Bezout matrices, Subresultants and Parameters

Jounaidi Abdeljaoued, Gema M. Diaz–Toca and Laureano Gonzalez–Vega

Keywords. Subresultants, Bezoutians.

1. Introduction

The problem of computing the greatest common divisor of two univariate polynomials in $\mathbb{D}[x]$, with $\mathbb{D}$ a field or an integral domain, is one of the cornerstones in Computer Algebra. When $\mathbb{D}$ is an integral domain we shall talk about a greatest common divisor as a polynomial in $\mathbb{D}[x]$ which is a greatest common divisor of the considered polynomials in $\mathbb{F}[x]$ with $\mathbb{F}$ the quotient field of $\mathbb{D}$.

There exist several algorithms for solving this problem. The oldest one uses the Euclidean algorithm which solves the problem when $\mathbb{D}$ is a field but, when dealing with one of the easiest examples of integral domains $\mathbb{D} = \mathbb{Z}$, Euclidean Algorithm produces problems with the size of intermediary computations. Moreover, in this case, the computations providing the greatest common divisor in $\mathbb{D}[x]$ need to be made in $\mathbb{F}$ and usually it is harder to work in $\mathbb{F}$ than in $\mathbb{D}$. Possibly one of the best known algorithms for dealing with this problem is the one using subresultants. In this case the greatest common divisor is obtained performing only operations in $\mathbb{D}$ and, in the case of integer coefficients, the size of intermediary computations is well bounded (see [3], [4], [7], [10] or [11]).

Here we address the problem of computing the subresultant sequence of two polynomials $P(x)$ and $Q(x)$ in $\mathbb{D}[T_1, \ldots, T_r][x]$ with $\mathbb{D}$ an integral domain and $T_1, \ldots, T_r$ parameters taking values in the algebraic closure of $\mathbb{D}$, through the subresultant sequence. Our goal is to compare some algorithms which compute this sequence.

On the one hand, we consider algorithms which compute the subresultant sequence through determinants. In [9], the author introduces a new matrix whose determinant directly defines the subresultant polynomial of index j. Here we present analogous matrices derived from the Bezout and the Hybrid Bezout matrices whose determinants also define the subresultant polynomial of index j.

This work was completed with the support of MEC grant MTM2005-08690-C02-02.

On the other hand, we consider algorithms derived from the classical Subresultant Algorithm. For instance, the Improved Subresultant algorithm and the Flipflop algorithm, which involve exact divisions by minors extracted from Sylvester matrices. In this case, when the coefficients of the given polynomials depend on parameters, the computation of the subresultant sequence may be expensive because of the required divisions.

It's well known that when coefficients do not depend on parameters, the use of determinants is not advisable to obtain subresultants. However, we are to show that this is no more true when parameters appear.

2. Subresultant sequence

Let $\mathbb{D}$ be an integral domain and $P(x), Q(x) \in \mathbb{D}[x]$, with $n = \deg(P) \geq m = \deg(Q)$, $P = \sum_{k=0}^{n} p_k x^{n-k}$ and $Q = \sum_{k=0}^{m} q_k x^{m-k}$. The concept of polynomial determinant associated to a matrix with entries in $\mathbb{D}$ provides the classical way to define subresultant polynomials.

Definition 2.1. Let Δ be a $m \times n$ matrix with $m \leq n$. The determinant polynomial of Δ, $\mathbf{detpol}(\Delta)$, is defined as:

$$\mathbf{detpol}(\Delta) = \sum_{k=0}^{n-m} \det(\Delta_k) x^{n-m-k}$$

where Δ_k is the square submatrix of Δ consisting of the first $m-1$ columns and the $(k+m)^{-th}$ column.

