Optimizations of the Subresultant Algorithm

Lionel Ducos September 1998

Département de Mathématiques
Université de Poitiers
40, Avenue du Recteur Pineau
86022 Poitiers cedex
FRANCE.
E-mail : ducos@mathlabo.univ-poitiers.fr

Abstract

The subresultant algorithm is the most universal and used tool to compute the resultant or the greatest common divisor of two polynomials with coefficients in an integral ring (see [1], [3], [4]). Nevertheless, there exists several notable ameliorations of this algorithm (see [5], [10]).

I propose in this article two improvements in the parts of the subresultant algorithm where the calculations are most costly. The computing-time decreases in a spectacular way (see page 10).

Contents

1 Introduction

In theory, computing the resultant of two polynomials in an integral ring R with a chain of pseudo-divisions is quite possible. Unfortunately, in practice if the multiplication computing-time in R increases with the size of the elements, then obtaining a result becomes hopeless because the growth of pseudo-remainder coefficients is exponential.

The subresultant algorithm solves this problem because the size of the coefficients of the subresultant polynomials is small. In particular, it is in general smaller than the size of the resultant (see [7] or [11]). For the reader's convenience, I recall briefly this algorithm:

Convention if $p = \deg(P) \geq \deg(Q) = q$, then $S_q = \mathrm{lc}(Q)^{p-q-1}Q$ where lc is the leading coefficient. Of course, if $p = q$, the coefficients of S_q belong to $\mathrm{Frac}(R)$, but the leading coefficient $s_q = \mathrm{lc}(Q)^{p-q}$ always belongs to R.

Subresultant algorithm. (see [2], [3], [8] or [12])
Inputs : $P, Q \in R[X]$ $\quad \deg(P) \geq \deg(Q) \geq 1$
Output : List of non-zero subresultants of P and Q

$S \leftarrow$ empty list
$s \leftarrow \mathrm{lc}(Q)^{\deg(P)-\deg(Q)}$
$A \leftarrow Q \ ; \ B \leftarrow \mathrm{prem}(P, -Q)$
loop
  $d \leftarrow \deg(A) \ ; \ e \leftarrow \deg(B)$
  — here, $A \sim S_d$ if $d = \deg(Q)$ —
  — here, $A = S_d$ if $d < \deg(Q)$ —
  — here, $B = S_{d-1}, \ s = \mathrm{lc}(S_d)$ for $d \leq \deg(Q)$ —
  if $B = 0$ then return S
  $S \leftarrow [B] \cup S$
  — here, $S = [S_{d-1}, S_d, \ldots]$ —
  $\delta \leftarrow d - e$
  if $\delta > 1$ then $C \leftarrow \dfrac{\mathrm{lc}(B)^{\delta-1}B}{s^{\delta-1}} \ ; \ S \leftarrow [C] \cup S$
  else $C \leftarrow B$
  — here, $C = S_e, \ S = [S_e, \ldots]$ —
  if $e = 0$ then return S
  $B \leftarrow \dfrac{\mathrm{prem}(A, -B)}{s^{\delta}\,\mathrm{lc}(A)}$
  — here, $B = S_{e-1}$ —
  $A \leftarrow C$
  $s \leftarrow \mathrm{lc}(A)$
end loop

where prem denotes the pseudo-remainder, $\cup$ the concatenation of two lists and $\sim$ means proportional.

In this version of the algorithm, all non-zero subresultant polynomials of P and Q are computed. Observe that a loop mainly constitutes this program and two main calculations are carried out in this loop. They are derived from these following relations:

Theorem 1 *Let R be an integral ring, S_d be a regular (i.e. of degree d) subresultant polynomial of $P, Q \in R[X]$ with $d \leq \min(\deg(P), \deg(Q))$, and $S_{d-1} \neq 0$ of degree $e \in [0, d-1]$. Then*

$$1.\quad S_e = \frac{\mathrm{lc}(S_{d-1})^{d-e-1} S_{d-1}}{\mathrm{lc}(S_d)^{d-e-1}} \qquad 2.\quad S_{e-1} = \frac{\mathrm{prem}(S_d, -S_{d-1})}{\mathrm{lc}(S_d)^{d-e+1}}$$

2 Lazard's optimization

The subresultant algorithm seems to be ideal to make small coefficient calculations. But let us look into the first equality of theorem 1. Can the computation S_e be optimized? Daniel Lazard has proved in [9] that it is possible to avoid the exponentiations $\mathrm{lc}(S_{d-1})^{d-e-1}$ and $\mathrm{lc}(S_d)^{d-e-1}$ and their division, which can be expensive. The following calculation can be made: $s_d = \mathrm{lc}(S_d)$,

$$S_e = \frac{\dfrac{\dfrac{\dfrac{\mathrm{lc}(S_{d-1})^2}{s_d} \times \mathrm{lc}(S_{d-1})}{s_d} \times \cdots \times \mathrm{lc}(S_{d-1})}{s_d} \times S_{d-1}}{s_d}$$

where every division is exact (see also [5]):

$$\text{for all } \delta \in [0, d-e[, \qquad \text{we have } \frac{\mathrm{lc}(S_{d-1})^{\delta+1}}{\mathrm{lc}(S_d)^{\delta}} \in R$$

Furthermore a dichotomous method may improve this calculation and then lowers its total cost:

$$\frac{\left(\dfrac{\mathrm{lc}(S_{d-1})^2}{s_d}\right)^2}{s_d} \cdots$$

Optimized calculation of S_e. "dichotomous Lazard"
Inputs : S_d, S_{d-1}
Output : S_e

