A Hybrid Approach for Normal Factorization of Polynomials

Nicos Karcanias[1], Marilena Mitrouli[2,⋆], and Dimitrios Triantafyllou[2]

[1] Control Engineering Centre, School of Engineering and Mathematical Science, City University Northampton Square, London EC1V 0HV, UK
N.Karcanias@city.ac.uk
[2] Department of Mathematics, University of Athens, Panepistemiopolis 15784, Athens, Greece
{mmitroul, dtriant}@math.uoa.gr

Abstract. The problem considered here is an integral part of computations for algebraic control problems. The paper introduces the notion of normal factorization of polynomials and then presents a new hybrid algorithm for the computation of this factorization. The advantage of such a factorization is that it handles the determination of multiplicities and produces factors of lower degree and with distinct roots. The presented algorithm has the ability to specify the roots of the polynomials without computing them explicitly. Also it may be used for investigating the clustering of the roots of the polynomials. The developed procedure is based on the use of algorithms determining the greatest common divisor of polynomials. The algorithm can be implemented symbolically for the specification of well separated roots and numerically for the specification of roots belonging in approximate clusters.

Keywords: Polynomials, factorization, approximate computations, greatest common divisor.

1 Statement of the Problem

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. The problem of finding the GCD of a set $\mathcal{P}_{m,d}$, of m polynomials of $\Re[s]$ of maximal degree d, is a classical problem that has been considered before, see [2, 6, 8]. 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 [2] and [8] 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. This

⋆ This research was financially supported by PENED 03ED740 (Greek Secretary of Research and Technology).

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 399–406, 2006.

approximate GCD can be used in several applications such as the definition of the almost zeros of a given polynomial system see [1]. The ERES method carries out succesfully the computation of approximate GCD. In [7] other methods for computing approximate GCD are also proposed.

The problem of factorizing polynomials is within the general framework of algebraic computations and it is crucial for determining Smith forms, solving Diofantine equations and general polynomials, or rational matrix equations. Of special interest is the problem of factorizing polynomials without resorting to root finding, as well as handling issues of approximate factorizations, when there is uncertainty on the exact values of the coefficients. This paper deals with the definition, the symbolic and numerical aspects of computation of a special factorization of polynomials, which is within the general factorization of polynomial, and can be performed without resorting to procedures based on finding roots.

For every polynomial $f(s) \in \Re[s]$ there exist positive integers $d_1, ..., d_\sigma$ where $d_1 > d_2 > ... > d_\sigma \geq 1$, such that f(s) may be expressed as

$$f(s) = e_1(s)^{d_1} e_2(s)^{d_2} ... e_\sigma(s)^{d_\sigma} \tag{1}$$

where the polynomials $\{e_1(s), e_2(s), \ldots, e_\sigma(s)\}$ are pairwise coprime and the polynomial $\hat{f}(s) = e_1(s)e_2(s)...e_\sigma(s)$ has distinct roots. □

The above factorization will be called a *normal factorization* of f(s) [3], $(d_i,\ i \in \widetilde{\sigma})$ will be called the *power set* and the set of polynomials $\{e_i(s),\ i \in \widetilde{\sigma}\}$ will be referred to as the *base* of the factorization. Such factorizations indicate the clustering of sets of roots, as this is indicated by the power set. Computing such factorizations is the problem considered in this paper. More precisely, we develop an algorithm specifying the set $\{e_i(s), i = 1, \ldots, k\}$ of polynomials and the corresponding degrees $\{d_i(s), i = 1, \ldots, k\}$. The advantage of such a factorization is that it handles the determination of multiplicities and produces factors of lower degree and with distinct roots. The use of algorithms for computing GCD is central to the current method and methods such as ERES [6], Matrix Pencil [2] etc may be used for this computation, and thus frequently lead to "approximate" rather than exact evaluations of the GCD. It is clear that with the notion of "approximate GCD" we also have the problem of defining "almost factorizations"of a polynomial. For approximate values of the GCD we define the order of approximation as a factor of the polynomial. The new computational procedure is demonstrated finally by a number of examples.

Throughout the paper $\Re[s]$ denotes the ring of real polynomials. The symbol $\partial\{f(s)\}$ denotes the degree of a polynomial. If a property is said to be true for $i \in \widetilde{n},\ \ n \in \mathcal{Z}^+$, this means it is true for all $1 \leq i \leq n$. If $g_1(s), g_2(s) \in \Re[s]$ and $g_1(s)$ divides $g_2(s)$ then we denote it by $g_1(s)/g_2(s)$.