Definition 2.2. If $i \in \{0, \ldots, \inf(n,m)-1\}$ then the Sylvester matrix of index i associated to P and Q is defined as:

$$\mathbf{Sylv}_i(P,Q) = \overbrace{\left(\begin{array}{cccccc} p_0 & \cdots & p_n & & & \\ & \ddots & & \ddots & & \\ & & & p_0 & \cdots & p_n \\ q_0 & \cdots & q_m & & & \\ & \ddots & & \ddots & & \\ & & & q_0 & \cdots & q_m \end{array}\right)}^{n+m-i} \begin{array}{l} \left.\vphantom{\begin{array}{c} p_0 \\ \ddots \\ p_0 \end{array}}\right\} m-i \\ \left.\vphantom{\begin{array}{c} q_0 \\ \ddots \\ q_0 \end{array}}\right\} n-i \end{array}$$

In these conditions the Subresultant polynomial of index i (or i–th subresultant polynomial) is defined as:

$$\mathbf{Sres}_i(P,Q) = \mathbf{detpol}(\mathbf{Sylv}_i(P,Q)).$$

The following theorem, see [1], introduces an algorithm for computing Subresultants (up to signs) that improves the well known Subresultant Algorithm in the defective case (see [10]). This improvement consists in avoiding the swell up of size of intermediate coefficients which occurs in the classical Subresultant Algorithm in the defective case.

Theorem 2.3 (Structure Theorem). *Let $0 \leq j < i \leq n$. Suppose that $\mathbf{Sres}_{i-1}(P,Q)$ is non-zero and of degree j. Let*

$$S_h = (-1)^{(n-h)(n-h-1)/2}\,\mathbf{Sres}_h(P,Q) \qquad R_h = (-1)^{(n-h)(n-h-1)/2}\,\mathbf{psc}_h(P,Q) \qquad (h \in \{0,\ldots,j\}).$$

1. *If S_{j-1} is zero, then $S_{i-1} = \gcd(P,Q)$ and $S_l = 0$, $l \leq j-1$.*
2. *If $S_{j-1} \neq 0$ has degree k then*

$$R_j \operatorname{lcoef}(S_{i-1})\, S_{k-1} = -\operatorname{Rem}(R_k \operatorname{lcoef}(S_{j-1})\, S_{i-1},\, S_{j-1}).$$

In fact, the quotient is in $\mathbb{D}[x]$.

Moreover if $j \leq m, k < j-1, S_k$ is proportional to S_{j-1}.

(a) $S_{j-2} = \cdots = S_{k+1} = 0$

(b) $R_k = (-1)^{(j-k)(j-k-1)/2}\,\dfrac{(\operatorname{lcoef}(S_{j-1}))^{j-k}}{R_j^{j-k-1}}$,

(c) $\operatorname{lcoef}(S_{j-1})\, S_k = R_k\, S_{j-1}$

In [11], *H. Lombardi et al* describe an algorithm which improves the subresultant algorithm in the non–defective case. Such algorithm is called Flipflop Algorithm and is based on the relations between the successive Sylvester matrix of $P(x)$ and $Q(x)$ on the one hand, and of $P(x)$ and $x\,Q(x)$ on the other hand. In the non–defective case, its main advantage is that it only requires to compute remainders of polynomials of same degree to obtain the Subresultant polynomials, in exchange for computing two principal subresultant sequences instead of one.

Another way to define the i–th subresultant polynomial is given by only one determinant (see [9]).

Theorem 2.4. *The i–th subresultant polynomial of P and Q with respect to the variable x is determined by the determinant of the following $(m+n-j)\times(m+n-j)$ matrix:*

$$\mathbf{Sres}_i(P,Q;y) = (-1)^{i(n-i+1)} \left| \begin{array}{ccccccccc} p_0 & p_1 & p_2 & \cdots & \cdots & p_n & & & \\ & \ddots & \ddots & \ddots & & & \ddots & & \\ & & p_0 & p_1 & p_2 & \cdots & \cdots & p_n & \\ & & & & & 1 & -x & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & 1 & -x \\ q_0 & q_1 & q_2 & \cdots & \cdots & \cdots & q_m & & \\ & \ddots & \ddots & \ddots & & & & \ddots & \\ & & q_0 & q_1 & q_2 & \cdots & \cdots & \cdots & q_m \end{array} \right| \begin{array}{l} \left.\vphantom{\begin{array}{c}p\\p\\p\end{array}}\right\} m-i \\ \left.\vphantom{\begin{array}{c}p\\p\\p\end{array}}\right\} i \\ \left.\vphantom{\begin{array}{c}p\\p\\p\end{array}}\right\} n-i \end{array} .$$

3. Bezout matrices and Subresultants

Next the definitions of the Bezout matrix and Hybrid Bezout matrix are introduced. The most general definition of the Bezout matrix of two polynomials is presented following [2].