$n \leftarrow \deg(S_d) - \deg(S_{d-1}) - 1$ — here, $n = n_0 = d - e - 1$
if $n = 0$ then return S_{d-1}
$(x, y) \leftarrow (\operatorname{lc}(S_{d-1}), \operatorname{lc}(S_d))$
$a \leftarrow 2^{\lfloor \log_2(n) \rfloor}$ — here, $a \leq n < 2a$
$c \leftarrow x$
$n \leftarrow n - a$
loop — here, $c = x^j / y^{j-1}$, $aj \leq n_0 < a(j+1)$, $a = 2^?$ —
  exit when $a = 1$
  $a \leftarrow \dfrac{a}{2}$; $c \leftarrow \dfrac{c^2}{y}$
  if $n \geq a$ then $c \leftarrow \dfrac{cx}{y}$; $n \leftarrow n - a$
end loop
return $\dfrac{cS_{d-1}}{y}$

3 A second optimization

In the same way, let us look into the second equality of theorem 1:

$$S_{e-1} = \frac{\operatorname{prem}(S_d, -S_{d-1})}{\operatorname{lc}(S_d)^{d-e+1}}$$

The calculations of the pseudo-remainder, the exponentiation and the quotient can be extremely expensive. Our aim is to compute S_{e-1} while limiting the size of the intermediate coefficients as we did for S_e.

In [5], I prove with an explicit algorithm that the problem is solvable: S_{e-1} can be obtained from intermediate coefficients of size roughly twice the size of S_{e-1}-coefficients.

Recently, T. Lickteig and M.-F. Roy proved in [10] the following relation of euclidean divisibility:

$$s_e c_{d-1} S_d = A S_{d-1} + (-1)^{d-e+1} s_d^2 S_{e-1} \qquad A \in R[X]$$

where $s_d = \operatorname{lc}(S_d)$, $s_e = \operatorname{lc}(S_e)$, $c_{d-1} = \operatorname{lc}(S_{d-1})$.

Unfortunately, the size of the intermediate coefficients is three times as big as the size of the S_{e-1}-coefficients, and this last formula does not bring any improvement if the degree of S_{d-1} is $d-1$ (*i.e.* $S_e = S_{d-1}$).

Now, I propose several new relations of euclidean divisibility between subresultant polynomials and any other polynomials:

Theorem 2 *Let R be an integral ring, S_d be a regular (i.e. of degree d) subresultant polynomial of $P, Q \in R[X]$, $S_{d-1} \neq 0$ of degree $e \in [0, d-1]$, s_d, c_{d-1} and s_e be respectively the leading coefficients of S_d, S_{d-1} and S_e. Then*

1. for all $G \in R[X]$ such that $\deg(G) < d$

$$s_d s_e G = AS_{d-1} + s_d B \qquad A, B \in R[X], \quad \deg(B) < e$$

2. in particular, if $G = S_d - s_d X^d$, we have a better relation

$$s_d s_e (S_d - s_d X^d) = AS_{d-1} + s_d^2 D \qquad A, D \in R[X], \quad \deg(D) < e$$

3. for all $G \in R[X]$ such that $\deg(G) \leq d$

$$s_e c_{d-1} G = AS_{d-1} + B \qquad A, B \in R[X], \quad \deg(B) < e$$

and c_{d-1} divides B if $\deg(G) < d$.

4. in particular, if $G = S_d$, we have a better relation

$$s_e c_{d-1} S_d = AS_{d-1} + (-1)^{d-e+1} s_d^2 S_{e-1} \qquad A \in R[X]$$

(T. Lickteig and M.-F. Roy's formula, see [10])

The proof of these relations can be found at the end of this paper (section 6) or in [6].

Now, let us take an interest in a new algorithm. Suppose we know S_d (of degree d) and $S_{d-1} \neq 0$ (of degree e). We can compute S_e with Lazard's method. How can S_{e-1} be calculated?

It follows from point *4.* of theorem 2 that

$$s_e c_{d-1} S_d \equiv (-1)^{d-e+1} s_d^2 S_{e-1} \mod S_{d-1}$$

Now $S_d = s_d X^d + (S_d - s_d X^d)$, therefore

$$s_d s_e c_{d-1} X^d + c_{d-1} s_e (S_d - s_d X^d) \equiv (-1)^{d-e+1} s_d^2 S_{e-1} \mod S_{d-1}$$

The remainder $\operatorname{rem}(s_e c_{d-1} X^d, S_{d-1})$ can be obtained by point *3.* of theorem 2:

$$H_d = \operatorname{rem}(s_e c_{d-1} X^d, S_{d-1}) \equiv s_e c_{d-1} X^d \mod S_{d-1} \qquad H_d \in R[X]$$

Moreover every remainder $\operatorname{rem}(s_e X^j, S_{d-1})$ (with $j < d$) can be obtained by point *1.* of theorem 2:

$$H_j = \frac{\operatorname{rem}(s_d s_e X^j, S_{d-1})}{s_d} \equiv s_e X^j \mod S_{d-1} \qquad H_j \in R[X]$$

To compute $(H_j)_{j\leq d}$, I propose the following method:

$$\begin{array}{rll} H_j = & s_e X^j & \text{for } j < e \\ H_j = & s_e X^e - S_e & \text{for } j = e \\ H_j = & \operatorname{rem}(XH_{j-1}, S_{d-1}) & \text{for } j \in \ ]e, d[\\ = & XH_{j-1} - \dfrac{\pi_e(XH_{j-1})S_{d-1}}{c_{d-1}} & \\ H_j = & \operatorname{rem}(c_{d-1}XH_{j-1}, S_{d-1}) & \text{for } j = d \\ = & c_{d-1}XH_{j-1} - \pi_e(XH_{j-1})S_{d-1} & \end{array}$$

where $\pi_e(XH_{j-1})$ denotes the coefficient of X^e in XH_{j-1}. The size of the intermediate coefficients of these formulas is roughly twice the size of S_{e-1}-coefficients (see the three remarks in the proof of theorem 2).

