T with zero main diagonal from the elements of B_u (or B_l) and compute the Sturm Sequence for T and ε_t [13]. The positive symmetric eigenvalues of T are the singular values of B_u (or B_l) and the number of sign agreements in Sturm Sequence correspond to the number of singular values, which are greater or equal to ε_t, [8].

```
     – Let $a = [a_i],\ i = 1, \ldots, k$ be the elements of the main diagonal
       of $B_u$ (or $B_l$), and $b = [b_i],\ i = 1, \ldots, k-1$ be the elements of
       the superdiagonal of $B_u$ (or the subdiagonal of $B_l$).
     – Construct $\underline{c} = [a_1, b_1, a_2, b_2, \ldots, b_{k-1}, a_k] \in \mathbb{R}^{2k-1}$
     – Compute the Sturm Sequence for $\theta := \varepsilon_t$ :
          $p_0(\theta) = 1,\ p_1(\theta) = -\theta$
          For $i$ from $2$ to $2k$ do
               $p_i(\theta) = -\theta\, p_{i-1}(\theta) - c_{i-1}^2\, p_{i-2}(\theta)$
          end do
     – Let $N :=$ the number of sign agreements between the
       sequential terms $p_i(\theta)$ of the Sturm Sequence.
       Convention: If $p_i(\theta) = 0$ then $p_i(\theta)$ is in sign agreement
       with $p_{i-1}(\theta)$.

     If $N \geq 2$ then
          use a bisection method to compute a new bound $\varepsilon_t$.
          (We use the bisection method to find an estimation of the
          second larger singular value $\sigma_{k-1}$ [8].)
          Quit
     else

STEP 3:   Back transformation phase.

          – Find the largest (by absolute value) diagonal element $a_h$
            for $h = 1, \ldots, k$ of $B_u$ (or $B_l$).
          – Construct an appropriate Givens Rotation matrix $G$ for
            the $h^{th}$ row of $B_u$ (or the $h^{th}$ column of $B_l$).

          If $m < n$ then
               $\sigma_k :=$ the $h^{th}$ diagonal element of $S := G \cdot B_l$
```

```
                $w$ := the $h^{th}$ row of the matrix $V_l^t$
            else
                $\sigma_k$ := the $h^{th}$ diagonal element of $S' := B_u \cdot G^t$
                $w$ := the $h^{th}$ row of the matrix $W := G \cdot V_u^t$
            end if

        end if

OUTPUT: $\sigma_k$, $w$, $\varepsilon_t$
```

The PSVD1 algorithm can detect a unity rank matrix very efficiently. The process that dominates the algorithm is the bidiagonalization of the initial matrix using Householder transformations. It is a stable process and requires about $O(2mn^2 - \frac{2}{3}n^3)$ multiplications if $m < \frac{5}{3}n$ or $O(mn^2 + n^3)$ multiplications if $m \geq \frac{5}{3}n$ [30]. Also, when the initial matrix has not an ε_t-rank equal to 1, it is not necessary to compute all the sequential terms of the Sturm sequence because in this case, we only need a couple of sign agreements to conclude that we do not have a ε_t-rank 1 matrix. This simple test helps to save more computational time and if we scale the terms of the sequence properly, we can have a very quick and efficient way to detect an approximate ε_t-rank 1 matrix. In the case where we do have a unity rank matrix, the unique singular value and right singular vector can be computed explicitly without matrix products. Finally, the overall cost in computational operations is about the same size of that of the bidiagonalization method. The bidiagonal reduction may also be applied on an upper triangular matrix R, obtained from A by orthogonal transformations such that $A := Q\,R$. This step can improve the performance of the algorithm [6]. A method for a more accurate bidiagonal reduction, which combines Householder and Givens transformations, is presented in [1].

The PSVD1 algorithm is a quick and effective tool for the detection of an approximate unity rank matrix. It can increase the performance of other methods, such as the ERES method, and can be easily implemented in any software programming environment.

3.4 Selection of the proper numerical tolerance ε_t, ε_G.

The numerical accuracy ε_G has to do with the magnitude of the elements of the matrix that we obtain at the end of the main procedure of the ERES algorithm in each iteration. In most cases, ε_G can be set equal to $2^{-1}\varepsilon_m$, where ε_m is the machine's epsilon (hardware numerical accuracy).

On the other hand, ε_t is linked with the accuracy of the solution, which is obtained from the rank-1 procedure of the Hybrid ERES algorithm. An initial value of ε_t can be set by the user as an input (usually $\varepsilon_t = \varepsilon_G \approx \varepsilon_m$). However, as we described previously, every time when the algorithm reaches the rank-1 procedure, an ε_t-GCD can be obtained according to a new value of ε_t, which is actually determined from the PSVD1 procedure. Thus, at the end of the Hybrid ERES algorithm, we can have a series of ε_t-GCD's for all the different values of ε_t, computed by the algorithm itself. Unlike the previous version of the numerical ERES algorithm [25] where the choice of the ε_t was absolutely arbitrary, in the new presented hybrid algorithm of the ERES method, the numerical accuracy ε_t is proposed by the algorithm and this helps us to develop a better strategy for the best selection of the GCD.

3.5 Behavior of the ERES algorithm using hybrid computations

The combination of rational and numerical operations aims at the improvement of the stability of the ERES algorithm and the presence of *good* approximate solutions. The main iterative procedure of the algorithm and especially the process of Gaussian elimination, is entirely performed by using rational operations. With this technique any additional errors from the Gaussian elimination are avoided. The operations during the Gaussian elimination are always performed accurately and if the input data are exactly known and a GCD exists, the output of the algorithm is produced accurately from any row of the final unity rank matrix. Obviously, rational operations do not reveal the presence of approximate solutions. In cases of sets of polynomials with inexact coefficients, the presence of an approximate solution relies on the proper determination of a numerical ε_t-rank 1 matrix for a specific accuracy ε_t. Therefore, the singular value decomposition together with the normalization process of the matrix $P_{h+1}^{(\kappa)}$ are performed by using floating-point operations. Optionally, when we are interested in an approximate solution, we can also specify a numerical accuracy ε_G to control the magnitude of the elements of every matrix $P_{h+1}^{(\kappa)}$. The polynomial that comes from the proposition 5, can be considered as a GCD approximation and represents the numerical output of the ERES algorithm.

The normalization of the rows of any matrix $P_{h+1}^{(\kappa)}$ (by the Euclidean norm) does not introduce significant errors and in fact the following result can be proved [25]:

Proposition 6 *The normalization $P_{h+1}^{(N)}$ of a matrix $P_{h+1}^{(\kappa)} \in \mathbb{R}^{h' \times n'}$, computed by the method in the κ^{th} iteration, using floating-point arithmetic with unit round-off u, satisfies the properties*

$$P_{h+1}^{(N)} = N \cdot P_{h+1}^{(\kappa)} + E_N, \quad \|E_N\|_\infty \leq 3.003 \cdot n' \cdot u$$

where $N = diag(\nu_1, \nu_2, \ldots, \nu_{h'}) \in \mathbb{R}^{h' \times h'}$*,* $\nu_i = \left(\left\|P_{h+1}^{(\kappa)}[i, 1 \ldots n']\right\|_2\right)^{-1}$ *for* $i = 1, \ldots, h'$ *is the matrix accounting for the performed transformations and* $E_N \in \mathbb{R}^{h' \times n'}$ *the error matrix.*

The PSVD1 method is applied to the matrix $P_{h+1}^{(N)}$. The preliminary stage in this algorithm is the bidiagonal reduction of $P_{h+1}^{(N)}$ and in most bidiagonal reduction methods the error is expressed in the following form:

$$P_{h+1}^{(N)} + \delta P_{h+1}^{(N)} = U\, B\, V^t\ ,$$

$$\|\delta P_{h+1}^{(N)}\|_2 \leq u\, f(h', n')\, \|P_{h+1}^{(N)}\|_2$$

where B is bidiagonal, U and V are orthogonal, u is the machine precision and $f(h', n')$ is a modestly growing function of the dimensions of $P_{h+1}^{(N)}$, where $h' < h + 1$ and $n' < n + 1$.

It is important to notice that the rank-1 procedure is actually applied to a numerical copy of the matrix $P_{h+1}^{(\kappa)}$ and thus the performed transformations during the rank-1 procedure do not affect the matrix $P_{h+1}^{(\kappa)}$ when returning to the main procedure. For this reason, there is no accumulation of numerical errors. The only errors appearing are from the rank-1 procedure concerning the normalization and the partial singular value decomposition of the last matrix $P_{h+1}^{(\kappa)}$ and represent the total numerical error of the Hybrid ERES algorithm.

The combination of symbolic and floating-point operations ensures the stability of the algorithm and gives to the ERES the characteristics of a hybrid computational method.