Definition 3.1. The Bezout matrix associated to $P(x)$ and $Q(x)$ is the symmetric matrix:

$$\mathrm{Bez}(P,Q) = \begin{pmatrix} c_{0,0} & \cdots & c_{0,n-1} \\ \vdots & & \vdots \\ c_{n-1,0} & \cdots & c_{n-1,n-1} \end{pmatrix}$$

where the $c_{i,j}$ are defined by the formula:

$$\frac{P(x)Q(y) - P(y)Q(x)}{x-y} = \sum_{i,j=0}^{n-1} c_{i,j} x^i y^j.$$

Next we introduce the definition of the Hybrid Bezout matrix.

Definition 3.2. Given $P(x)$ and $Q(x)$, $\deg(P) = n \geq \deg(B) = m$, the Hybrid Bezout matrix associated to $P(x)$ and $Q(x)$, denoted by $\mathrm{Hbez}(P,Q)$, is a square matrix of size n whose entries are defined by:

- for $1 \leq i \leq m$, $1 \leq j \leq n$, the (i,j)–entry is the coefficient of x^{n-j} in the polynomial

$$K_{m-i+1} = (p_0 x^{m-i} + \ldots + p_{m-i})(q_{m-i+1} x^{n-m+i-1} + \ldots + q_m x^{n-m}) \\ -(p_{m-i+1} x^{n-m+i-1} + \ldots + p_n)(q_0 x^{m-i} + \ldots + q_{m-i});$$

- for $m+1 \leq i \leq n$, $1 \leq j \leq n$, the (i,j)–entry is the coefficient of x^{n-j} in the polynomial $x^{n-i}Q(x)$.

Both the Bezout and Hybrid Bezout matrix can also provide Subresultant polynomials by the following results.

Theorem 3.3. *Then Subresultant polynomials of $P(x)$ and $Q(x)$ can be obtained as follows (for $n-m \leq k \leq n$):*

- $(-1)^{k(k-1)/2} p_0^{n-m} \mathbf{Sres}_{n-k}(P,Q) = \mathbf{B}_{k,0} x^{n-k} + \mathbf{B}_{k,1} x^{n-k-1} + \ldots + \mathbf{B}_{k,n-k}$, *where $\mathbf{B}_{k,t}$ (for $0 \leq t \leq n-k$) denotes the $k \times k$ minor extracted from the last k columns, the last $k-1$ rows and the $n-k-t+1$–th row of* $\mathrm{Bez}(P,Q)$.
- $\mathbf{Sres}_{n-k}(P,Q) = \mathbf{Hb}_{k,0} x^{n-k} + \mathbf{Hb}_{k,1} x^{n-k-1} + \ldots + \mathbf{Hb}_{k,n-k}$, *where $\mathbf{Hb}_{k,t}$ denotes the $k \times k$ minor extracted from the first k columns, the last $k-1$ rows and the $n-k-t+1$–th row of* $\mathrm{Hbez}(P,Q)$.

For proof, see [6]. This theorem leads us to other definitions of subresultants. Let J_n denote the backward identity matrix of order n and so

$$J_n \mathrm{Bez}(P,Q) J_n = \begin{pmatrix} c_{n-1,n-1} & \cdots & c_{n-1,0} \\ \vdots & & \vdots \\ c_{0,n-1} & \cdots & c_{0,0} \end{pmatrix}.$$

Polynomials	SUB sequence Bez. matrix Theorem 3.4	SUB sequence Hybrid Bez. matrix Theorem 3.4	SUB sequence Theorem 2.4	SUB sequence Flipflop Alg.	SUB sequence Subres. Alg.
$(\mathbf{P}_1, 12, 9, 2, 6)$	2’, 15”	3’, 11”	1’, 31”	1’, 45”	3’, 5”
$(\mathbf{P}_2, 11, 7, 2, 6)$	13’,21”	9’ , 58”	6’,6”	7’, 46”	9’, 29”
$(\mathbf{P}_3, 7, 5, 3, 4)$	18’, 8”	14’, 49”	5’, 55”	42’, 34”	1h, 39’, 36’
$(\mathbf{P}_4, 7, 6, 3, 5)$	11’, 24”	10’, 55”	19’, 6”	1h, 12’, 46”	2h, 12’, 46”

