A resultant based computation of the Greatest Common Divisor of two polynomials

N. Karcanias , M. Mitrouli , S. Fatouros

Abstract— **The problem of finding the greatest common divisor (GCD) of a given polynomial set has interested mathematicians for a very long time and has widespread applications in several branches of Control Theory, Matrix Theory, Statistics, Network Theory etc. Since the existence of a common divisor of polynomials is a property that holds for specific sets and is not true generically, extra care is needed in the development of efficient numerical algorithms calculating correctly the required GCD. In the present paper, we study the application of a resultant-based computation of the GCD of two polynomials to two numerical methods, the ERES method and the Matrix Pencil method. A comparison of these two methods is performed and various numerical results are described.**

Keywords— **Polynomials, Resultant set, Greatest Common Divisor.**

I. Introduction

Some of the key problems of algebraic computations are the computation of the greatest common divisor (GCD), the computation of the least common multiple (LCM) of a set of polynomials and the computation of the factors of a polynomial. From the engineering applications in control theory viewpoint, the GCD is linked with the characterisation of zeros of representation whereas LCM is connected with the derivation of minimal representations of rational models. The problem of finding the GCD of a set $\mathcal{P}_{m,d}$, of m polynomials of $R[s]$ of maximal degree d, is a classical problem that has been considered before, see [5], [6], [8], [10], [2]. The numerical computation of GCD has been considered so far by transforming it to an equivalent problem of real matrix computations (see methods such as Extended Row Equivalence and Shifting (ERES) [6], Matrix Pencil see [5] and [10] for other methods). The advantage of real matrix computations is that we can discuss the problem of approximate solutions and thus introduce the notion of "approximate GCD". In several engineering computations it is useful to define an approximate GCD of the set within a specified accuracy. The ERES method carries out successfully the computation of approximate GCD. In [9] other methods for computing approximate GCD are also proposed. The problem of computing the LCM of polynomials has also widespread applications and requires implementation of algorithms computing the GCD.

In the present paper we examine a resultant-based computation of the GCD of two polynomials and we study its application to the ERES and the Matrix Pencil method.

Throughout the paper $R[s]$ denotes the ring of real polynomials. The symbol $\partial\{f(s)\}$ denotes the degree of a polynomial. $N_r(A)$ denotes the right null space of a matrix A.

Consider the two polynomials $a(s), b(s) \in R[s]$, $\partial\{a(s)\} = m$, monic and $\partial\{b(s)\} = n$, $n \leq m$, where

$$\begin{aligned} a(s) &= s^m + a_{m-1}s^{m-1} + \ldots + a_1 s + a_0 \\ b(s) &= b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1 s + b_0 \end{aligned} \tag{1}$$

The resultant of the two polynomials $S(a, b)$ is defined by

$$\begin{bmatrix} 1 & a_{m-1} & \ldots & \ldots & a_0 & 0 & \ldots & \ldots & 0 & 0 \\ 0 & 1 & \ldots & \ldots & a_1 & a_0 & 0 & \ldots & 0 & 0 \\ . & . & \ldots & \ldots & . & . & \ldots & \ldots & . & . \\ 0 & 0 & \ldots & 1 & . & . & . & \ldots & a_1 & a_0 \\ - & - & - & - & - & - & - & - & - & - \\ b_n & b_{n-1} & \ldots & \ldots & \ldots & b_0 & 0 & \ldots & 0 & 0 \\ 0 & b_n & \ldots & \ldots & \ldots & b_1 & b_0 & \ldots & 0 & 0 \\ . & . & \ldots & \ldots & . & . & \ldots & \ldots & . & . \\ 0 & 0 & \ldots & \ldots & b_n & b_{n-1} & \ldots & \ldots & b_1 & b_0 \end{bmatrix} \tag{2}$$

and clearly $S(a, b) \in R^{(n+m)\times(n+m)}$. With the pair $(a(s), b(s))$ we also define the **associated resultant set** of polynomials $\mathcal{P}_{a,b}$ as the polynomial coordinates of the vector

$$P_{a,b}(s) = S(a, b) \begin{bmatrix} 1 \\ s \\ . \\ \vdots \\ s^{n+m-1} \end{bmatrix} = S(a, b)\underline{e}_{n+m}(s) \tag{3}$$