3.6 Computational cost

For a set of polynomials the amount of multiplications or divisions performed in the κ^{th} iteration of the algorithm depends on the size of the matrix $P_{h+1}^{(\kappa)}$ and it is summarized in Table 1. The first iteration is the most computationally expensive iteration since the initial basis matrix is larger than any $P_{h+1}^{(\kappa)}$. Unless we know exactly the degree of the GCD of the set we cannot specify from the beginning the number of iterations required by the algorithm. Practically, the number of iterations is about $O(n)$. The computational cost of the PSVD1 method is dominated by the bidiagonalization of the input matrix.

The ERES method and its hybrid form presented here always produces estimates of the GCD as the result of the rank 1 approximation. Estimating how good such approximations are has been an open issue until recently [18], when

Table 1
Required operations for the matrix $P_{h+1}^{(\kappa)} \in \mathbb{R}^{h' \times n'}$

Gaussian elimination	Normalization	PSVD1
$O(\frac{z^3}{3}),\ z = \min\{h'-1, n'\}$	$O(2h'n')$	$O(2h'n'^2 - \frac{2}{3}n'^3)$

a proper framework for defining the notion of the approximate GCD and the evaluation of its strength has been defined. These issues are considered next.

4 The Notion of Approximate GCD and its Computation

The following results provide the fundamentals of a framework for the characterization of the almost GCD of a polynomial set and its strength. The notion of approximate GCD is defined as a distance problem and the quality of the approximate GCD is defined by the strength of the approximation between the given set and the given d-GCD variety [18]. Note that almost every polynomial may be considered as an approximate GCD as long as its degree is less or equal to the smallest degree in the set [18].

4.1 Representation of the GCD

The representation of the GCD relies on the square nonsingular Toeplitz matrices [11]. The following result provides a representation in matrix terms of the standard factorization of the GCD of a set of polynomials.

Definition 7 *Let*

$$v(s) = \lambda_r s^r + \cdots + \lambda_1 s + \lambda_0 \in \mathbb{R}[s] \text{ where } r \in \mathbb{Z}_+^*,\ \lambda_r, \lambda_0 \neq 0 \qquad (3)$$

be a polynomial. A special Toeplitz matrix representation $\hat{\Phi}_v \in \mathbb{R}^{(n+p)\times(n+p)}$ of $v(s)$ can be defined by

$$\hat{\Phi}_v = \begin{bmatrix} \lambda_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \lambda_1 & \lambda_0 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \lambda_r & \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \lambda_r & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \lambda_0 & 0 \\ 0 & \cdots & 0 & \lambda_r & \cdots & \lambda_1 & \lambda_0 \end{bmatrix} \qquad (4)$$

The matrix $\hat{\Phi}_v$ is lower triangular and unless its main diagonal is zero, it is always invertible. Its inverse $\hat{\Phi}_v^{-1} \in \mathbb{R}^{(n+p)\times(n+p)}$ is also lower triangular and can be found easily by computing its first column only.

Definition 8 *Consider the set $\mathcal{P} \equiv \mathcal{P}_{h+1,n}$, (1).*

(i) We can define a $p \times (n+p)$ matrix associated with $a(s)$:

$$S_0 = \begin{bmatrix} a_n & a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 & 0 & \cdots & 0 \\ 0 & a_n & a_{n-1} & \cdots & \cdots & a_1 & a_0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n & a_{n-1} & \cdots & \cdots & a_1 & a_0 \end{bmatrix}$$

and $n \times (n+p)$ matrices associated with each $b_i(s)$, $i = 1, 2, \dots, h$:

$$S_i = \begin{bmatrix} b_{i,p} & b_{i,p-1} & b_{i,p-2} & \cdots & b_{i,1} & b_{i,0} & 0 & \cdots & 0 \\ 0 & b_{i,p} & b_{i,p-1} & \cdots & \cdots & b_{i,1} & b_{i,0} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & b_{i,p} & b_{i,p-1} & \cdots & \cdots & b_{i,1} & b_{i,0} \end{bmatrix}$$

An extended Sylvester matrix or generalized resultant for the set $\mathcal{P}$ is defined by :

$$S_{\mathcal{P}} = \begin{bmatrix} S_0 \\ S_1 \\ \vdots \\ S_h \end{bmatrix} \in \mathbb{R}^{(p+hn)\times(n+p)} \tag{5}$$

(ii) The matrix $S_{\mathcal{P}}$ is the basis matrix of the set of polynomials :

$$S[\mathcal{P}] = \{a(s), sa(s), \dots, s^{p-1}a(s); b_1(s), \dots, b_h(s), sb_h(s), \dots, s^{n-1}b_h(s)\}$$

which is also referred to as the Sylvester Resultant *set of the given set $\mathcal{P}$ [12,32].*

We can relate an (n, p) extended Sylvester matrix to any polynomial set $\mathcal{P}_{h+1,n'}$ with two maximal degrees $n' = n - j$ and $p' = p - j$, $j > 0$ by assuming the first j coefficients of the polynomials of $\mathcal{P}_{h+1,n'}$ to be zero. The new matrix will be called (n, p)-*expanded generalized resultant* of the set $\mathcal{P}_{h+1,n'}$.

Remark 9 *The set of all generalized resultants corresponding to $h + 1$ polynomials with maximal nominal degrees (n, p) will be denoted by $\Psi(n, p; h + 1)$.*

Note 1 *In the following, we will denote by m the row dimension of the above extended Sylvester matrix $S_{\mathcal{P}}$, where $m = p + hn$.*

Toeplitz matrices and their properties are crucial elements in the representation of the GCD, which is defined by the following factorization of resultants result [11] :

Theorem 10 *Let $\mathcal{P} \in \Pi(n, p; h+1)$ be a proper polynomial set (1). Let $S_{\mathcal{P}}$ be the respective extended Sylvester matrix (5), and $\phi(s)$ be the GCD of the set with degree $0 < d \leq p$. Then, there exists a transformation matrix $\hat{\Phi}_\phi$ (4), such that*

$$S_{\mathcal{P}} = \left[\mathbb{O}_{m,d} | \tilde{S}^{(d)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_\phi \tag{6}$$

where $\mathbb{O}_{m,d}$ is the $m \times d$ zero matrix, $m = p+hn$, $\mathcal{P}^* \in \Pi(n-d, p-d; h+1)$ is the set of coprime polynomials obtained from the original set $\mathcal{P}$ after dividing its elements by the GCD, $\phi(s)$, and $\tilde{S}^{(d)}_{\mathcal{P}^*}$ is the respective $(m, n+p-d)$ extended Sylvester matrix of $\mathcal{P}^*$.

We will denote by $S^{(d)}_{\mathcal{P}^*} = [\mathbb{O}_{m,d} | \tilde{S}^{(d)}_{\mathcal{P}^*}]$ the corresponding (n, p)-*expanded generalized resultant* of the reduced co prime set $\mathcal{P}^* \equiv \mathcal{P}^*_{h+1,n-d}$. The following results show an important property of generalized resultants [11,31].

Theorem 11 *Let $\mathcal{P} \in \Pi(n, p; h+1)$ be a polynomial set (1) and $S_{\mathcal{P}}$ the respective generalized resultant (5). Then*

$$\rho(S_{\mathcal{P}}) = n + p - d \;\Leftrightarrow\; \deg\{\gcd(\mathcal{P})\} = d$$

Proposition 12 *The GCD of $\mathcal{P}$ is the same as the GCD of $S[\mathcal{P}]$, that is :*

$$\gcd(\mathcal{P}) = \gcd(S[\mathcal{P}])$$

4.2 The Notion of the Approximate GCD

We will now consider the notion of the *approximate GCD* and the development of a computational procedure that allows the evaluation of how good is the given *approximate GCD*. Defining approximate notions of GCD using the Euclidean approach for two polynomials has been an issue that has attracted a lot of attention recently [10,26,28]. Our approach [18] is based on the resultant properties of the GCD and applies to any number of polynomials without resorting to the features of a particular algorithm.

The essence of current methods for introduction of *approximate GCD* is the relaxation of conditions characterizing the exact notion. We will define the strength or quality of a given *approximate GCD* by the size of the minimal perturbation required to make a chosen *approximate GCD* an exact GCD of a perturbed set of polynomials.