Table 1. Experimental Analysis

Theorem 3.4. *The i–th subresultant polynomial of P and Q with respect to the variable x is given by the determinant of the $n \times n$ following matrices:*

$$(-1)^{(n-i)(n-i-1)/2} p_0^{n-m} \mathbf{Sres}_i(P,Q;y) = (-1)^i \left|\begin{array}{cccccc} c_{n-1,n-1} & c_{n-1,n-2} & \cdots & \cdots & \cdots & c_{n-1,0} \\ \vdots & \vdots & \ldots & \ldots & \ldots & \vdots \\ c_{j,n-1} & c_{j,n-2} & \cdots & \cdots & \cdots & c_{j,0} \\ & & 1 & -x & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -x \end{array}\right| \begin{array}{l} \left.\vphantom{\begin{array}{c} a \\ a \\ a \end{array}}\right\} n-i \\ \left.\vphantom{\begin{array}{c} a \\ a \\ a \end{array}}\right\} i \end{array},$$

$$\mathbf{Sres}_i(P,Q;y) = (-1)^{i+i(i-1)/2} \begin{array}{c} \overbrace{\hphantom{h_{0,0} \quad \cdots \quad h_{0,n-i-1}}}^{n-i} \quad \overbrace{\hphantom{-x \quad \cdots \quad -x}}^{i} \\ \left|\begin{array}{cccccc} h_{0,0} & \cdots & h_{0,n-i-1} & & & -x \\ \vdots & & \vdots & & \iddots & 1 \\ & & & -x & & \\ \vdots & & \vdots & 1 & & \\ & & & & & \\ \vdots & & \vdots & & & \\ h_{n-1,0} & \cdots & h_{n-1,n-i-1} & & & \end{array}\right| \end{array}$$

Remark that the order of these matrices is equal to n, while the order of matrices introduced in Theorem 2.4 varies in accordance with the index of the considered subresultant polynomial, from $m+n$ to n.

Experimental Results

Then the presented algorithms have been tested with pairs of polynomials in x, randomly generated, whose coefficients are in $\mathbb{Z}[L]$, being L the list of parameters. All the computations have been done with the Computer Algebra System `Maple 11`. These experimental results show us that, in order to compute the subresultant sequence, the computation of determinants is generally faster than the methods derived from the classical Subresultant Algorithm when the coefficients of the polynomials depend on parameters. Next we present some of these results in Table 1.

$\mathbf{P}_1$:

$$\begin{aligned}
p_1 = {} & x^{12} + (1056a^2b^2 - 780a^2b)x^5 - 912a^2b^3x^3 + (-408a^4b^2 - 504a)x + 372b^4 \\
q_1 = {} & x^9 - 594ax^6 - 288bx^5 + 351a^4x^3 + 612b^5x + 702ab^2 + 846b^5
\end{aligned}$$

$\mathbf{P}_2$:

$$\begin{aligned}
p_2 = {} & x^{11} - 220ax^7 + 506a^2x^6 + 660ab^2x^4 + 1045a^2b^4x^2 + 1045a^2b^3x - 198b^3 \\
q_2 = {} & x^7 + (-56ab^2 - 83a^3 - 91b^2 + 92b^4 - 93b^3 + 91a^3b)x^6 + \\
& (37b^3 - 46 - 68a - 42b - 47a^2 - 32b^2)x^5 + (-67 + 68b + 45ab^2 + 76 \\
& +a^3 + 6a^4 + 72a^2b^2)x^4 + (43ab^2 - 4a^3 - 50ab + 50a^4 + 67ab^3 - 39a^3b)x^3 \\
& +(58b - 94b^4 - 68a^4 + 14ab^3 - 35a^3b - 14a^2b^2)x^2 + (88 + a + 30a^3 \\
& +81b^4 - 5a^4 - 28a^3b)x - 88a - 43ab^2 - 73a^3 + 25ab + 4b^4 - 59a^4
\end{aligned}$$