Then, by point *2.* of the same theorem, we have

$$s_d^2 D = \operatorname{rem}(s_d s_e(S_d - s_d X^d), S_{d-1}) = \sum_{j<d} s_d \pi_j(S_d) H_j \qquad\qquad D \in R[X]$$

where $\pi_j(S_d)$ denotes the coefficient of X^j in S_d. Note that

$$D = \frac{\displaystyle\sum_{j<d} \pi_j(S_d) H_j}{s_d} \qquad \text{and} \qquad s_d D \equiv s_e(S_d - s_d X^d) \mod S_{d-1}$$

Finally, $(-1)^{d-e+1} s_d^2 S_{e-1} \equiv s_d H_d + c_{d-1} s_d D \mod S_{d-1}$
Since the degrees of S_{e-1}, H_d and D are lower than $\deg(S_{d-1})$, it is an equality:

$$S_{e-1} = (-1)^{d-e+1} \frac{H_d + c_{d-1} D}{s_d} = (-1)^{d-e+1} \frac{c_{d-1}(XH_{d-1} + D) - \pi_e(XH_{d-1}) S_{d-1}}{s_d}$$

Optimized calculation of S_{e-1}.
Inputs : $A \sim S_d,\ S_{d-1},\ S_e,\ s_d$
Output : S_{e-1}

$(d, e) \leftarrow (\deg(A), \deg(S_{d-1}))$
$(c_{d-1}, s_e) \leftarrow (\mathrm{lc}(S_{d-1}), \mathrm{lc}(S_e))$
for j in $0 \ldots e-1$ loop
    $H_j \leftarrow s_e X^j$
end loop
$H_e \leftarrow s_e X^e - S_e$
for j in $e+1 \ldots d-1$ loop
    $H_j \leftarrow XH_{j-1} - \dfrac{\pi_e(XH_{j-1})S_{d-1}}{c_{d-1}}$
end loop
$D \leftarrow \dfrac{\sum_{j<d} \pi_j(A)H_j}{\mathrm{lc}(A)}$ — here, $D = \frac{\sum_{j<d} \pi_j(S_d)H_j}{\mathrm{lc}(S_d)}$ —
return $(-1)^{d-e+1} \dfrac{c_{d-1}(XH_{d-1} + D) - \pi_e(XH_{d-1})S_{d-1}}{s_d}$

Optimized subresultant algorithm.
Inputs : $P, Q \in R[X] \qquad \deg(P) \geq \deg(Q) \geq 1$
Output : List of non-zero subresultants of P and Q

$S \leftarrow$ empty list
$s \leftarrow \mathrm{lc}(Q)^{\deg(P)-\deg(Q)}$
$A \leftarrow Q \ ; \ B \leftarrow \mathrm{prem}(P, -Q)$
loop
  $d \leftarrow \deg(A) \ ; \ e \leftarrow \deg(B)$
  — here, $A \sim S_d \quad$ if $d = \deg(Q)$ —
  — here, $A = S_d \quad$ if $d < \deg(Q)$ —
  — here, $B = S_{d-1}, \ s = \mathrm{lc}(S_d) \quad$ for $d \leq \deg(Q)$ —
  if $B = 0$ then return S
  $S \leftarrow [B] \cup S$
  — here, $S = [S_{d-1}, S_d, \ldots]$ —
  $\delta \leftarrow d - e$
  if $\delta > 1$ then $C \leftarrow$ optimized calculation of $S_e \ ; \ S \leftarrow [C] \cup S$
  else $C \leftarrow B$
  — here, $C = S_e, \ S = [S_e, \ldots]$ —
  if $e = 0$ then return S
  $B \leftarrow$ optimized calculation of S_{e-1}
  $A \leftarrow C$
  $s \leftarrow \mathrm{lc}(A)$
end loop

4 Computing-time analysis

The complexity of this algorithm is calculated in the most unfavorable case, *i.e.* when $\deg(S_i(P, Q)) = i$ for all $i \in [0, n]$ with $P, Q \in \mathbf{Z}[X]$ of degree n.

Obtaining S_{d-1} from S_{d+1} and S_d requires about $4d$ multiplications and $2d$ divisions (the cost of an addition is negligible). The total numbers of multiplications and divisions of this algorithm are respectively equivalent to $2n^2$ and n^2.

Let $M(t, t)$ be the cost of a multiplication in $\mathbf{Z}$ of two t-sized elements, and $D(2t, t)$ be the cost of a division in $\mathbf{Z}$ of a $2t$-sized element by a t-sized one: thus $M(t, t), D(2t, t) \in \mathcal{O}(t^2)$. If c is the largest coefficient of P and Q, then Hadamard's inequality applied to Sylvester's matrix shows that the largest coefficient that appears in their subresultant polynomials is smaller than $(2nc^2)^n$ (see [1], page 253). Let τ be the size of $(2nc^2)^n$, *i.e.* $\tau \in \mathcal{O}(n \log(nc))$. So, the

total complexity of the optimization is bounded by

$$2n^2M(\tau,\tau) + n^2D(2\tau,\tau)$$

Remark. The complexity of the procedure "dichotomous Lazard" is bounded by $(2\log_2(d-e)+e)M(\tau,\tau) + (2\log_2(d-e)+e)D(2\tau,\tau)$, or more simply by $nM(\tau,\tau) + nD(2\tau,\tau)$.