Let us have a set $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$ as defined in (1) and (2) and $a(s) = \underline{a}^t\, \underline{e}_n(s)$, $b_i(s) = \underline{b}_i^t\, \underline{e}_p(s)$, $i = 1,\ldots h$ with $\underline{e}_j(s) = [s^j, s^{j-1}, \ldots, s, 1]^t$ for $j = n$ or p respectively. We may correspond to the set $\mathcal{P}_{h+1,n}$ the vector

$$\underline{p}_{h+1,n} = \left[\underline{a}^t,\, \underline{b}_1^t,\, \ldots,\, \underline{b}_h^t\right]^t \in \mathbb{R}^N \tag{7}$$

where $N = (n+1) + h\,(p+1)$, or alternatively a point $P_{h+1,n}$ in the projective space P^{N-1}. The set $\Pi(n,p;h+1)$ is clearly isomorphic with $\mathbb{R}^N$, or P^{N-1}. An important question relates to the characterization of all points of P^{N-1}, which correspond to sets of polynomials with a given degree GCD. Such sets of polynomials correspond to certain varieties of P^{N-1}, which are defined below. We first note that an alternative representation of $\mathcal{P}_{h+1,n}$ is provided by the generalised Sylvester resultant $S_{\mathcal{P}} \in \mathbb{R}^{(p+hn)\times(n+p)}$ which is a matrix defined by the vector of coefficients $\underline{p}_{h+1,n}$. If we denote by $C_k(\cdot)$ the k^{th} compound of $S_{\mathcal{P}}$ [24], then the varieties characterising the sets having, a given degree d, GCD are defined below [18]:

Proposition 13 *Let $\Pi(n,p;h+1)$ be the set of all polynomial sets $\mathcal{P}_{h+1,n}$ with $h+1$ elements and with the two higher degrees (n,p), $n \geq p$ and let $S_{\mathcal{P}}$ be the Sylvester resultant of the general set $\mathcal{P}_{h+1,n}$. The variety of P^{N-1}, which characterise all sets $\mathcal{P}_{h+1,n}$ having a GCD with degree d, $0 < d \leq p$ is defined by the set of equations*

$$C_{n+p-d+1}\left(S_{\mathcal{P}}\right) = 0 \tag{8}$$

Conditions (8) define polynomial equations in the parameters of the vector $\underline{p}_{h+1,n}$, or the point $\mathcal{P}_{h+1,n}$ of P^{N-1}. The set of equations in (8) define a variety of P^{N-1}, which will be denoted by $\Delta_d(n,p;h+1)$ and will be referred to as the *d-GCD variety* of P^{N-1}. $\Delta_d(n,p;h+1)$ characterises all sets in $\Pi(n,p;h+1)$, which have a GCD with degree d.

Remark 14 *The sets $\Delta_d(n,p;h+1)$ have measure zero [7] and thus the existence of a nontrivial GCD of degree $d > 0$ is a nongeneric property.*

The important question that is posed, is how close the given set $\mathcal{P}_{h+1,n}$ is to given variety $\Delta_d(n,p;h+1)$. Defining the notion of the *approximate GCD* is linked to introducing an appropriate distance of $\mathcal{P}_{h+1,n}$ from $\Delta_d(n,p;h+1)$. In fact, if $\mathcal{Q}^i_{h+1,n}$ is some perturbation set (to be properly defined) and assuming that $\mathcal{P}'^{\,i}_{h+1,n} = \mathcal{P}_{h+1,n} + \mathcal{Q}^i_{h+1,n}$ such that $\mathcal{P}'^{\,i}_{h+1,n} \in \Delta_d(n,p;h+1)$, then the GCD of $\mathcal{P}'^{\,i}_{h+1,n}$, $\phi(s)$, with degree d defines the notion of the approximate

GCD and its strength is defined by the *size* of the perturbation $\mathcal{Q}^i_{h+1,n}$. Numerical procedures such as ERES, produce estimates of an approximate GCD. Estimating the size of the corresponding perturbations provides the means to evaluate how good such approximations are. By letting the parameters of the GCD free and searching for the minimal size of the corresponding perturbations a distance problem is formulated that is linked to the definition of the *optimal approximate GCD.* The key questions which have to be considered for such studies are:

(i) Existence of perturbations of $\mathcal{P}_{h+1,n}$ yielding

$$\mathcal{P}^{'}_{h+1,n} = \mathcal{P}_{h+1,n} + \mathcal{Q}_{h+1,n} \quad \in \Delta_d(n,p;h+1)$$

(ii) Parameterisations of all such perturbations.

(iii) Determine the minimal distance of $\mathcal{P}_{h+1,n}$ from an element of $\Delta_d(n,p;h+1)$ with a given GCD $u(s)$, and thus evaluation of strength of $u(s)$.

(iv) Determine the minimal distance of $\mathcal{P}_{h+1,n}$ from $\Delta_d(n,p;h+1)$ and thus compute the *optimal approximate GCD.*

Here we are concerned with the issues (i)-(iii), which relate to the evaluation of the strength of a given approximation, which is not necessarily optimal.

4.3 Parametrisation of GCD varieties and the Computation of Strength of the Approximate GCD

The characterisation of the $\Delta_d(n,p;h+1)$ variety in a parametric form, as well as subvarieties of it, is a crucial issue for the further development of the topic. The subset of $\Delta_d(n,p;h+1)$, characterised by the property that all $\mathcal{P}_{h+1,n}$ in it have a given GCD $u(s) \in \mathbb{R}[s]$, $\deg\{u(s)\} = d$, can be shown to be a subvariety of $\Delta_d(n,p;h+1)$ and shall be denoted by $\Delta^u_d(n,p;h+1)$. In fact $\Delta^u_d(n,p;h+1)$ is characterised by the equations of $\Delta_d(n,p;h+1)$ and a set of additional linear relations amongst the parameters of the vector $\underline{p}_{h+1,n}$.

Proposition 15 *Consider the set* $\Pi(n,p;h+1)$*,* P^{N-1} *be the associated projective space,* $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$ *and let* $S_{\mathcal{P}}$ *be the associated resultant. Then,*

(i) The variety $\Delta_d(n,p;h+1)$ *of* P^{N-1} *is expressed parametrically by the resultant*

$$S_{\mathcal{P}} = \left[\mathbb{O}_{m,d} | \tilde{S}^{(d)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_u \tag{9}$$

where $\mathbb{O}_{m,d}$ *is the* $m \times d$ *zero matrix,* $m = p+hn$*,* $\hat{\Phi}_u$ *is the* $(n+p)\times(n+p)$ *Toeplitz representation of an arbitrary* $u(s) \in \mathbb{R}[s]$ *with* $\deg\{u(s)\} = d$ *and* $\tilde{S}^{(d)}_{\mathcal{P}^*} \in \mathbb{R}^{m\times(n+p-d)}$ *is the* (n,d)*-expanded generalized resultant of an arbitrary set of polynomials* $\mathcal{P}^* \in \Pi(n-d,p-d;h+1)$*.*

(ii) The variety $\Delta_d^u(n,p;h+1)$ of P^{N-1} is defined by (9) with the additional constraint that $u(s) \in \mathbb{R}[s]$ is given.

Clearly, the free parameters in $\Delta_d(n,p;h+1)$ are the coefficients of the polynomials of $\Pi(n-d,p-d;h+1)$. Having defined the description of these varieties we consider next the perturbations that transfer a general set $\mathcal{P}_{h+1,n}$ on a set $\mathcal{P}'_{h+1,n}$ on them. If $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$ we can define an (n,p)-ordered perturbed set by:

$$\begin{aligned}\mathcal{P}'_{h+1,n} &\triangleq \mathcal{P}_{h+1,n} - \mathcal{Q}_{h+1,n} \in \Pi(n,p;h+1) \qquad (10)\\ &= \Big\{ p'_i(s) = p_i(s) - q_i(s) : \deg\{q_i(s)\} \leq \deg\{p_i(s)\}, i = 0,\ldots,h \Big\}\end{aligned}$$

Using the set of perturbations defined above we may now show that any polynomial from a certain class may become an exact GCD of a perturbed set under a family of perturbations.

Proposition 16 *Given a set $\mathcal{P}_{h+1,n}$ with maximal degrees (n,p), $n \geq p$ and a polynomial $v(s) \in \mathbb{R}[s]$ with $\deg\{v(s)\} \leq p$. There always exists a family of (n,p)-ordered perturbations $\mathcal{Q}_{h+1,n}$ such that for every element of this family $\mathcal{P}'_{h+1,n} = \mathcal{P}_{h+1,n} - \mathcal{Q}_{h+1,n}$ has a GCD which is divisible by $v(s)$.*

The above result establishes the existence of perturbations making $v(s)$ an exact GCD of the perturbed set and motivates the following definition, which defines $v(s)$ as an approximate GCD in an optimal sense.

Definition 17 *Let $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$ and $v(s) \in \mathbb{R}[s]$ be a given polynomial with $\deg\{v(s)\} = r \leq p$. Furthermore, let $\Sigma_v = \{\mathcal{Q}_{h+1,n}\}$ be the set of all (n,p)-order perturbations such that*

$$\mathcal{P}'_{h+1,n} = \mathcal{P}_{h+1,n} - \mathcal{Q}_{h+1,n} \in \Pi(n,p;h+1) \qquad (11)$$

with the property that $v(s)$ is a common factor of the elements of $\mathcal{P}'_{h+1,n}$. If $\mathcal{Q}^\circ_{h+1,n}$ is the minimal norm element of the set Σ_v, then $v(s)$ is referred as an r-order almost common factor of $\mathcal{P}_{h+1,n}$, and the norm of $\mathcal{Q}^\circ_{h+1,n}$, denoted by $\|\mathcal{Q}^\circ\|$ is defined as the strength of $v(s)$. If $v(s)$ is the GCD of

$$\mathcal{P}^\circ_{h+1,n} = \mathcal{P}_{h+1,n} - \mathcal{Q}^\circ_{h+1,n} \qquad (12)$$

then $v(s)$ will be called an r-order almost GCD of $\mathcal{P}_{h+1,n}$ with strength $\|\mathcal{Q}^\circ\|$.