The properties of GCD of $(a(s), b(s))$ in terms of the resultant are summarised below:

Theorem 1: [1] Let $a(s), b(s) \in R[s], \partial\{a(s)\} = m, \partial\{b(s)\} = n$, $m \geq n$ and let $z(s) = s^r + z_{r-1}s^{r-1} + \ldots + z_1 s + z_0$ be their GCD. The following properties hold true:
(i) $(a(s), b(s))$ is coprime, if and only if $\text{rank}(S(a, b)) = n + m$.
(ii) $r = \partial\{z(s)\} = m + n - \text{rank}(S(a, b))$.
(iii) If $S_H(a, b)$ is the row echelon form of $S(a, b)$, then the last non-vanishing row gives the coefficients of GCD of $(a(s), b(s))$.

II. The resultant ERES and the resultant Matrix Pencil methods

The computation of GCD may be performed using the standard methodologies of ERES, or Matrix Pencils. For

N. Karcanias is in Department of Electrical Electronic and Information Engineering, Control Engineering Center, City University, Northampton Square, London, EC1V 0HB, U.K.

M. Mitrouli is in Department of Mathematics University of Athens, Panepistimiopolis 15784, Athens, Greece. E-mail:mmitroul@math.uoa.gr

S. Fatouros is with the Control Engineering Center, City University, London

two polynomials, it will be shown that the above methodologies may be drastically simplified using properties of the resultant of two polynomials.

In the following, we will deploy the above two methods for computing the GCD of two polynomial using simpler procedures than the two general procedures mentioned before. More specifically, we shall explore the special properties of computation of GCD of two polynomials using both the ERES framework [6], [8] and the Matrix Pencil approach [5].

The ERES method applied on the basis matrix of the pair $(a(s), b(s))$ involves a number of triangularisations and shifting operations and eventually leads to the GCD which is defined by the nonzero row of a unity rank matrix. If we use the resultant set $\mathcal{P}_{a,b}$ a simplified computational procedure is introduced by the following result.

Corollary (1): Let $S(a,b)$ be the resultant of the pair of polynomials $(a(s), b(s)), \rho = \text{rank}(S(a,b))$ and let $S^*(a,b)$ denote the upper triangular form of $S(a,b)$ obtained under the Gauss row transformations, i.e.

$$S^*(a,b) = \begin{bmatrix} x & x & \dots & \dots & x & \dots & \dots & x \\ 0 & x & \dots & \dots & x & \dots & \dots & x \\ \vdots & \vdots & \dots & \dots & \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & \dots & x & \dots & \dots & x \\ \vdots & \vdots & \dots & \dots & \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & \dots & 0 & \dots & \dots & 0 \end{bmatrix} \quad (4)$$

where the x leading element of each nonzero row is also nonzero. The nonzero elements of the last nonzero row of $S^*(a,b)$ define the coefficients of GCD in reverse order.

Proof: By reducing $S(a,b)$ to the $S^*(a,b)$ form we have completed the first step in deriving the echelon form. The following steps involve making the first nonzero element of each of the nonzero rows (Pivots), 1, and then with row operations eliminating all elements above them. However, such a procedure does not affect the elements of the last nonzero row (apart from scaling by the leading coefficient). Thus, the resulting echelon form has the last nonzero row equivalent modulo scaling to that of $S^*(a,b)$ and by Theorem (1) the result follows.

Remark (1): The above result provides a simpler way for computing the GCD of two polynomials and implies that only the first step of the ERES algorithm is needed (a single triangularization with no shifting and further triangularisations) when the resultant set is used. The increase in the number of polynomials (from two to $n+m$), improves the speed of the algorithm since the additional polynomials are simply defined from the original set.

The above procedure will be referred to as **triangularization of the resultant set**. Clearly, partial pivoting may be used in the Gaussian transformation to improve numerical stability.

Algorithm res-ERES

STEP (1): Form the resultant matrix $S(a,b)$.
STEP (2): Specify $r = \partial\{z(s)\} = m + n - \text{rank}(S(a,b))$.
STEP (3): Apply Gauss row transformations with partial pivoting and transform $S(a,b)$ to $S^*(a,b)$.
STEP (4): The coefficients of the last nonzero row of $S^*(a,b)$ define the coefficients of the GCD in reverse order.