In the same way, the total complexity of the subresultant algorithm is bounded by

$$n^2M(\tau,\tau) + n^2M(2\tau,\tau) + \frac{n^2}{2}D(3\tau,2\tau)$$

5 Examples

test 1 $P = aX^6 + bX^5 + cX^4 + dX^3 + eX^2 + fX + g$
$Q = P'$

test 2 $P = X^5 + aX^4 + bX^3 + cX^2 + dX + e$
$Q = X^5 + fX^4 + gX^3 + hX^2 + iX + j$

test 3 $P = X^7 + aX^3 + bX^2 + cX + d$
$Q = X^7 + eX^3 + fX^2 + gX + h$

test 4 $P = X^{20} + aX^{15} + b$
$Q = X^{20} + cX^5 + d$

test 5 $P = (X+a)^{15}$
$Q = (X+z)^{15}$

test 6 $P = X^{30} + aX^{20} + 2aX^{10} + 3a$
$Q = X^{25} + 4bX^{15} + 5bX^5$

test 7 $P = (a+X)^{90}$
$Q = (a-X)^{60}$

test 8 $P = \sum_{j=0}^{75} a^{75-j}X^j$
$Q = \sum_{j=0}^{75} ja^jX^j$

test 9 $P = \sum_{j=0}^{200} X^j$
$Q = 1 + \sum_{j=0}^{100} jX^j$

test 10 $P = 1 + \sum_{j=1}^{900} jX^j$
$Q = 1 + \sum_{j=1}^{900} j^2X^j$

test 11 $P, Q \in \mathbf{Z}[X]$ two random polynomials of degree 140

test 1	0,1,2,3,4	test 6	0,0,5,5,10,10,15,15,20,20
test 2	0,1,2,3,4	test 7	0,1,2,3,...,58,59
test 3	0,1,2,3	test 8	0,1,1,73,74
test 4	0,0,5,5,10,10,15,15	test 9	0,1,2,3,3,97,98,99
test 5	0,1,2,3,...,13,14	test 10	0,1,2,2,898,899
		test 11	0,1,2,...,138,139

Degrees of the non-zero subresultant polynomials

	subresultant algorithm	optimized algorithm		subresultant algorithm	optimized algorithm
test 1	71	7.8	test 6	935	27
test 2	2364	80	test 7	58	51
test 3	1162	77	test 8	2342	7.6
test 4	1091	59	test 9	39	1.3
test 5	499	245	test 10	264	14
			test 11	199	166

Computing-time in seconds

6 Proof of theorem 2

Recalling

Henceforth, R is an integral ring with unity.

Definition 1 (see [5], pages 320-323) *Let M and N be two R-modules. Let $g : M^n \to R$ and $f : M^m \to N$ be two R-multilinear alternating applications. The* ***exterior product*** *$g \wedge f$ is given by the formula :*

$$M^{n+m} \to N \ : \quad (v_1, \ldots, v_{n+m}) \mapsto \sum_{\sigma} \operatorname{sgn}(\sigma) g(v_{\sigma_1}, \ldots, v_{\sigma_n}) f(v_{\sigma_{n+1}}, \ldots, v_{\sigma_{n+m}})$$

Definition 2 *Let $g : M^{n-1} \to R$ be a (n-1)-****multilinear form*** *of M. Then $g^\natural$ denotes the n-****multilinear application***

$$\begin{array}{rccl} g^\natural = g \wedge \mathrm{Id}_M & : & M^n & \longrightarrow & M \\ & & v & \longmapsto & \displaystyle\sum_{i=1}^{n} (-1)^{n-i} g(v_1, \ldots, \not{v_i}, \ldots, v_n) v_i \end{array}$$

Definition 3 *Let $g : M^n \to N$ be a R-multilinear alternating application. We shall call* $\ker g$ *the R-submodule $\{z \in M \mid g(z, \ldots) = 0\}$.*

Theorem 3 *Let M et N two R-modules, $f : M^{n+1} \to N$ a (n+1)-multilinear application and $g : M^{m-1} \to R$ a (m-1)-multilinear form. Consider $x \in M^m$ and $z \in M^n$. If* $\mathrm{Vect}(z) \subset \ker g \subset M$, *then*

$$f(g^\natural(x), z) = (g \wedge f)(x, z).$$

Proof

$$\begin{aligned}(g \wedge f)(x,z) &= \sum_{i=1}^{m}(-1)^{m-i} g(x_1,\ldots,\not{x}_i,\ldots,x_m) f(x_i,z) \qquad z \subset \ker g,\\ &= f\left(\sum_{i=1}^{m}(-1)^{m-i}\ g(x_1,\ldots,\not{x}_i,\ldots,x_m).x_i, z\right) \quad f \text{ is linear},\\ &= f(g^\natural(x), z).\end{aligned}$$