Thus, any polynomial *v(s)* may be considered as an *approximate GCD*, as long as $\deg\{v(s)\} \leq p$.

Theorem 18 *[18] For $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$, let $S_{\mathcal{P}} \in \Psi(n,p;h+1)$ be the corresponding generalized resultant and let $v(s) \in \mathbb{R}[s]$, $\deg\{v(s)\} = r \leq p$. Then:*

(i) Any perturbation set $\mathcal{Q}_{h+1,n} \in \Pi(n,p;h+1)$ that leads to $\mathcal{P}'_{h+1,n} = \mathcal{P}_{h+1,n} - \mathcal{Q}_{h+1,n}$, which has $v(s)$ as common divisor, has a generalized resultant $S_\mathcal{Q} \in \Psi(n,p;h+1)$ that is expressed as shown below:

(a) If $v(0) \neq 0$ then

$$S_\mathcal{Q} = S_\mathcal{P} - S^{(r)}_{\mathcal{P}^*} \cdot \hat{\Phi}_v = S_\mathcal{P} - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_v \tag{13}$$

where $\mathbb{O}_{m,r}$ is the $m \times r$ zero matrix, $\hat{\Phi}_v$ is the $(n+p) \times (n+p)$ Toeplitz representation of $v(s)$ as defined in (4) and $S^{(r)}_{\mathcal{P}^*} \in \mathbb{R}^{m \times (n+p)}$ is the (n,p)-expanded generalized resultant of an arbitrary set of polynomials $\mathcal{P}^* \in \Pi(n-r,p-r;h+1)$.

(b) If $v(s)$ has k zeros at $s=0$, then

$$S_\mathcal{Q} = S_\mathcal{P} - \tilde{S}^{(r)}_{\mathcal{P}^*} \cdot \Theta_v \tag{14}$$

where $\tilde{S}^{(r)}_{\mathcal{P}^*}$ is again the (n,p)-expanded resultant of an arbitrary set of polynomials $\mathcal{P}^* \in \Pi(n-r,p-r;h+1)$ and Θ_v is the $(n+p-k) \times (n+p)$ representation of $v(s)$ defined by:

$$\Theta_v = \begin{bmatrix} \lambda_k & \lambda_{k-1} & \lambda_{k-2} & \cdots & \cdots & \lambda_0 & 0 & \cdots & \cdots & 0 \\ 0 & \lambda_k & \lambda_{k-1} & \lambda_{k-2} & \cdots & \cdots & \lambda_0 & 0 & & 0 \\ \vdots & \ddots & \ddots & & & & & \ddots & & \\ \vdots & & & \ddots & & & & & \ddots & \\ 0 & \cdots & \cdots & 0 & \lambda_k & \lambda_{k-1} & \lambda_{k-2} & \cdots & \cdots & \lambda_0 \end{bmatrix} \tag{15}$$

(ii) If the parameters of $\mathcal{P}^*$ are constrained such that $\tilde{S}^{(r)}_{\mathcal{P}^*}$ has full rank, then $v(s)$ is a GCD of the perturbed set $\mathcal{P}'_{h+1,n}$.

Corollary 19 *Let $\mathcal{P}_{h+1,n} \in \Pi(n,p;h+1)$ and $v(s) \in \mathbb{R}[s]$, $\deg\{v(s)\} = r \leq p$. The polynomial $v(s)$ is an r-order almost common divisor of $\mathcal{P}_{h+1,n}$ and its strength is defined as a solution of the following minimization problems:*

(a) If $v(0) = 0$, then its strength is defined by the global minimum of

$$f(\mathcal{P},\mathcal{P}^*) = \min_{\forall \mathcal{P}^*} \left\| S_\mathcal{P} - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_v \right\|_F \tag{16}$$

(b) If $v(s)$ has k zeros at $s=0$, then its strength is defined by the global minimum of

$$f(P,\mathcal{P}^*) = \min_{\forall \mathcal{P}^*} \| S_\mathcal{P} - \tilde{S}^{(r)}_{\mathcal{P}^*} \cdot \Theta_v \|_F \tag{17}$$

where $\mathcal{P}^*$ takes values from the set $\Pi(n,p;h+1)$.

Furthermore $v(s)$ is an r-order almost GCD of $\mathcal{P}_{h+1,n}$, if the minimal corresponds to a coprime set $\mathcal{P}^*$ or to full rank $S_{\mathcal{P}^*}$.

For the computation of the minimization problems in (16) or (17) we need an appropriate numerical procedure. However, the successful computation of such a global minimum is not always guaranteed. The minimization problem in (16) or (17) is actually non-convex and in cases of sets of many polynomials, where the number of arbitrary parameters is usually large, it is very likely to lead to unsatisfactory results. On the other hand, it is easier to find some bounds for the main function in (16) which is:

$$\|S_{\mathcal{Q}}\| = \left\| S_{\mathcal{P}} - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_v \right\|$$

We shall analyze how the norm $\|S_{\mathcal{Q}}\|$ is bounded and what information we can get from these bounds. Without loss of generality, we assume a given polynomial $v(s)$ with no zero roots. If we combine the relations (4) and (13), we shall have the following equation :

$$S_{\mathcal{Q}} \cdot \hat{\Phi}_v^{-1} = S_{\mathcal{P}} \cdot \hat{\Phi}_v^{-1} - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \tag{18}$$

Let $\widehat{S}_{\mathcal{P}} = S_{\mathcal{P}} \cdot \hat{\Phi}_v^{-1}$ and split $\widehat{S}_{\mathcal{P}}$ such that

$$\widehat{S}_{\mathcal{P}} = \hat{S}'_{\mathcal{P}} + \hat{S}''_{\mathcal{P}} \tag{19}$$

where $\hat{S}''_{\mathcal{P}}$ has the same structure as $S^{(r)}_{\mathcal{P}^*} = \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right]$. Specifically, if we denote by $A[i,j]$ the (i,j) element of a matrix A, the partition of $\widehat{S}_{\mathcal{P}}$ is based on the next rule :

$$\hat{S}'_{\mathcal{P}}[i,j] = \begin{cases} \widehat{S}_{\mathcal{P}}[i,j], & \text{if} \quad S^{(r)}_{\mathcal{P}^*}[i,j] = 0 \\ 0, & \text{if} \quad S^{(r)}_{\mathcal{P}^*}[i,j] \neq 0 \end{cases} \quad \forall\, i,j \tag{20}$$

Therefore, $\hat{S}''_{\mathcal{P}}$ can be presented as $\hat{S}''_{\mathcal{P}} = \left[\mathbb{O}_{m,r} | \bar{S}\right]$, where $\bar{S}$ is a $m \times (n+p-r)$ matrix. From (18) and (19) we get the following relation:

$$\begin{aligned} S_{\mathcal{Q}} \cdot \hat{\Phi}_v^{-1} &= \hat{S}'_{\mathcal{P}} + \left[\mathbb{O}_{m,r} | \bar{S}\right] - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] = \\ &= \hat{S}'_{\mathcal{P}} + \left[\mathbb{O}_{m,r} | \bar{S} - \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \end{aligned} \tag{21}$$

It readily follows that

$$S_{\mathcal{Q}} = \hat{S}'_{\mathcal{P}} \cdot \hat{\Phi}_v + \left[\mathbb{O}_{m,r} | \bar{S} - \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \cdot \hat{\Phi}_v \tag{22}$$

and if we use the Frobenius norm, which relates in a direct way to the set of polynomials, we get :

$$\|S_{\mathcal{Q}}\| \leq \|\hat{S}'_{\mathcal{P}}\| \, \|\hat{\Phi}_v\| + \| \left[\mathbb{O}_{m,r} | \bar{S} - \tilde{S}^{(r)}_{\mathcal{P}^*}\right] \| \, \|\hat{\Phi}_v\| \tag{23}$$

It is clear that any exact common factor of the polynomials of the set is expecting to give $\|S_{\mathcal{Q}}\| = 0$. Therefore, we may consider a polynomial as a good approximation of an exact common divisor or the exact GCD of a given set, if $\|S_{\mathcal{Q}}\|$ is close enough to zero.

The structure of the matrices here allows us to select the arbitrary parameters of the set $\mathcal{P}^*$ such that

$$\bar{S} = \tilde{S}^{(r)}_{\mathcal{P}^*} \tag{24}$$

Then, if we apply the Frobenius norm and the result (24) to the equations (21) and (22) and since the condition number of $\hat{\Phi}_v$ according to Frobenius norm is [8] :

$$Cond(\hat{\Phi}_v) = \|\hat{\Phi}_v\|\,\|\hat{\Phi}_v^{-1}\| \geq n + p > 1 \tag{25}$$

the following important inequality will be obtained :

$$\frac{\|\hat{S}'_{\mathcal{P}}\|}{\|\hat{\Phi}_v^{-1}\|} \leq \|S_{\mathcal{Q}}\| \leq \|\hat{S}'_{\mathcal{P}}\|\,\|\hat{\Phi}_v\| \tag{26}$$

Apparently, if $\|\hat{S}'_{\mathcal{P}}\| = 0$, then $\|S_{\mathcal{Q}}\| = 0$ and therefore the given polynomial $v(s)$ can be considered as an exact common divisor of degree r of the original set. Otherwise, the inequality (26) gives a lower bound of $\|S_{\mathcal{Q}}\|$, which indicates the minimum distance towards $\|S_{\mathcal{Q}}\| = 0$.