Implementation of the algorithm

Computational Complexity: Since the algorithm uses Gaussian elimination with partial pivoting the complexity of the algorithm will be $O(\frac{k^3}{3})$, where k is the dimension of the matrix $S(a,b)$.

Error Analysis: The following Theorem holds.

Theorem 2: Let $S(a,b)$ a given matrix of order k, If we perform Gaussian Elimination with partial pivoting using floating point arithmetic with unit roundoff u, the following relation holds:

$$L \cdot S^*(a,b) = S(a,b) + E, \quad ||E||_\infty \leq k^3 \cdot u \cdot p \cdot ||S(a,b)||_\infty$$

where L a lower triangular matrix with units on the diagonal, and p is the growth factor of the Gaussian elimination.

Remark (2): The triangularisation of the resultant set and thus the computation of the GCD may also be achieved by QR factorisation using orthogonal transformations. This provides an alternative procedure for computing GCD than that based on Gaussian transformations.

An alternative procedure for computing the GCD may now be indroduced by using the framework of the Matrix Pencil approach for the computation of GCD [5]. Let $\mathcal{P}_{m,d} = \{p_i(s) : p_i(s) \in \mathcal{R}[s],\ i = 1, 2, \dots, m,\ d_i = deg\{p_i(s)\},$

$$d = max\{d_i,\ i = 1, 2, \dots, m\}\}$$

be the set of m polynomials of $\mathcal{R}[s]$ of maximal degree d.

For any $\mathcal{P}_{m,d}$ set we define a vector representative $\underline{p}_m(s)$ and a basis matrix P_m by

$$\underline{p}_m(s) = [p_1(s), \dots, p_m(s)]^t = [\underline{p}_0, \underline{p}_1, \dots, \underline{p}_d]\, \underline{e}_d(s) = P_m \underline{e}_d(s) \quad (1)$$

where $P_m \in R^{k \times \mu},\ \mu = d - p + 1,\ \underline{e}_d(s) = [1, s, \dots, s^d]^t$.

$\text{rank}(P_m) = v < d + 1$, we define a basis $M \in R^{k \times \mu}, \mu = d - p + 1$, for the right null space of P_m denoted by $\mathcal{N}_r\{P\}$ and denote by $M_1, M_2 \in R^{d \times \mu}$ the matrices obtained from M by deleting the last, first row of M respectively. The pencil $Z(s) = sM_1 - M_2$ is known as the **GCD pencil** of the set $\mathcal{P}$ and its properties are summarised below:

Theorem 3: [5] The GCD pencil $Z(s) = sM_1 - M_2$ has the following properties:
(i) The set of Kronecker invariants consists of row minimal indices (rmi) and possibly finite elementary divisors (fed).

(ii) The zero polynomial of ($Z(s)$ product of all fed) is the GCD of the set $\mathcal{P}$.

The above result provides the basis for the matrix pencil approach. The fact that $Z(s)$ may have nonzero rmi (generic case from dimensions) requires further analysis for the computation of the zero polynomial. For the special case where $\mathcal{P} = (a(s), b(s))$ and we use the associated resultant set, the above result takes the following form:

Theorem 4: Consider the pair of polynomials $(a(s), b(s))$ with resultant polynomial set $\mathcal{P}_{a,b}$ and associated basis matrix the resultant $S(a, b)$. Then,
(i) The polynomials $(a(s), b(s))$ are coprime, if and only if $\mathcal{N}_r\{S(a, b)\} = \{0\}$.
(ii) The polynomials $(a(s), b(s))$ have a nontrivial GCD $(t1)$ if and only if $\mathcal{N}_r\{S(a, b)\} \neq \{0\}$. In this case $\partial\{z(s)\} = \dim \mathcal{N}_r\{S(a, b)\}$ and for the GCD pencil $Z(s) = sM_1 - M_2$ we have the properties:
(a) $Z(s)$ is characterised by fed and posibly only zero rmi.
(b) $Z(s)$ may be expressed as

$$Z(s) = sM_1 - M_2 = \widetilde{M}(sI - \widetilde{Z}) \tag{5}$$

where the characteristic polynomial of $\widetilde{Z}$ defines the monic GCD $z(s)$.