Theorem 4 *Let M et N two R-modules, $g : M^{n+1} \to N$ a (n+1)-multilinear application, $f : M^{k-1} \to R$ and $h : M^m \to R$ two multilinear forms. Consider $x \in M^k$, $z \in M^n$, $z' \in M^m$ such that* $\mathrm{Vect}(z') \subset \mathrm{Vect}(z) \subset \ker f$, *then*

$$g\Big(f \wedge h^\natural\ (x,\ z'),\ z\Big) = \pm h(z')\ (f \wedge g)(x,\ z).$$

Proof

$$\begin{aligned} g\Big(f \wedge h^\natural\ (x,\ z'),\ z\Big) &= \pm g\Big(h(z')\ f^\natural(x),\ z\Big) \quad \text{because } z' \subset \mathrm{Vect}(z) \subset \ker f\\ &= \pm h(z')\ g\Big(f^\natural(x),\ z\Big)\\ &= \pm h(z')\ (f \wedge g)(x,z) \quad \text{theorem 3 with } g \text{ and } f\end{aligned}$$

Notations (see [5], pages 329-330) : If $P \in R[X]$, the expression $X^{[j,i]}P$ $(j \geq i)$, where $j \geq i$, denotes the list

$$X^jP,\ X^{j-1}P, \ldots, X^{i+1}P,\ X^iP$$

and the empty list if $j < i$. Furthermore, $\pi_k(P)$ will point out the coefficient of degree k of P. We note $\begin{bmatrix} j \\ i \end{bmatrix}$ the list $\{j, j-1, \ldots, i+1, i\}$ if $j \geq i$, or the empty list if $j < i$. If K is the list $\{a, b, c, \ldots, z\}$, we define these applications :

$$\begin{aligned}\det\nolimits_K &= \pi_a \wedge \pi_b \wedge \pi_c \wedge \cdots \wedge \pi_z\\ \det\nolimits_K^\natural &= \pi_a \wedge \pi_b \wedge \pi_c \wedge \cdots \wedge \pi_z \wedge \mathrm{Id}\end{aligned}$$

and for instance, if $j \geq i$:

$$\det\nolimits_{\begin{bmatrix} j \\ i \end{bmatrix}}(X^{[j,i]}P) = (\pi_j \wedge \pi_{j-1} \wedge \cdots \wedge \pi_i)(X^jP,\ X^{j-1}P, \ldots,\ X^iP)$$

Definition 4 *In [11], by definition, the subresultant S_d of two polynomials $P, Q \in R[X]$ (respectively of degree p and q) is the determinant polynomial of the matrix given by the polynomials $X^{[q-d-1,0]}P$ and $X^{[p-d-1,0]}Q$ (with $d < \min(p,q)$). So we have*

$$S_d = \det^{\natural}_{\left[\substack{p+q-d-1 \\ d+1}\right]}(X^{[q-d-1,0]}P, X^{[p-d-1,0]}Q)$$

$$\pi_d(S_d) = \det_{\left[\substack{p+q-d-1 \\ d}\right]}(X^{[q-d-1,0]}P, X^{[p-d-1,0]}Q)$$

Property 1 *Let $k \in \mathbf{N}$. For $d \leq \min(p,q)$, we have*

$$X^k S_{d-1} = \det^{\natural}_{\left[\substack{p+q-d+k \\ d+k}\right]}(X^{[q-d+k,k]}P, X^{[p-d+k,k]}Q)$$

Some technical lemmas

Henceforth, we suppose that $q = \deg(Q)$ is lower (or equal) than $p = \deg(P)$. Then, we can define $S_q = \mathrm{lc}(Q)^{p-q-1}Q$ (with coefficients in $\mathrm{Frac}(R)$) and $s_q = \mathrm{lc}(S_q) = \mathrm{lc}(Q)^{p-q} \in R$.

Remark that $s_q = 1$ if $p = q$.

Lemma 1 *Let $d \leq q$ $(\leq p)$ and $i \leq j < \alpha$ be in $\mathbf{N}$ such that*

$$\deg(S_{d-1}) + j < p + q - d + i = \alpha$$

Let $g : R[X]^n \to R[X]$ be a R-multilinear alternating application. Let G be a finite list of $R[X]$ such that $\deg(z) < \alpha$ for any polynomial $z \in G$. Then

$$g(G,\ X^{[\alpha-p-1,i]}P,\ X^{[\alpha-q-1,i]}Q,\ X^{[j,i]}S_{d-1}) = \pm s_d^{j-i+1}\ (\det_{\left[\substack{\alpha+j-i \\ \alpha}\right]} \wedge g)(G,\ X^{[q-d+j,i]}P,\ X^{[p-d+j,i]}Q)$$

or straightforwardly

$$g(G,\ X^{[\alpha-p-1,i]}P,\ X^{[\alpha-q-1,i]}Q,\ X^{[j,i]}S_{d-1}) \quad \in \quad s_d^{j-i+1}R[X]$$