Remark 20 *Given a polynomial $v(s)$ with no zero roots, we shall denote by:*

(i) $\mathcal{S}(v)$ the strength of $v(s)$ given by the minimization problem (16).

(ii) $\underline{\mathcal{S}}(v) = \|\hat{S}'_{\mathcal{P}}\| \left(\|\hat{\Phi}_v^{-1}\|\right)^{-1}$ the lower strength bound of $v(s)$.

(iii) $\overline{\mathcal{S}}(v) = \|\hat{S}'_{\mathcal{P}}\|\,\|\hat{\Phi}_v\|$ the upper strength bound of $v(s)$.

The computation of the strength bounds $\underline{\mathcal{S}}(v)$ and $\overline{\mathcal{S}}(v)$ is straightforward and the results can be used as an indicator of the strength of the given approximation. For example, if $\underline{\mathcal{S}}(v) >> 1$, then the given approximation has very poor quality and the opposite holds if $\overline{\mathcal{S}}(v) << 1$. The strength bounds are very reliable indicators of the strength of a given GCD approximation $v(s)$ provided that the respective matrix $\hat{\Phi}_v$ is well conditioned ($Cond(\hat{\Phi}_v) \approx O(n+p)$).

The following algorithm establishes a methodology for the evaluation of the strength bounds of a given approximate GCD.

Algorithm 2 ***The algorithm of Strength Bounds.***

INPUT : *Give a set $\mathcal{P} \in \Pi(n, p; h+1)$ of univariate polynomials.*
Give a univariate polynomial $v(s)$ of degree $r \leq p$ with
no zero roots.

```
STEP 1 :  Construct the $(n,p)$-expanded Sylvester matrix $S_{\mathcal{P}}$ of $\mathcal{P}$.
          Construct the special Toeplitz representation $\hat{\Phi}_v$ of $v(s)$.
          Compute the first column of the inverse of $\hat{\Phi}_v$ and
          construct the matrix $\hat{\Phi}_v^{-1}$.
STEP 2 :  Compute the matrix $\widehat{S}_{\mathcal{P}}$ by solving the linear system :
          $\hat{\Phi}_v^{\,t} \cdot \widehat{S}_{\mathcal{P}}^{\,t} = S_{\mathcal{P}}^{\,t}$
STEP 3 :  Split appropriately $\widehat{S}_{\mathcal{P}}$ such that $\widehat{S}_{\mathcal{P}} = \hat{S}'_{\mathcal{P}} + \hat{S}''_{\mathcal{P}}$ , (20) .
          Compute the Frobenius norms $\|\hat{S}'_{\mathcal{P}}\|$, $\|\hat{\Phi}_v^{-1}\|$ and $\|\hat{\Phi}_v\|$.
OUTPUT :  $\underline{\mathcal{S}}(v) = \|\hat{S}'_{\mathcal{P}}\| \left(\|\hat{\Phi}_v^{-1}\|\right)^{-1}$ , $\overline{\mathcal{S}}(v) = \|\hat{S}'_{\mathcal{P}}\| \, \|\hat{\Phi}_v\|$
```

Due to the special structure of the matrices $S_{\mathcal{P}}$ and $\hat{\Phi}_v^{-1}$, it is possible to avoid the matrix operations and compute the norms explicitly and more efficiently. The inverse of the lower triangular matrix $\hat{\Phi}_v$ is computed by solving a simple linear system of the form $\hat{\Phi}_v \; \underline{x} = \underline{e}_{n+p}^{1}$, where $\underline{x}$ represents the first column of the matrix $\hat{\Phi}_v^{-1}$ and $\underline{e}_{n+p}^{1} = [1, 0, \ldots, 0]^t \in \mathbb{R}^{n+p}$. The number of operations required by the above algorithm is presented in Table 2. The total amount of operations is about $O\left(\frac{3n^2h+10n^2+2nr-r^2}{2}\right)$ for $n = p$. For an effective computation of the GCD, the respective matrix $\hat{S}'_{\mathcal{P}}$ is quite sparse and the required operations are less than $O(2hn^2)$.

Table 2
Required operations for the computation of the Strength Bounds $\underline{\mathcal{S}}(v)$ and $\overline{\mathcal{S}}(v)$.

$\hat{\Phi}_v^{-1}$	$\widehat{S}_{\mathcal{P}}$	$\|\hat{S}'_{\mathcal{P}}\|$	$\|\hat{\Phi}_v^{-1}\|$	$\|\hat{\Phi}_v\|$
$O\left(\frac{(n+p)^2}{2}\right)$	$O\left(\frac{n^2+p^2h}{2}\right)$	$O\left(np(h+1)\right)$	$O\left(\frac{(n+p)^2}{2}\right)$	$O\left(\frac{(n+p)r-r^2}{2}\right)$

5 Computational results

In this section we shall present a number of results given by the previous algorithms for different sets of several polynomials. More specifically, we use the ERES algorithm to find the ε_t-GCD of a set of polynomials $\mathcal{P}_{h+1,n}$ for ε_t accuracy and we evaluate the given GCD approximation v by computing the strength bounds $\underline{\mathcal{S}}(v)$ and $\overline{\mathcal{S}}(v)$ and, of course, its strength $\mathcal{S}(v)$. Our intention is to show the advantages of the hybrid implementation of the ERES algorithm in contrast with its fully numerical implementation. Thus, we shall denote by *H-ERES* the Hybrid ERES algorithm as described in the previous section 3 and presented in figure 1, and *N-ERES* the fully numerical ERES algorithm as presented in [25]. The algorithms were implemented in the software programming environment of Maple using both rational and variable floating-point arithmetic[2] . For the computation of the GCD in the examples 21 and

[2] The program codes of the ERES and the other presented algorithms are available from the authors.

22, the software numerical accuracy of Maple was set to 16 digits in order to simulate the common hardware accuracy. On the other hand, for the computation of the strength bounds and the strength of the given approximation, the software numerical accuracy was set to 64 digits for more reliable results. The strength of the given approximation was computed in Maple by using a built-in numerical minimization routine. An extended example is presented in the appendix.

Example 21 *We consider the set $\mathcal{P}_{3,11} \in \Pi(11, 11; 3)$, $h = 2$, $n = p = 11$ with polynomials :*

$$\begin{aligned}
a(s) = & -16.316\, s^{11} + 182.73\, s^{10} - 185.83\, s^9 + 106.68\, s^8 \\
& -266.22\, s^7 + 125.80\, s^6 - 195.53\, s^5 + 243.81\, s^4 \\
& +23.013\, s^3 + 64.186\, s^2 - 24.300\, s - 43.810 \\
b_1(s) = & \, 4.6618\, s^{11} - 52.209\, s^{10} + 53.094\, s^9 - 30.481\, s^8 \\
& +76.064\, s^7 - 35.944\, s^6 + 55.866\, s^5 - 69.659\, s^4 \\
& -6.5751\, s^3 - 18.339\, s^2 + 6.9428\, s + 12.517 \\
b_2(s) = & -4.1155\, s^{11} + 47.507\, s^{10} - 59.034\, s^9 + 2.2157\, s^8 \\
& -45.276\, s^7 + 83.932\, s^6 - 34.013\, s^5 + 15.007\, s^4 \\
& +4.3083\, s^3 - 9.0031\, s^2 + 14.297\, s - 14.783
\end{aligned}$$

The exact GCD of the set $\mathcal{P}_{3,11}$ is $g(s) = 1$ and the tolerance ε_G is set equal to the machine's epsilon in 16-digits accuracy $\varepsilon_m \approx 2.2204 \cdot 10^{-16}$. The H-ERES algorithm gave us five possible approximate solutions for different values of the tolerance ε_t. In Table 3 we denote these solutions by $v_i(s)$, $i = 1, 2, \ldots, 5$.

Table 3
Results for the ε_t-GCD of the set $\mathcal{P}_{3,11}$.