2 Background of Approximate Algebraic Computations

In engineering computations, the numerical procedures must have the ability to work on data with numerical inaccuracies and thus lead to "approximate

algebraic computations". In the sequel we study the notions of "approximate GCD" and "approximate normal factorization".

Let $\mathcal{P}_{m,d} = \{p_i(s) : p_i(s) \in \mathcal{R}[s],\ i = 1, 2, \ldots, m,\ d_i = deg\{p_i(s)\},\ d = max\{d_i,\ i = 1, 2, \ldots, m\}\}$ be the set of m polynomials of $\mathcal{R}[s]$ of maximal degree d. Given a set of polynomials $t_i(s) \in \mathcal{R}[s]$, $i = 1, 2, \ldots, k$, define a numerical procedure for the computation of their "approximate" GCD, and associated factorization avoiding root finding.

Approximate GCD

The notion of the GCD of many polynomials is characterized by the property that its computation is nongeneric; in fact, the set of polynomials for which a nontrivial GCD ($\neq 1$) may be defined is a set of measure zero. However, the need for defining notions such as "almost zeros" and "approximate GCD" has been recognized as important in many applications. Methods computing the GCD of the set $\mathcal{P}$, which deploy relaxation of the exact conditions for GCD evaluation, such as the ERES method [6] lead to expressions for the "approximate GCD". In [7] the "approximate GCD" problem has been considered in the context of Euclidean division and for the case of two polynomials. Recently Karcanias etc [4], introduced formally the notion of the "approximate GCD" and then developed a computational procedure that allows the evaluation of how good is the given "approximate GCD" by estimating its strength of approximation.

2.1 Algorithm Approximate GCD

The Resultant Matrix Pencil Algorithm [5] appropriately modified allows the capturing of "approximate GCDs" and this is summarized by the following algorithm applied on the Modified Sylvester Resultant [9] of the given polynomial set and which exploits the notion of "near nullspace" of a matrix. The basic steps are:

Step1: Define a threshold $t > 0$. Apply the SVD algorithm to S^* to define a basis $\widetilde{M}$for the right 'near nullspace' of the Sylvester matrix S^* constructed from the given polynomial set.
Step2: Define the GCD Matrix Pencil $\widetilde{Z}(s) = s\widetilde{M}_1 - \widetilde{M}_2$,where $\widetilde{M}_1$ and $\widetilde{M}_2$ are the matrices obtained from $\widetilde{M}$ by deleting the last and the first row of $\widetilde{M}$ respectively.
Step 3: Construct a matrix with elements all nonzero minor determinants $d(s)$ of $\widetilde{Z}(s)$ and compute its basis $\widetilde{B}$.
Step 4: Apply the SVD algorithm to matrix $\widetilde{B}$: $\widetilde{B} = \widetilde{U}^T \widetilde{\Sigma} \widetilde{V}$. The corresponding to the largest singular value column of $\widetilde{V}$ defines the approximate GCD.
Step 5: Compute the angle of the spaces $\widetilde{M}_1$ and $\widetilde{M}_2$ as an indicator for the strength of approximation.

The next example demonstrates the implementation of the above algorithm to a computer.

Example 1. Consider the following set of polynomials :

$\{p_1(s) = s^3 - 6\,s^2 + 11\,s - 6, p_2(s) = s^2 - 3\,s + 2, p_3(s) = s^2 - 2\,s + 0.9999,$ $p_4(s) = s^2 - 1\}$

The above set of polynomials does not have an exact GCD. We have the following modified Sylvester matrix :

$$S^* = \begin{bmatrix} 1 & -6 & 11 & -6 & 0 \\ 1 & -3 & 2 & 0 & 0 \\ 1 & -2 & 0.9999 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -6 & 11 & -6 \\ 0 & 1 & -3 & 2 & 0 \\ 0 & 1 & -2 & 0.9999 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -3 & 2 \\ 0 & 0 & 1 & -2 & 0.9999 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \in \Re^{11\times 5}$$