Proof:
(i) This part follows directly from Theorem (1).
(ii) Clearly the GCD is nontrivial when $\mathcal{N}_r\{S(a, b)\} \neq \{0\}$. In this case $sM_1 - M_2$ is defined and by Theorem (3) it has rmi and always fed, as this follows from part (i). The pencil $Z(s)$ has the following general Kronecker decomposition

$$Z(s) = R \left[\begin{array}{c|c} \cdots & \cdots \\ 0 & 0 \\ \cdots & \cdots \\ L_n(s) & 0 \\ 0 & sI - A \end{array}\right], \ Q \tag{6}$$

where $L_n(s)$ is the set of blocks associated with the nonzero rmi and $sT - A$ characterise the finite zeros. Cleraly, if $L_n(s)$ exist, then $\partial = \partial\{z(s)\} = \partial\{|sI - A|\} < \dim \mathcal{N}_r\{S(a, b)\}$ and by Theorem (1) we are led to a contradiction. Thus, $Z(s)$ has nonzero rmi and thus its structure as expressed by the Kronecker decomposition becomes

$$Z(s) = R \left[\begin{array}{c} 0 \\ \cdots \\ sI - A \end{array}\right], \ Q \tag{7}$$

By partitioning R according to the partitioning of the Kronecker form we have

$$Z(s) = [R', \quad R] \left[\begin{array}{c} 0 \\ \cdots \\ sI - A \end{array}\right], \ Q = \bar{R}(sI - A)Q \tag{8}$$

and part (b) of (ii) is established.

Remark (3): If $\mathcal{N}_r(S(a, b)) \neq \{0\}$ and $Z(s)$ is the GCD pencil of $\mathcal{P}_{a,b}$, then any minor of maximal order of $Z(s)$ which is not identically zero defines the GCD of $(a(s), b(s))$.

Algorithm res-MP

STEP (1): Form the resultant matrix $S(a, b)$.
STEP (2): Specify a basis M for $\mathcal{N}_r(S(a, b))$.
STEP (3): Form the GCD pencil $Z(s)$.
STEP (4): Compute any minor of maximal order of $Z(s)$ which is not zero. From this minor define the coefficients of the GCD.

Implementation of the algorithm

Computational Complexity: The determination of the basis matrix will be accomplished numerically using the Singular Value Decomposition (SVD) of matrix $S(a, b)$. The complexity of this computation will be $O(k^3)$, where k is the dimension of the matrix $S(a, b)$. The specification of a nonzero maximal minor can be done symbolically without requiring additional flops.

Error Analysis: The computation of the right null space of $S(a, b)$ using the SVD is a stable process which guarantees the stability of the computation.

III. Numerical results

The results of the above section lead to two new alternative procedures for computing the GCD of a pair $(a(s), b(s))$, which are summarised below:

Computation of GCD of a pair of polynomials

For the pair $(a(s), b(s))$ with resultant set $\mathcal{P}_{a,b}$ and resultant matrix $S(a, b)$ their GCD is defined by the following two alternative by equivalent new procedures:

Procedure (A): Reduce $S(a, b)$ to upper triangular form by elementary Gaussian tranformations or orthogonal transformations and then define the coefficients of GCD from the nontrivial elements of the last nonzero row.

Procedure (B): Compute the GCD pencil from $\mathcal{N}_r\{S(a, b)\}$ using SVD and then define GCD as any maximal order nontrivial minor of this pencil.

The above procedures for GCD evaluation were programmed in MATLAB environment and tested on a Pentium machine over several sets of polynomials $P_{m,d}$ characterised by various properties. Next, for each set of data we present the exact GCD and a table summarising the achieved results. In the first column of the table the applied method is mentioned. Specifically, the notation ERES denotes the standard ERES method, Res-ERES denotes the modification of the ERES using the resultant matrix, Orthogonal denotes the application of QR factorisation, MP denotes the Matrix Pencil method and Res-MP denotes the modification of the MP method using the resultant matrix. In the second column of the table the obtained relative error in the final result is written; in the third column the required accuracy of the method is mentioned for several

intermediate calculations performed by each method, and finally in the fifth column the total number of floating point operations (flops) , counted using an appropriate MATLAB function is given.