H-ERES	$v_1(s)$	$v_2(s)$	$v_3(s)$	$v_4(s)$	$v_5(s)$
Degree	1	8	4	5	2
ε_t	$9.5 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$3 \cdot 10^{-3}$	$6 \cdot 10^{-4}$	$5 \cdot 10^{-5}$
$\mathcal{S}(v)$	14.1684562	144.132727	0.72610271	4.00725481	0.02571431
$\underline{\mathcal{S}}(v)$	3.07315597	0.53653745	0.22191334	0.06286887	0.00319468
$\overline{\mathcal{S}}(v)$	69.9370280	3563.87058	930.143927	322.951121	0.19880416
$Cond(\hat{\Phi}_v)$	22.7573962	6642.35195	4191.47371	5136.90032	62.2296806
Main Iter.	20	6	14	12	18
Rank-1 Iter.	11	4	8	7	10

From the presented results in Table 3 we can conclude that:

(i) The "best" approximate solution is the polynomial

$$v_5(s) = s^2 - 11.28371806974011\, s + 11.64469379842480$$

which has the lowest strength $\mathcal{S}(v) = 0.02571431$ and it is given for $\varepsilon_t = 5 \cdot 10^{-5}$.

(ii) The strength $\mathcal{S}(v)$ is well bounded when $Cond(\hat{\Phi}_v) < (h+1)(n+p) = 66$.

(iii) The number of iterations of the main procedure is greater than the respective number of iterations of the rank-1 procedure.

The N-ERES algorithm gave approximately the same results with a restriction to 8 digits of accuracy and $\varepsilon_G > 10^{-6}$. □

Example 22 *[9,28] We consider the set $\mathcal{P}_{7,5} \in \Pi(5,4;7)$, with polynomials :*

$$\begin{aligned}
a(s) &= s^5 - 4.000001\, s^3 + 0.99999999\, s^2 - 4.0000009\\
b_1(s) &= 1.000001\, s^4 - 4.9999999\, s^2 + 3.999996\\
b_2(s) &= 2.000001\, s^3 + 1.000002\, s^2 - 8.000008\, s - 4.00001\\
b_3(s) &= s^3 + 1.000002\, s^2 - 4.000001\, s - 4.000009\\
b_4(s) &= 2.000003\, s^5 + 1.000003\, s^2 - 8.000008\, s^3 - 4.00001\\
b_5(s) &= 3.0000003\, s^4 - 14.0000009\, s^2 + 7.9999998\\
b_6(s) &= 1.000001\, s^4 - 7.0000032\, s^2 + 12.000001
\end{aligned}$$

The exact GCD of the set is $g(s) = 1$.

Table 4
Results for the ε-GCD $v(s)$ of the set $\mathcal{P}_{5,7}$.

Alg.	Deg	ε_t, ε_G	$\mathcal{S}(v)$	$\underline{\mathcal{S}}(v)$	$\overline{\mathcal{S}}(v)$
H-ERES	2	10^{-4}	$3.8340580 \cdot 10^{-7}$	$1.2513283 \cdot 10^{-7}$	$1.3154232 \cdot 10^{-6}$
Alg [9]	2	10^{-4}	$4.5660000 \cdot 10^{-7}$	$1.4922755 \cdot 10^{-7}$	$1.5687127 \cdot 10^{-6}$
Alg1 [28]	2	10^{-4}	$5.1120196 \cdot 10^{-7}$	$1.6715856 \cdot 10^{-7}$	$1.7572068 \cdot 10^{-6}$
Alg2 [28]	2	10^{-4}	$2.2630548 \cdot 10^{-6}$	$7.4117014 \cdot 10^{-7}$	$7.7913400 \cdot 10^{-6}$

In Table 4, we denote by Alg [#] the algorithm presented in the respective citation. In [28] there are two variations of the presented algorithm, denoted here by Alg1 and Alg2. In this example, there was no significant difference in the performance of the H-ERES and N-ERES algorithms. In fact, we obtained the same GCD from both of them but it is worthwhile to compare its strength with the solutions proposed in [9,28]. Actually, the GCD provided by the H-ERES algorithm is $v(s) = s^2 - 4$ and it has the lowest strength amongst them. We can also notice that the lower strength bound $\underline{\mathcal{S}}(v)$ and the upper strength bound $\overline{\mathcal{S}}(v)$ give us a good estimation of the quality of our approximation before we solve the minimization problem and evaluate the actual strength $\mathcal{S}(v)$. The condition number of the matrix $\hat{\Phi}_v$ is $Cond(\hat{\Phi}_v) = 10.51221518$. □

Example 23 *We tested and compared the algorithms H-ERES and N-ERES for various random sets of polynomials and some representative results are pre-*

sented in Table 5. More speciffically, we examined random sets of polynomials with integer coefficients between -10^7 and 10^7, which have an exact GCD with rational coefficients. The N-ERES algorithm uses the data as floating-point numbers and in every polynomial set $\mathcal{P}_{h+1,n}$ in Table 5 the selected software floating-point accuracy of the system, denoted by Dig, is selected as the minimum accuracy that N-ERES requires to produce a polynomial, which has at least the same degree as the respective exact GCD. The H-ERES algorithm uses the data as symbolic numbers and produces the GCD of the set accurately. However, we can also have a numerical solution given by the rank-1 procedure (PSVD1 algorithm) according to the proposition 5. For the numerical part of the H-ERES algorithm, the software floating-point accuracy remained the same with the selected accuracy for the N-ERES algorithm in order to compare the results. The internal accuracies of the algorithm are $\varepsilon_t = \varepsilon_G \approx 10^{-Digits}$.

We compared the two algorithms in respect with the numerical relative error between the exact GCD and the given solution, the number of main iterations and the required time of execution. The results in Table 5 show that the H-ERES algorithm produces numerical solutions of better quality than the N-ERES algorithm.

The following notation is used in the next Tables 5 and 6 :

Set	*Set $\mathcal{P}_{h+1,n}$ of random polynomials.*
Main	*Number of main iterations.*
h, n, p	*Parameters of the set as defined in (1).*
SVD	*Number of SVD or PSVD1 calls.*
d	*Degree of the exact GCD of the set $\mathcal{P}_{h+1,n}$.*
Time	*Time of execution in seconds.*
Dig.	*Number of digits of software accuracy.*
Rel. Err.	*Relative error.*
Alg.	*Type of algorithm.*

The relative error is given by $Rel = \frac{\|\underline{v}-\underline{g}\|_2}{\|\underline{g}\|_2}$, where $\underline{v}$, $\underline{g}$ are the coefficient vectors of the provided solution $v(s)$ and the exact GCD $g(s)$ respectively. $\|\cdot\|_2$ denotes the Euclidean norm.

In the case of large sets of polynomials, the N-ERES fails to produce accurate results in the standard floating-point precision of 16-digits of accuracy. On the contrary, the H-ERES algorithm works with this accuracy and gives very good results, which are presented in Table 6 [3] *.*

□

[3] The sets of polynomials in Tables 5 and 6 are the same.

Table 5
Results from H-ERES and N-ERES algorithms for random sets of polynomials.

Set	h, n, p, d	Dig.	Alg.	Main	SVD	Time	Rel. Err.
$\mathcal{P}_{11,10}$	$10, 10, 10, 1$	16	N-ERES	5	2	0.203	$5.28309 \cdot 10^{-16}$
			H-ERES	3	3	0.266	$6.44538 \cdot 10^{-16}$
$\mathcal{P}_{21,20}$	$20, 20, 20, 2$	55	N-ERES	5	2	0.688	$1.02570 \cdot 10^{-19}$
			H-ERES	3	3	0.797	$1.00764 \cdot 10^{-53}$
$\mathcal{P}_{31,30}$	$30, 30, 30, 3$	34	N-ERES	6	2	1.749	$3.38425 \cdot 10^{-20}$
			H-ERES	2	2	2.156	$2.53046 \cdot 10^{-33}$
$\mathcal{P}_{31,40}$	$30, 40, 40, 4$	45	N-ERES	10	2	3.375	$3.45159 \cdot 10^{-21}$
			H-ERES	4	3	14.250	$1.14197 \cdot 10^{-44}$
$\mathcal{P}_{51,30}$	$50, 30, 30, 5$	58	N-ERES	2	2	3.812	$1.27734 \cdot 10^{-19}$
			H-ERES	3	3	3.703	$4.14280 \cdot 10^{-56}$

Table 6
Results from H-ERES algorithm in 16 digits of accuracy.

Set	h, n, p, d	Dig.	Main	PSVD1	Time	Rel. Err.
$\mathcal{P}_{11,10}$	$10, 10, 10, 1$	16	3	3	0.266	$6.44538 \cdot 10^{-16}$
$\mathcal{P}_{21,20}$	$20, 20, 20, 2$	16	3	3	0.704	$2.65026 \cdot 10^{-16}$
$\mathcal{P}_{31,30}$	$30, 30, 30, 3$	16	3	3	2.171	$4.78899 \cdot 10^{-16}$
$\mathcal{P}_{31,40}$	$30, 40, 40, 4$	16	5	4	14.156	$1.65847 \cdot 10^{-16}$
$\mathcal{P}_{51,30}$	$50, 30, 30, 5$	16	3	3	3.094	$5.44165 \cdot 10^{-16}$

6 Conclusions

The computation of an approximate GCD of set of many polynomials is a challenging problem. There are several algorithms, which have employed the process of singular value decomposition in their structures in order to estimate the degree of a GCD for a specific tolerance ε_t [7,9,10]. The ERES method is a simple method, which use the SVD method as a numerical tool for the computation of an approximate GCD. A new kind of implementation of the ERES algorithm is presented, more reliable and efficient in computing approximate solutions. This new implementation takes advantage of the hybrid nature of the algorithm and its ability to yield exact or approximate solutions by manipulating the input data.