Proof (It is obvious if $d = p = q$ because $s_d = 1$.)
Step i : Let $x = \{X^{q-d+i}P,\ X^{p-d+i}Q\}$, $z' = X^{[q-d+i-1,i]}P \cup X^{[p-d+i-1,i]}Q$, $z = z' \cup G \cup X^{[j,i+1]}S_{d-1}$, $f = \pi_{p+q-d+i}$, $h = \det_{\left[\substack{p+q-d+i-1 \\ d+i}\right]}$, then theorem 4 directly gives

$$g(G, X^{[q-d+i-1,i]}P, X^{[p-d+i-1,i]}Q, X^{[j,i]}S_{d-1}) = \pm s_d\ (\pi_{p+q-d+i} \wedge g)(G,\ X^{[q-d+i,i]}P,\ X^{[p-d+i,i]}Q,\ X^{[j,i+1]}S_{d-1})$$

because $X^i S_{d-1} = f \wedge h^{\natural}(x, z')$ and $s_d = \pi_{d+i}(X^i S_d) = h(z')$. Repeating the steps $i+1, \ldots, j$, we finally obtain

$$g(G, X^{[q-d+i-1,i]}P, X^{[p-d+i-1,i]}Q, X^{[j,i]}S_{d-1}) = \pm s_d^{j-i+1}\ (\det_{\left[\substack{p+q-d+j \\ p+q-d+i}\right]} \wedge g)(G,\ X^{[q-d+j,i]}P,\ X^{[p-d+j,i]}Q)$$

Lemma 2 *Let $d \leq q$ $(\leq p)$ and j be in* $\mathbf{N}$ *such that* $\deg(S_{d-1}) + j \leq d$. *Let $f : R[X]^n \to R[X]$ be a R-multilinear alternating application. Let G be a finite list of $R[X]$ such that* $\deg(z) \leq d$ *for any polynomial $z \in G$. Then*

$$f(G, S_d, X^{[j,0]}S_{d-1}) = \pm s_d^{j+1}\ (\det_{\left[\substack{p+q-d+j \\ d+1}\right]} \wedge f)(G,\ X^{[q-d+j,0]}P,\ X^{[p-d+j,0]}Q)$$

or straightforwardly

$$f(G, S_d, X^{[j,0]}S_{d-1}) \quad \in \quad s_d^{j+1}R[X]$$

Proof

$f(G,\ S_d,\ X^{[j,0]}S_{d-1})$

$= \pm(\det_{\left[\substack{p+q-d-1 \\ d+1}\right]} \wedge f)(G,\ X^{[q-d-1,0]}P,\ X^{[p-d-1,0]}Q,\ X^{[j,0]}S_{d-1})$

(definition 4 and theorem 3)

$= \pm s_d^{j+1}\ (\det_{\left[\substack{p+q-d+j \\ d+1}\right]} \wedge f)(G,\ X^{[q-d+j,0]}P,\ X^{[p-d+j,0]}Q)$

(lemma 1 applied with $g = \det_{\left[\substack{p+q-d-1 \\ d+1}\right]} \wedge f$)

Lemma 3 *Let $d \leq q$ $(\leq p)$ and $i \leq j$ be in* $\mathbf{N}$ *such that* $\deg(S_{d-1}) + j < d + i$. *Let $f : R[X]^n \to R[X]$ be a R-multilinear alternating application. Let G' be a finite list of $R[X]$ such that* $\deg(z) < d + i$ *for any polynomial $z \in G'$. Then*

$$f(G', X^{[j,i]}S_{d-1}) = \pm s_d^{j-i}\ (\det_{\left[\substack{p+q-d+j \\ d+i}\right]} \wedge f)(G',\ X^{[q-d+j,i]}P,\ X^{[p-d+j,i]}Q)$$

or straightforwardly

$$f(G', X^{[j,i]}S_{d-1}) \quad \in \quad s_d^{j-i}R[X]$$

Proof

$f(G',\ X^{[j,i]}S_{d-1})$

$= \pm(\det_{\left[\substack{p+q-d+i \\ d+i}\right]} \wedge f)(G',\ X^{[q-d+i,i]}P,\ X^{[p-d+i,i]}Q,\ X^{[j,i+1]}S_{d-1})$

(property 1 and theorem 3)

$= \pm s_d^{j-i}\ (\det_{\left[\substack{p+q-d+j \\ d+i}\right]} \wedge f)(G',\ X^{[q-d+j,i]}P,\ X^{[p-d+j,i]}Q)$

(lemma 1 applied with $g = \det_{\left[\substack{p+q-d+i \\ d+i}\right]} \wedge f$ and $G = G' \cup [X^iP, X^iQ]$)

Proof of theorem 2

1. Let $G \in R[X]$ be a polynomial such that $\deg(G) < d$. We consider the following Euclidean division:

$$c_{d-1}^{d-e}G = US_{d-1} + V \qquad\qquad U, V \in R[X],\ \deg(V) < e$$

where $e = \deg(S_{d-1})$ and c_{d-1} the leading coefficient of S_{d-1}. We are going to prove that U and V respectively belong to $s_d^{d-e-2}R[X]$ and $s_d^{d-e-1}R[X]$. Developing the exterior product $(\det_{\binom{d-1}{e}} \wedge \mathrm{Id})(G, X^{[d-e-1,0]}S_{d-1})$, we find again the expression of the previous division with

$$U = \sum_{k=0}^{d-e-1} \pm \det_{\binom{d-1}{e}}(G, X^{[d-e-1,k+1]}S_{d-1}, X^{[k-1,0]}S_{d-1})\ X^k$$

$$V = \pm \det^{\natural}_{\binom{d-1}{e}}(G, X^{[d-e-1,0]}S_{d-1})$$

On one hand, lemma 3 proves that $V = s_d^{d-e-1}B$ where

$$B = \pm \det^{\natural}_{\binom{p+q-e-1}{e}}(G,\ X^{[q-e-1,0]}P,\ X^{[p-e-1,0]}Q)$$

with $f = \det^{\natural}_{\binom{d-1}{e}}$, $j = d-e-1$ and $i = 0$.
On the other hand, for $k \in \{0, \ldots, d-e-1\}$, the coefficient z_k of X^k in U is

$$\begin{aligned} z_k &= \pm c_{d-1}^k \det_{\binom{d-1}{e+k}}(G, X^{[d-e-1,k+1]}S_{d-1}) \\ &= \pm \det_{\binom{d+k-1}{e+k}}(G, X^{[d+k-e-1,k+1]}S_{d-1}) \end{aligned}$$

Lemma 3 proves that $U \in s_d^{d-e-2}R[X]$ with $f = \det_{\binom{d+k-1}{e+k}}$, $j = d+k-e-1$ and $i = k+1$.