$\mathbf{P}_3$:

$$\begin{aligned}
p_3 = {} & x^7 + (-84c + 8a^2b - 10ab^2 + 63abc^2 - 39ac^2 - 48ab)x^6 + (32a^2c^2 + 5b^2c^2 \\
& +8a^2bc + 63b^4 - 80a^2b^2)x^5 + (-65a + 52a^2c^2 + 99b^3c - 39ac + 49b^2c \\
& -76c^4)x^4 + (64abc - 34a + 60bc + 99c^3 + 48a^4 + 36c^4)x^3 + (63a^3c - 85b^2 \\
& -36bc^3 - 35ab^3 + 11b^3 + 90a^2b^2)x^2 + (-19 - 4b + 38a^2c^2 - 61a^3 - 58b^2 \\
& -91abc^2)x + 76a^3c - 68a^3 + 67b^2c - 38bc^2 + 23a^4 + 14a^2c \\
q_3 = {} & x^5 + (67 - 79c + 53ab^2 + 63a^3 - 24bc - 29c^2)x^4 + (74 - 60b + 19c - 68a^3 \\
& +78bc^2 + 34b^3)x^3 + (32abc + 65ab^2 - 26bc + 87ac^2 + 33a^2c + 76c^2)x^2 \\
& +(-81 - 47a^2b + 28bc + 73ab - 24a^2c + 8c^2)x + 45 - 12ab^2 + 28a^3 \\
& +3b^2 + 13bc^2 - 5b^3
\end{aligned}$$

$\mathbf{P}_4$:

$$\begin{aligned}
p_4 = {} & x^7 + (5abc^2 + 9a^3b - 6bc^2 - 12c^3a)x^5 - (7a^2b^2 + 7c^3)x^4 + 3cb^3x^2 - b^4c + 3cb^3, \\
q_4 = {} & x^6 + (a^2c^2 - 2a^4c - 47b^2c^3 - 3a^3bc + 7a^4 - 3a^5)x^4 + (-3b^4 - 82a^3b + 16bc^2 \\
& -40a^4c + 21a^2c - 94c^2)x^2 + 1
\end{aligned}$$

References

[1] S. Basu, R. Pollack, M.–F. Roy: *Algorithms in Real Algebraic Geometry*. Algorithms and Computations in Mathematics 10, Springer–Verlag (2003).

[2] D. Bini and V. Pan: *Polynomial and Matrix Computations.* Birkhäuser (1994).

[3] W. S. Brown and J. F. Traub: *On Euclid's algorithm and the theory of subresultants.* Journal of Association for Computing Machinery **118**, 505–514 (1971).

[4] G. E. Collins: *Subresultants and reduced polynomial remainder sequences.* Journal of Association for Computing Machinery **14**, 128–142 (1967).

[5] G. Diaz–Toca and L. Gonzalez–Vega: *Barnett's Theorem about the greatest common divisor of several univariate polynomials throug Bezout–like Matrices.* Journal of Symbolic Computation **34**, 59–81, (2002).

[6] G. Diaz–Toca and L. Gonzalez–Vega: *Various New Expressions for Subresultants and Their Appplications.* To appear in Applicable Algebra in Engineering, Communication and Computing (2004).

[7] L. Gonzalez–Vega, H. Lombardi, T. Recio and M.–F. Roy: *Specialisation de la suite de Sturm et sous-resultants (I).* Informatique Theorique et Applications **24**(6), 561–588 (1990).

[8] M. Kerber:*Division–free computation of subresultants using Bezout matrices.* Tech. Report MPI-I-2006-1-006 (2006).

[9] Y. B. Li: A new approach for constructing subresultants. Applied Mathematics and Computation **183** (2006) 471–476.

[10] R. Loos: *Generalized polynomial remainder sequences.* Computer Algebra, Computing Suplementum **4**, 115–138, Springer–Verlag (1982.)

[11] H. Lombardi, M.–F. Roy and M. Safey: *New structure theorem for subresultants.* Journal of Symbolic Computation **29**, 663–689 (2000).

Jounaidi Abdeljaoued
Université de Tunis
Tunisia
e-mail: jounaidi@esstt.rnu.tn

Gema M. Diaz–Toca
Universidad de Murcia
Spain
e-mail: gemadiaz@um.es

Laureano Gonzalez–Vega
Universidad de Cantabria
Spain
e-mail: laureano.gonzalez@unican.es