The ERES method is quite effective, when properly implemented in a programming environment of hybrid arithmetic. We can have large sets of real polynomials without restrictions to the type of data. Actually the method is proved to be faster when the polynomials of a given set $\mathcal{P}_{h+1,n}$ are linearly depended. The main advantage of the ERES algorithm is that it succeeds in reducing the size of the original matrix very quickly during the iterations and this leads to fast data processing and low memory consumption. What is more important with ERES is that for a fixed tolerance ε_G and without any further cost, the method can detect various degrees of polynomials for different tolerances ε_t, which can be considered as degrees of approximate GCDs. This important result derives from theorem 4 and this theorem guarantees the existence of an ε_t-GCD. The performance of the algorithm is better, when using hybrid computations. This type of computations provides us with more accurate results within acceptable time limits and without using high software floating-point accuracy. Furthermore, we can make the algorithm faster by using the PSVD1 algorithm.

The investigation of the approximate GCD for many polynomials has been introduced and the overall approach has been based on its characterization as a distance problem. This has been achieved by a combination of previous results related to the representation theory [11], the definition of the strength of the approximation [18] and the study of the optimisation properties of the defined problem. The theoretic characterisation of the *best* approximate GCD requires further investigation of the optimisation results of [18].

A An extended example

In this appendix we shall demonstrate the steps of the Hybrid ERES algorithm computing the GCD of a set of polynomials.

Let us have the following set $\mathcal{P}_{3,2} \in \Pi(2,2;3)$, $h = n = p = 2$ with polynomials:

$$\begin{aligned}
a(s) &= 2.0\,s^2 + 2.380952380952381\,s - 0.3809523809523810\\
b_1(s) &= s^2 - 3.642857142857143\,s + 0.5\\
b_2(s) &= 1.5\,s^2 - 7.214285714285714\,s + 1.0
\end{aligned}$$

The coefficients of the set $\mathcal{P}_{3,2}$ are given in 16-digits floating-point format. The basis matrix of the set has the form :

$$P_3 = \begin{bmatrix} -0.3809523809523810 & 2.380952380952381 & 2.0 \\ 0.5 & -3.642857142857143 & 1.0 \\ 1.0 & -7.214285714285714 & 1.5 \end{bmatrix}$$

The data of P_3 are converted to an approximate[4] rational format and the

[4] The conversion is done according to Maple's arithmetic by using the directive `convert`.

matrix obtains the following form:

$$P'_3 = \begin{bmatrix} -\frac{8}{21} & \frac{50}{21} & 2 \\ \frac{1}{2} & -\frac{51}{14} & 1 \\ 1 & -\frac{101}{14} & \frac{3}{2} \end{bmatrix}$$

The error from this conversion is

$$\|P'_3 - P_3\|_\infty = 2.8571428571428571 \cdot 10^{-16}$$

This is our initial error in 16-digits floating-point arithmetic, where the respective machine's epsilon is $\varepsilon_m = 2.22044604925031308 \cdot 10^{-16}$. This error does not grow during the iterations of the algorithm. $\| \cdot \|_\infty$ denotes the infinity matrix norm, [8,14].

We are starting now the process of the H-ERES algorithm :

- *Main procedure. Iteration 1*

Initial matrix $P_3^{(0)}$:

$$\begin{bmatrix} -\frac{8}{21} & \frac{50}{21} & 2 \\ \frac{1}{2} & -\frac{51}{14} & 1 \\ 1 & -\frac{101}{14} & \frac{3}{2} \end{bmatrix} \tag{A1}$$

Row reordering and Scaling :

$$\begin{bmatrix} -\frac{80}{21} & \frac{500}{21} & 20 \\ \frac{1}{20} & -\frac{51}{140} & \frac{1}{10} \\ \frac{1}{10} & -\frac{101}{140} & \frac{3}{20} \end{bmatrix} \tag{A2}$$

Gaussian elimination with partial pivoting :

$$\begin{bmatrix} -\frac{80}{21} & \frac{500}{21} & 20 \\ 0 & -\frac{27}{280} & \frac{27}{40} \\ 0 & 0 & 0 \end{bmatrix} \tag{A3}$$

Shifting transformation :

$$\begin{bmatrix} -\frac{80}{21} & \frac{500}{21} & 20 \\ -\frac{27}{280} & \frac{27}{40} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A4}$$

- *Main procedure. Iteration 2*

Initial matrix $P_3^{(1)}$:

$$\begin{bmatrix} -\frac{80}{21} & \frac{500}{21} & 20 \\ -\frac{27}{280} & \frac{27}{40} & 0 \end{bmatrix} \tag{A5}$$

Row reordering and Scaling :

$$\begin{bmatrix} -\frac{27}{280} & \frac{27}{40} & 0 \\ -\frac{4}{105} & \frac{5}{21} & \frac{1}{5} \end{bmatrix} \tag{A6}$$

Gaussian elimination with partial pivoting :

$$\begin{bmatrix} -\frac{27}{280} & \frac{27}{40} & 0 \\ 0 & -\frac{1}{35} & \frac{1}{5} \end{bmatrix} \tag{A7}$$

Shifting transformation :

$$\begin{bmatrix} -\frac{27}{280} & \frac{27}{40} & 0 \\ -\frac{1}{35} & \frac{1}{5} & 0 \end{bmatrix} \tag{A8}$$

Deletion of zero columns :

$$\begin{bmatrix} -\frac{27}{280} & \frac{27}{40} \\ -\frac{1}{35} & \frac{1}{5} \end{bmatrix} \tag{A9}$$

Passing to rank-1 procedure.

- *Rank-1 procedure* Normalization :

$$P_3^{(N)} = \begin{bmatrix} -0.09642857142857143 & 0.6750000000000000 \\ -0.02857142857142857 & 0.2000000000000000 \end{bmatrix} \tag{A10}$$

PSVD1 :

The final matrix $P_3^{(N)}$ has rank 1 for selected tolerance $\varepsilon_t = 10^{-15}$ and the respective right singular vector is

$$\underline{w}^t = [-0.141421356237309, 0.9899494936611661]^t$$

- *GCD computation :*

  The solution is given either from the rows of the final matrix of the main iterative procedure :

  $$v(s) = s - \frac{1}{7}$$

  or from the vector $\underline{w}$:

  $$v'(s) = s - 0.1428571428571427$$

  The polynomial $v'(s)$ is an ε_t-GCD and the distance between these two solutions is

  $$\|\underline{v} - \underline{v}'\|_\infty \approx 1.57 \cdot 10^{-16}$$

Now, we shall evaluate the strength bounds of the polynomial $v'(s)$ according to the *Algorithm 2*. We follow the next steps :

- Construction of the required matrices :

$$S_{\mathcal{P}} = \begin{bmatrix} 2.0 & 2.380952380952381 & -0.3809523809523810 & 0 \\ 0 & 2.0 & 2.380952380952381 & -0.3809523809523810 \\ 1.0 & -3.642857142857143 & 0.5 & 0 \\ 0 & 1.0 & -3.642857142857143 & 0.5 \\ 1.5 & -7.214285714285714 & 1.0 & 0 \\ 0 & 1.5 & -7.214285714285714 & 1.0 \end{bmatrix} \tag{A11}$$

$$\hat{\Phi}_{v'} = \begin{bmatrix} -0.1428571428571427 & 0 & 0 & 0 \\ 1.0 & -0.1428571428571427 & 0 & 0 \\ 0 & 1.0 & -0.1428571428571427 & 0 \\ 0 & 0 & 1.0 & -0.1428571428571427 \end{bmatrix} \tag{A12}$$

$$\hat{\Phi}_{v'}^{-1} = \begin{bmatrix} -7.000000000000008 & 0.0 & 0.0 & 0.0 \\ -49.00000000000011 & -7.000000000000008 & 0.0 & 0.0 \\ -343.0000000000011 & -49.00000000000011 & -7.000000000000008 & 0.0 \\ -2401.000000000010 & -343.0000000000011 & -49.00000000000011 & -7.000000000000008 \end{bmatrix} \tag{A13}$$

$$\widehat{S}_{\mathcal{P}} =$$

$$\begin{bmatrix} 5.425000000000012 \cdot 10^{-14} & -0.2142857142857063 & 1.0 & 0.0 \\ 3.797500000000013 \cdot 10^{-13} & 5.425000000000012 \cdot 10^{-14} & -0.2142857142857063 & 1.0 \\ 2.485000000000006 \cdot 10^{-14} & -0.1428571428571391 & 0.5000000000000000 & 0.0 \\ 1.739500000000006 \cdot 10^{-13} & 2.485000000000006 \cdot 10^{-14} & -0.1428571428571391 & 0.5000000000000000 \\ -2.473333333333339 \cdot 10^{-14} & -0.2857142857142889 & -0.3809523809523810 & 0.0 \\ -1.731333333333339 \cdot 10^{-13} & -2.473333333333339 \cdot 10^{-14} & -0.2857142857142889 & -0.3809523809523810 \end{bmatrix} \tag{A14}$$