So, we can write $U = s_d^{d-e-2}A$ and $V = s_d^{d-e-1}B$ where $A, B \in R[X]$. Finally, the first Euclidean division becomes

$$s_d s_e G = \frac{c_{d-1}^{d-e}}{s_d^{d-e-2}} G = AS_{d-1} + s_d B \qquad\qquad A, B \in R[X],\ \deg(B) < e$$

> Remark. The degree of $B = \frac{\mathrm{rem}(s_d s_e G, S_{d-1})}{s_d}$ is lower than $e-1$, of course. But, if $e \le j < d$ and $G = X^j$, then any coefficient of the polynomial B is a minor of the Sylvester's matrix of P and Q : for all $i < e$, we have
>
> $$\begin{aligned} \pi_i(B) = &\ \pm(\det_{\binom{p+q-e-1}{e}} \wedge \pi_i)(X^j,\ X^{[q-e-1,0]}P,\ X^{[p-e-1,0]}Q) \\ = &\ \pm(\det_{\binom{p+q-e-1}{j+1}} \wedge \det_{\binom{j-1}{e}} \wedge \pi_i)(X^{[q-e-1,0]}P,\ X^{[p-e-1,0]}Q) \end{aligned}$$

2. Now, if $G = S_d - s_d X^d = \pi_d^{\natural}(X^d, S_d)$, then the rest of the division $c_{d-1}^{d-e}G = US_{d-1} + V$ belongs to $s_d^{d-e}R[X]$. To prove this, we write:
$V = \pm \det^{\natural}_{\binom{d-1}{e}} \left(\pi_d^{\natural}(X^d, S_d), X^{[d-e-1,0]}S_{d-1}\right)$

$$= \pm \det^{\natural}_{\left\lfloor \substack{d \\ e} \right\rfloor}(X^d, S_d, X^{[d-e-1,0]} S_{d-1}) \qquad \text{(theorem 3)}$$

and lemma 2 shows that $V = s_d^{d-e} D$ where

$$D = \pm \det^{\natural}_{\left\lfloor \substack{p+q-e-1 \\ e} \right\rfloor}(X^d,\ X^{[q-e-1,0]} P,\ X^{[p-e-1,0]} Q)$$

with $f = \det^{\natural}_{\left\lfloor \substack{d \\ e} \right\rfloor}$, $G = X^d$, $j = d-e-1$, and $i = 0$. Then, the first Euclidean division becomes

$$s_d s_e (S_d - s_d X^d) = \frac{c_{d-1}^{d-e}}{s_d^{d-e-2}} G = A S_{d-1} + s_d^2 D \qquad A, D \in R[X],\ \deg(D) < e$$

> Remark. The degree of $D = \frac{\mathrm{rem}(s_d s_e (S_d - s_d X^d), S_{d-1})}{s_d^2}$ is lower than $e-1$ and any coefficient of this polynomial is a minor of the Sylvester's matrix of P and Q : forall $i < e$, we have
>
> $$\begin{aligned} \pi_i(D) = &\ \pm(\det_{\left\lfloor \substack{p+q-e-1 \\ e} \right\rfloor} \wedge \pi_i)(X^d,\ X^{[q-e-1,0]} P,\ X^{[p-e-1,0]} Q) \\ = &\ \pm(\det_{\left\lfloor \substack{p+q-e-1 \\ d+1} \right\rfloor} \wedge \det_{\left\lfloor \substack{d-1 \\ e} \right\rfloor} \wedge \pi_i)(X^{[q-e-1,0]} P,\ X^{[p-e-1,0]} Q) \end{aligned}$$

3. Let $G \in R[X]$ be a polynomial such that $\deg(G) \leq d$. We consider the following Euclidean division:

$$c_{d-1}^{d-e+1} G = U S_{d-1} + V \qquad U, V \in R[X],\ \deg(V) < e$$

We are going to prove that U and V belong in $s_d^{d-e-1} R[X]$. Developing the exterior product $(\det_{\left\lfloor \substack{d \\ e} \right\rfloor} \wedge \mathrm{Id})(G, X^{[d-e,0]} S_{d-1})$, we find again the expression of the previous division where

$$U = \sum_{k=0}^{d-e} \pm \det_{\left\lfloor \substack{d-1 \\ e} \right\rfloor}(G, X^{[d-e,k+1]} S_{d-1}, X^{[k-1,0]} S_{d-1})\ X^k$$

$$V = \pm \det^{\natural}_{\left\lfloor \substack{d \\ e} \right\rfloor}(G, X^{[d-e,0]} S_{d-1})$$