Applying the SVD algorithm to S^* we obtain as numerical results the following singular values and the matrix V are :

$$\Sigma = \begin{bmatrix} 19.2719 & 0 & 0 & 0 & 0 \\ 0 & 8.6015 & 0 & 0 & 0 \\ 0 & 0 & 2.6935 & 0 & 0 \\ 0 & 0 & 0 & 1.1627 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 \end{bmatrix} \; and V = \begin{bmatrix} -0.0426 & -0.1206 & -0.4488 & 0.7630 & 0.4472 \\ 0.2834 & 0.5225 & 0.6347 & 0.2096 & 0.4472 \\ -0.6642 & -0.4355 & 0.3864 & -0.1408 & 0.4472 \\ 0.6516 & -0.4942 & -0.0828 & -0.3526 & 0.4472 \\ -0.2282 & 0.5279 & -0.4895 & -0.4793 & 0.4472 \end{bmatrix}$$

The tolerance we use is 10^{-15}. The last singular value is 10^{-4}. If the threshold t=0.0002 then the last singular value is smaller than t. The corresponding column of V is the last one and so $\widetilde{M}$ is :

$$\widetilde{M} = \begin{bmatrix} 0.4472 \\ 0.4472 \\ 0.4472 \\ 0.4472 \\ 0.4472 \end{bmatrix}, \widetilde{M}_1 = \begin{bmatrix} 0.4472 \\ 0.4472 \\ 0.4472 \\ 0.4472 \end{bmatrix}, \widetilde{M}_2 = \begin{bmatrix} 0.4472 \\ 0.4472 \\ 0.4472 \\ 0.4472 \end{bmatrix}$$

The angle of $\widetilde{M_1}$ and $\widetilde{M_2}$ acts as an indicator for the approximation. Their angle is: 0 (very good approximation).

The GCD Pencil and its basis matrix are correspondingly :

$$\widetilde{Z}(s) = s\widetilde{M}_1 - \widetilde{M}_2 = \begin{bmatrix} 0.4472s - 0.4472 \\ 0.4472s - 0.4472 \\ 0.4472s - 0.4472 \\ 0.4472s - 0.4472 \end{bmatrix} and \widetilde{B} = \begin{bmatrix} 0.4472 & -0.4472 \\ 0.4472 & -0.4472 \\ 0.4472 & -0.4472 \\ 0.4472 & -0.4472 \end{bmatrix}$$

Applying the SVD algorithm to $\widetilde{B}$ we have :

$$\widetilde{\Sigma} = \begin{bmatrix} 1.2649 & 0 \\ 0 & 0 \end{bmatrix} \; and \; \widetilde{V} = \begin{bmatrix} 0.7071 & -0.7071 \\ -0.7071 & -0.7071 \end{bmatrix}$$

The maximal singular value is the first one and the corresponding column of V is also the first. This columns gives the coefficients of the GCD : 0.7071 and -0.7071, which divided by 0.7071 give 1 and -1. So the GCD is s-1.

ϵ-Normal Factorization
Next we introduce the definition of the approximate normal factorization with accuracy ϵ, or the ϵ-normal factorization.

For every polynomial $f(s) \in \Re[s]$ there exist positive integers $d_1, ..., d_\sigma$ where $d_1 > d_2 > ... > d_\sigma \geq 1$, such that f(s) may be expressed as

$$f(s) = \prod_{1}^{\sigma} e_i(s)^{d_i} + \Delta\ P(x) \tag{2}$$

where the polynomials $\{e_1(s), e_2(s), \ldots, e_\sigma(s)\}$ are pairwise coprime , the polynomial $\hat{f}(s) = e_1(s)e_2(s)...e_\sigma(s)$ has distinct roots and $||\Delta\ P|| \leq \epsilon$, where ϵ a selected accuracy for the computation. □

3 The Algorithm

Let $f(s) \in \Re[s]$ be a given polynomial. The following algorithm computes the normal factorization of $f(s)$. More specifically $f(s) = e_1(s)^{d_1} e_2(s)^{d_2} ... e_\sigma(s)^{d_\sigma}$.

3.1 Algorithm NF

STEP 0: Compute the GCD of $\{f^{(0)}(s), f^{(1)}(s)\}$.
If the polynomials are coprime
$f^{(0)}(s)$ has well separated roots. Perform numerically or symbolically STEPS 1 - 4
Else
compute numerically the approximate GCD.