$$\hat{S}'_{\mathcal{P}} = \begin{bmatrix} 5.425000000000012 \cdot 10^{-14} & 0.0 & 0.0 & 0.0 \\ 3.797500000000013 \cdot 10^{-13} & 5.425000000000012 \cdot 10^{-14} & 0.0 & 0.0 \\ 2.485000000000006 \cdot 10^{-14} & 0.0 & 0.0 & 0.0 \\ 1.739500000000006 \cdot 10^{-13} & 2.485000000000006 \cdot 10^{-14} & 0.0 & 0.0 \\ -2.473333333333339 \cdot 10^{-14} & 0.0 & 0.0 & 0.0 \\ -1.731333333333339 \cdot 10^{-13} & -2.473333333333339 \cdot 10^{-14} & 0.0 & 0.0 \end{bmatrix} \tag{A15}$$

$$\bar{S} = \tilde{S}^{(r)}_{\mathcal{P}^*} = \begin{bmatrix} -0.2142857142857063 & 1.0 & 0.0 \\ 0.0 & -0.2142857142857063 & 1.0 \\ -0.1428571428571391 & 0.5000000000000000 & 0.0 \\ 0.0 & -0.1428571428571391 & 0.5000000000000000 \\ -0.2857142857142889 & -0.3809523809523810 & 0.0 \\ 0.0 & -0.2857142857142889 & -0.3809523809523810 \end{bmatrix} \tag{A16}$$

- Computation of the Frobenius norms $\|\hat{S}'_{\mathcal{P}}\|$, $\|\hat{\Phi}_v^{-1}\|$ and $\|\hat{\Phi}_v\|$:

$$\begin{aligned} \|\hat{S}'_{\mathcal{P}}\| &= 4.6128999736247051 \cdot 10^{-13} \\ \|\hat{\Phi}_{v'}\| &= 12.288205727444521 \\ \|\hat{\Phi}_{v'}^{-1}\| &= 350.14568396597551 \\ Cond(\hat{\Phi}_{v'}) &= 4302.6621991506793 \end{aligned}$$

- Computation of the strength bounds :

$$\begin{aligned} \underline{\mathcal{S}}(v') = \|\hat{S}'_{\mathcal{P}}\| / \|\hat{\Phi}_{v'}^{-1}\| &= 1.317423056990459 \cdot 10^{-15} \\ \overline{\mathcal{S}}(v') = \|\hat{S}'_{\mathcal{P}}\| \cdot \|\hat{\Phi}_{v'}\| &= 5.668426387602378 \cdot 10^{-12} \end{aligned}$$

The strength of the polynomial $v'(s) = s - 0.1428571428571424$ is evaluated by the minimization problem (16) :

$$f(\mathcal{P}, \mathcal{P}^*) = \min_{\forall \mathcal{P}^*} \left\| S_{\mathcal{P}} - \left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*} \right] \cdot \hat{\Phi}_{v'} \right\|_F$$

where

$$\left[\mathbb{O}_{m,r} | \tilde{S}^{(r)}_{\mathcal{P}^*}\right] = \left[\begin{array}{c|cccc} 0 & a_1 & a_2 & 0 \\ 0 & 0 & a_1 & a_2 \\ 0 & a_3 & a_4 & 0 \\ 0 & 0 & a_3 & a_4 \\ 0 & a_5 & a_6 & 0 \\ 0 & 0 & a_5 & a_6 \end{array}\right]$$

and $a_i,\ i = 1, \ldots, 6$ are arbitrary parameters. The built-in minimization routine of Maple gives the next result :

$$\mathcal{S}(v') = f(\mathcal{P}, \mathcal{P}^*) = 1.8451526088542018 \cdot 10^{-15}$$

for

$$\{a_1, a_2, a_3, a_4, a_5, a_6\} = \{2.0, 2.666666666666668, 1.0, -3.5, 1.5, -7.0\}$$

in 16-digits software accuracy. □

Acknowledgements

We would like to thank Professor A. Feldstein and the referees for their valuable comments and suggestions that contributed to a significant improvement of the presentation of this paper.

References

[1] J. L. Barlow, More accurate bidiagonal reduction for computing the singular value decomposition, SIAM J. Matrix Anal. Appl. 23 (3) (2002) 761–798.

[2] S. Barnett, Greatest common divisor of several polynomials, in: Proc. Cambridge Philos. Soc., 1971.

[3] S. Barnett, Greatest common divisor from generalized sylvester matrices, in: Proc. Cambridge Philos. Soc., vol. 8, 1980.

[4] W. Blankiship, A new version of euclid algorithm, American Mathematics Monthly 70 (1963) 742–744.

[5] W. S. Brown, On euclid's algorithm and the computation of polynomials greatest common divisors, Journal of the Association for Computer Machinery 18 (4) (1971) 478–504.

[6] T. F. Chan, An improved algorithm for computing the singular value decomposition, ACM Trans. Math. Soft. 8 (1982) 72–83.

[7] R. M. Corless, P. M. Gianni, B. M. Trager, S. M. Watt, The singular value decomposition for polynomial systems, in: Proc. ISSAC '95, Quebec, Canada, 1995.

[8] B. N. Datta, Numerical Linear Algebra and Applications, Brooks/Cole Publishing Company - ITP, 1995.

[9] G. M. Diaz-Toca, L. Gonzalez-Vega, Computing greatest common divisors and squarefree decompositions through matrix methods: The parametric and approximate cases, Linear Algebra and its Applications 412 (2006) 222–246.

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

[11] S. Fatouros, N. Karcanias, Resultant properties of the gcd of many polynomials and a factorisation representation of gcd, Int. Journ. Control 76 (16) (2003) 1666–1683.

[12] L. Foster, Rank and null space calculations using matrix decomposition without column interchanges, Linear Algebra and its Applications 74 (1986) 47–71.

[13] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Num. Anal. Ser. B2 (1965) 205–224.

[14] G. Golub, C. V. Loan, Matrix Computations, 2nd ed., The John Hopkins University Press, Baltimore, London, 1989.

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

[16] T. Kailath, Linear Systems, Prentice Hall, Inc, Englewood Cliffs, New Jersey, 1980.

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

[18] N. Karcanias, S. Fatouros, M. Mitrouli, G. Halikias, Approximate greatest common divisor of many polynomials, generalised resultants and strength of approximation, Computers & Mathematics with Applications 51 (12) (2006) 1817–1830.

[19] N. Karcanias, G. Giannakopoulos, M. Hubbard, Almost zeros of a set of polynomials, Int. Journ. Control 40 (1984) 439–457.

[20] N. Karcanias, M. Mitrouli, A matrix pencil based numerical method for the computation of the gcd of polynomials, IEEE Trans. Autom. Cont. 39 (1994) 977–981.

[21] N. Karcanias, M. Mitrouli, Approximate algebraic computations of algebraic invariants, in: Symbolic methods in control systems analysis and design, vol. 56 of IEE Control Engin. Series, 1999, pp. 162–168.

[22] N. Karmarkar, Y. N. Lakshman, On approximate gcds of univariate polynomials, J. Symbolic Computation 26 (6) (1998) 653–666.

[23] J. Leventides, N. Karcanias, Genericity, generic system properties and generic values of invariants, Research Report SDCU 047, Control Engineering Centre, City University, London (May 1996).

[24] M. Marcus, M. Minc, A survey of matrix theory and matrix inequalities, Allyn and Bacon, Boston, 1964.

[25] M. Mitrouli, N. Karcanias, Computation of the gcd of polynomials using gaussian transformation and shifting, Int. Journ. Control 58 (1993) 211–228.

[26] M. Noda, T. Sasaki, Approximate gcd and its applications to ill-conditioned algebraic equations, Jour. of Comp. and Appl. Math. 38 (1991) 335–351.

[27] I. S. Pace, S. Barnett, Comparison of algorithms for calulation of g.c.d of polynomials, Int. Journ. Control 4 (2) (1973) 211–216.

[28] D. Rupprecht, An algorithm for computing certified approximate gcd of n univariate polynomials, Journ. of Pure and Applied Algebra 139 (1999) 255–284.

[29] S. Van Huffel, Partial singular value decomposition algorithm, Journ. of Comp. and Appl. Math. 33 (1990) 105–112.

[30] S. Van Huffel, J. Vandewalle, An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values, Journ. of Comp. and Appl. Math. 19 (1987) 313–330.

[31] A. I. G. Vardulakis, P. N. R. Stoyle, Generalized resultant theorem, IMA Journal of Applied Mathematics 22 (3) (1978) 331–335.

[32] W. M. Wonham, Linear Multivariable Control: A Geometric Approach, 2nd ed., Springer Verlag, New York, 1984.

[33] Z. Zeng, The approximate gcd of inexact polynomials, Part I: A univariate algorithm, manuscript (2004).