Lemma 3 immediately proves that $V = s_d^{d-e-1} B$ where

$$B = \pm \det^{\natural}_{\left\lfloor \substack{p+q-e \\ e} \right\rfloor}(G, S_{d-1},\ X^{[q-e,1]} P,\ X^{[p-e,1]} Q)$$

with $f = \det^{\natural}_{\left\lfloor \substack{d \\ e} \right\rfloor}$, $j = d-e$, $i = 1$, and $G' = \{G, S_{d-1}\}$.
Furthermore, for $k \in \{0, \ldots, d-e\}$, the coefficient z_k of X^k in U is

$$\begin{aligned} z_k &= \pm c_{d-1}^k \det_{\left[\begin{smallmatrix} d \\ e+k \end{smallmatrix}\right]}(G, X^{[d-e,k+1]} S_{d-1}) \\ &= \pm \det_{\left[\begin{smallmatrix} d+k \\ e+k \end{smallmatrix}\right]}(G, X^{[d+k-e,k+1]} S_{d-1}) \end{aligned}$$

Lemma 3 proves that $U \in s_d^{d-e-1} R[X]$ where $f = \det_{\left[\begin{smallmatrix} d+k \\ e+k \end{smallmatrix}\right]}$, $j = d+k-e$ and $i = k+1$.

So, we can write $U = s_d^{d-e-1} A$ and $V = s_d^{d-e-1} B$ with $A, B \in R[X]$. Finally, the first Euclidean division becomes

$$c_{d-1} s_e G = \frac{c_{d-1}^{d-e+1}}{s_d^{d-e-1}} G = A S_{d-1} + B \qquad\qquad A, B \in R[X],\ \deg(B) < e$$

> Remark. The degree of $B = \mathrm{rem}(c_{d-1} s_e G, S_{d-1})$ is lower than $e-1$. If $G = X^d$, then any coefficient of the polynomial B is a sum of two products of two Sylvester's minors : forall $i < e$, we have

$$\begin{aligned} \pi_i(B) = \ & \pm(\det_{\left[\begin{smallmatrix} p+q-e \\ e \end{smallmatrix}\right]} \wedge \pi_i)(X^d, S_{d-1},\ X^{[q-e,1]} P,\ X^{[p-e,1]} Q) \\ = \ & \pm(\det_{\left[\begin{smallmatrix} p+q-e \\ d+1 \end{smallmatrix}\right]} \wedge \det_{\left[\begin{smallmatrix} d-1 \\ e \end{smallmatrix}\right]} \wedge \pi_i)(S_{d-1},\ X^{[q-e,1]} P,\ X^{[p-e,1]} Q) \\ = \ & \pm\ \pi_i(S_{d-1}) \cdot (\det_{\left[\begin{smallmatrix} p+q-e \\ d+1 \end{smallmatrix}\right]} \wedge \det_{\left[\begin{smallmatrix} d-1 \\ e \end{smallmatrix}\right]})(X^{[q-e,1]} P,\ X^{[p-e,1]} Q) \\ & \pm\ c_{d-1} \cdot (\det_{\left[\begin{smallmatrix} p+q-e \\ d+1 \end{smallmatrix}\right]} \wedge \det_{\left[\begin{smallmatrix} d-1 \\ e+1 \end{smallmatrix}\right]} \wedge \pi_i)(X^{[q-e,1]} P,\ X^{[p-e,1]} Q) \end{aligned}$$

4. With $G = S_d$, consider the last relation, $c_{d-1} s_e S_d = A S_{d-1} + B$, and the classic one (theorem 1), $c_{d-1}^{d-e+1} S_d = U S_{d-1} + (-1)^{d-e+1} s_d^{d-e+1} S_{e-1}$. Then $B = (-1)^{d-e+1} \dfrac{s_d^{d-e+1}}{s_d^{d-e-1}} S_{e-1}$ and we obtain

$$c_{d-1} s_e S_d = A S_{d-1} + (-1)^{d-e+1} s_d^2 S_{e-1} \qquad\qquad A \in R[X]$$

References

[1] A.G. Akritas. *Elements of computer algebra with applications.* John Wiley and Sons, 1989.

[2] W.S. Brown and J.F. Traub. On Euclid's Algorithm and Theory of Subresultants. *Ass. Comp. Mach.*, 18(4):505–514, Octobre 1971.

[3] H. Cohen. *A course in computational algebraic number theory,* ch.3. Springer-Verlag, 1993.

[4] S.R. Czapor, K.O. Geddes, and G. Labahn. *Algorithms for computer algebra*. Kluwer Academic Publishers, 1992.

[5] L. Ducos. Algorithme de Bareiss, Algorithme des sous-résultants. *Theoretical Informatics And Applications*, 30(4):319–347, 1996.

[6] L. Ducos. *Effectivité en théorie de Galois. Sous-résultants.* Université de Poitiers, 1997. Thèse doctorale.

[7] L. González-Vega, H. Lombardi, T. Recio, and M.-F. Roy. Spécialisation de la suite de Sturm et sous-résultants (I). *Informatique théorique et Applications*, 24(6):561–588, Décembre 1990.

[8] D.E. Knuth. *The Art of Computer Programming*. Addison-Wesley Publishing Company, 1959. second edition.

[9] D. Lazard. Sous-résultants. Manuscrit non publié.

[10] T. Lickteig and M.-F. Roy. Cauchy index computation. Manuscrit non publié (à paraître), Novembre 1996.

[11] R. Loos. Generalized Polynomial Remainder Sequences. *Symbolic and algebraic computation*, Computing, Supplementum(4):115–137, 1982. Springer-Verlag.

[12] C. Quitté. Calcul du pgcd et du résultant dans les anneaux de polynômes. Cours de D.E.A.(1994), mars 1992.