STEP 1: Construct the associated set of $f(s)$
$G_f^{d_1} = \{g_0(s), g_1(s), ..., g_{d_1-1}(s), g_{d_1}(s)\}$, :
$g_k(s) = \text{GCD}\{f^{(0)}(s), f^{(1)}(s), ..., f^{(k)}(s)\}$
where the index d_1 is defined from
$d_1 = min_k g_k(s) = 1$

STEP 2: Define the prime set of $f(s)$
$\mathcal{T}_f = \{t_1(s), t_2(s), ..., t_{d_1}(s)\}$, where
$t_i(s) = \frac{g_{i-1}(s)}{g_i(s)}, \quad i = 1, \ldots, d_1$

STEP 3: Define the factorization set of $f(s)$
$P_f = \{p_i(s), i = 1, 2, \ldots, d_1\}$, where
$p_j(s) = \frac{t_j(s)}{t_{j+1}(s)}, \quad j = d_1, d_{1-1}, \ldots, 1, \quad t_{d_1+1} = 1$

STEP 4: From $\mathcal{T}_f$ construct the Ferrer diagram
Specify the index set
$\mathcal{I}_f = \{(d_i, v_i),\ i \in \widetilde{\sigma},\ d_1 > d_2 > ... > d_\sigma\}$
Form the normal factorization $f(s) = e_1(s)^{d_1} ... e_\sigma(s)^{d_\sigma}$ as follows:

Construct the essential factorization $\mathcal{P}_f^*$ of $f(s)$
$\mathcal{P}_f^* = \{p_{d_i}(s), \ \ i = 1, 2, ..., \sigma, \ d_1 > d_2 > ... > d_\sigma \geq 1\}$
base=$\{e_i(s) = p_{d_i}(s) : p_{d_i}(s) \in \mathcal{P}_f^*, \ \ i = 1, 2, \ldots, \sigma\}$
power set= $\{d_i, \ i = 1, 2, \ldots, \sigma\}$
v_i=number of all e.d. of $f(s)$ over $\mathcal{C}$ with degree d_i.

3.2 Numerical-Symbolical Results

The above method for NF evaluation has a hybrid nature. Its numerical part was programmed in Matlab. The complexity is of order concerning mostly the execution of the Matrix Pencil algorithm [2]. The algorithm was tested on a Pentium machine over several polynomials characterized by various properties. The symbolical implementation of the method was performed using Maple. According to the nature of the examples the numerical or the symbolical implementation will be chosen.

Example 2.
Compute the normal factorization of the following polynomial with real coefficients:
$f(s) = s^7 - 24.35s^6 + 225.2s^5 - 1021.225s^4 + 2526.7875s^3 - 3494.896875s^2 + 2544.75s - 761.4$

Implementing the algorithm numerically with internal required tolerance for the GCD computation of order $O(10^{-8})$ we obtain the following factors: $(s-1.5)^4(s-2.35)(s-8)^2$ (real roots well separated). The same result gives also the symbolical implementation of the algorithm with increase of the required time.

Example 3.
This example demonstrates the ability of the method to distinguish between "almost equal factors". Let us suppose that we are given the following polynomial:

$$f(s) = (s+1)^2(s+5)^2(s+3)(s+\epsilon)^k$$

(i) For $\epsilon = 1.001$ and $k = 1$ implementing the algorithm numerically with internal required accuracy for the GCD computation of order $O(10^{-12})$ we obtain the following factors : $(s^2+6s+5)^2(s^2+4.001s+3.003)$ We see that the first factor contains the product of $(s+1)(s+5)$ whereas the second factor contains the product of $(s+1.001)(s+3)$. If we require to know the exact values of the roots of $f(s)$ we can now proceed computing the roots of the appearing factors which as we can see are of rather lower degree i.e. 2nd degree each instead of 6th degree that is the original polynomial.
(ii) For $\epsilon = 1.001$ and $k = 2$ implementing the algorithm numerically with internal required accuracy for the GCD computation of order $O(10^{-12})$ we obtain the following factors : $(s^3+7.001s^2+11.006s+5.005)^2(s+3)$ We see that the first factor contains the product of $(s+1)(s+5)(s+1.001)$.
(iii) For $\epsilon = 1.000001$ and $k = 1$ implementing the algorithm numerically with internal required accuracy for the GCD computation of order $O(10^{-12})$ using

he modified LU or QR factorization we obtain the following factors : $(s + 1.00000029520885)^3(s+5)^2(s+3)$ We see that the product $(s+1)^2(s+1.000001)$ is the six digit approximation of the first appearing factor. Thus 1.00000029520885 is the approximate value of the three roots of $f(s)$ that they belong to the same cluster.

If we have polynomials with roots belonging in same clusters we can apply the following Criterion:

The Clustering Criterion [3]: A polynomial $f(s) \in \mathcal{R}(s)$ has distinct roots if and only if $f(s)$ and $f^{(1)}(s)$ are coprime.