Example (1):
Consider the two polynomials $a(s), b(s) \in R[s]$, $\partial\{a(s)\} = 4$, monic and $\partial\{b(s)\} = 3$, $3 \leq 4$, where

$$a(s) = s^4 + s^3 + 12s^2 + s + 11, \quad b(s) = 2s^3 + 5s^2 + 2s + 5$$

Exact $GCD = s^2 + 1$

METHOD	ERROR	ACCUR.	FLOPS
ERES	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	826
Res-ERES	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	464
Orthogonal	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	1328
MP	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	3283
Res-MP	Exact Formula	$\epsilon = 10^{-15}$	5529

Table 1

Example (2):
Consider the two polynomials $a(s), b(s) \in R[s]$, $\partial\{a(s)\} = 4$, monic and $\partial\{b(s)\} = 3$, $2 \leq 4$, where

$$a(s) = s^4 - 59s^3 - 4560s^2 + 45500s + 50000, \quad b(s) = s^2 + 31s + 30$$

Exact $GCD = s + 1$

METHOD	ERROR	ACCUR.	FLOPS
ERES	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	1029
Res-ERES	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	312
Orthogonal	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	905
MP	$\leq 10^{-16}$	$\epsilon = 10^{-15}$	3318
Res-MP	Exact Formula	$\epsilon = 10^{-15}$	3282

Table 2

Remark (4): From the above tables we see that the resultant ERES method attains the less number of required floating point operations. Actually, for two polynomials the ERES has the worst behaviour [8] and thus the resultant ERES is a great improvement. This difference is apparent only if we don't compute the rank of the resultant matrix. If this computation is done then since the dimension of the resultant matrix is large, many flops will be required and the method will not be efficient. Thus we will perform the full triangularisation of the resultant matrix and then we will specify as GCD the first nonzero row from the bottom of the modified matrix. The orthogonal method uses QR for the triangularization and thus a remarkable number of flops is required. The difference between the MP method and the resultant MP method is not so great since both methods requires the computation of null spaces. The resultant MP requires the computation of the null space of a much larger matrix (the resultant matrix) and thus it has more flops. Actually all the required flops in the resultant MP method are applied for the computation of the null space since the rest computations can be performed symbolically. From the above tables we see that the relative error for all the above methods is very small i.e. the accuracy of the computed results is very good. Specifically the Res-ERES method applies only Gaussian transformations with partial pivoting which guarantees the stability of the applied algorithm.

References

[1] S. Barnett, *Matrices Methods and Applications*, Clarendon Press, Oxford, 1990.

[2] R. R. Bitmead, S. Y. Kung, B. D. O. Anderson and T. Kailath, *Greatest Common Divisors via Generalised Sylvester and Bezout matrices*, IEEE Trans. Autom. Contr., Vol. AC-23, No 6, (1978) pp. 1043-1047.

[3] N. Karcanias and M. Mitrouli, *Normal Factorisation of polynomials and its symbolic computation*, Control 2000 Conference, Cambridge 4-7 September 2000.

[4] N. Karcanias and M. Mitrouli, *Numerical computation of the least common multiple of a set of polynomials*, Reliable Computing, Issue 4, Vol. 6 (2000) pp. 439-457.

[5] N. Karcanias and M. Mitrouli , *A Matrix Pencil Based Numerical Method for the Computation of the GCD of Polynomials*, IEEE Trans. Autom. Cont., Vol. 39 (1994), 977-981.

[6] M. Mitrouli and N. Karcanias, *Computation of the GCD of polynomials using Gaussian transformation and shifting* Int. Journ. Control 58 (1993), 211-228.

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

[8] M. Mitrouli, N. Karcanias and C. Koukouvinos , *Further numerical aspects of the ERES algorithm for the computation of the greatest common divisor of polynomials and comparison with other existing methodologies*, Utilitas Mathematica 50 (1996), 65-84.

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

[10] I. S. Pace and S. Barnett , *Comparison of algorithms for calculation of GCD of polynomials*, Int. Journ. System Scien. 4 (1973), 211-226.

[11] A. I. G. Vardulakis and P. N. R. Stoyle, *Generalized resultant theorem*, Journ. of IMA 22 (1978), 331-335.