When we want to apply the numerical method for polynomials of rather high degree problems might appear due to rounding errors since it will be required the computation of several derivatives and their GCDs. Thus for such cases might be better the symbolical computation as illustrated by the following example:

Example 4.
Compute the normal factorization of the following polynomial:

$$f(s) = (s-2)^{15}(s-4)^5$$

Applying algorithm NF symbolically in Maple we get exactly as basic factors of the above expanded polynomial the terms: $(s-2)$ and $(s-4)$ with degrees 15 and 5 respectively. From this example it is evident that implementing the method symbolically we can determine the normal factorization of any polynomial of any degree with well separated roots.

Example 5. Specify the root clustering of the following polynomial:

$$f(s) = s^3 - 5.000001s^2 + 7.000004s - 3.000003$$

Applying the Clustering Criterion we get that $\{f(s), f^{(1)}(s)\}$ have "almost common factor" the polynomial $s - 1.0000005$ whereas $\{f(s), f^{(1)}(s), f^{(2)}(s)\}$ are coprime. Thus $f(s)$ has an "almost common factor" of degree 2 and therefore its 2 roots belong in the same cluster. A representative of this cluster is the value 1.0000005.Thus we attain the following almost factorization: $f(s) = (s - 1.0000005)^2(s + \lambda)$ where λ is defined by division or almost factorization. Note that theoretically the above polynomial is the product of the following factors: $f(s) = (s-1)(s-1.000001)(s-3)$. Thus $f(s)$ has "almost common factors" (or "almost zeros"). Using the Clustering Criterion we determined them. Applying the approximate resultant matrix pencil we get the the factor $s - 0.99999949999911$ of degree 2.

Root clustering can also be applied to polynomials with complex conjugate roots.

4 Conclusions

An algorithm achieving the normal factorization of polynomials has been introduced. The algorithm may be worked out using only algebraic tools, such

as determination of GCD and symbolic computation of derivatives. The normal factorization can be considered as a first step into the factorization of polynomials, where multiplicities are computed first and then root finding is reduced to smaller degree and distinct root polynomials. The nature of the presented algorithm is hybrid. It can be applied symbolically or numerically according to the given data and the required results. The comparison of the current approach with other procedures that use different approximate GCD computation techniques is under study. The current approach also allows the study of root clustering, since the approximate GCD allows the grouping of roots which are close to each other.

References

1. Karcanias, N., Giannakopoulos, C., Hubbard, M.: Almost zeros of a set of polynomials of $\mathcal{R}[s]$. Int. J. Control, **38**, (1983) 1213–1238.
2. Karcanias, N., Mitrouli M.: A Matrix Pencil Based Numerical Method for the Computation of the GCD of Polynomials. IEEE Trans. Autom. Cont., **39**, (1994) 977–981.
3. Karcanias, N., Mitrouli, M.: Normal Factorization of Polynomials and Computational Issues. Computers and Mathematics with Applications, **45**, (2003) 229–245.
4. Karcanias, N., Mitrouli, M., Fatouros, S., Halikias, G.: Approximate Greatest Common Divisor of many polynomials and generalised resultants. Computers and Mathematics with Applications, to appear (2006).
5. Karcanias, N., Mitrouli, M., Fatouros, S.: A resultant based computation of the Greatest Common Divisor of two polynomials. Proc. of 11th IEEE Med. Conf on Control and Automation, Rodos Palace Hotel, Rhodes, Greece, (2003b).
6. Mitrouli, Mm., Karcanias, N.: Computation of the GCD of polynomials using Gaussian transformation and shifting. Int. Journ. Control, **58**, (1993) 211–228.
7. Noda, M., Sasaki, T.: Approximate GCD and its applications to ill-conditioned algebraic equations. Jour. of Comp. and Appl. Math., **38**, (1991) 335–351.
8. Pace, I., S., Barnett, S.: Comparison of algorithms for calculation of GCD of polynomials. Int. Journ. System Scien., **4**, (1973) 211–226.
9. Triantafyllou, D., Mitrouli, M.: Two Resultant Based Methods Computing the Greatest Common Divisor of Two Polynomials. In: Li, Z., Vulkov, L., Wasniewski, J. (eds.): Numerical Analysis and Its Applications. Lecture Notes in Computer Science. Vol. 3401, (2005) 519–526.
10. Turnbull, H., W., Aitken, A., C.: An Introduction to the Theory of Canonical Matrices. Dover Publ., New York (1961).