Extended Euclidean Algorithm and Determinants

佐々木建昭
（理化学研究所）

Being given polynomials $P_1=F$ and $P_2=G$, with $\deg(F)\geq\deg(G)$, we can generate a polynomial remainder sequence $(P_3, P_4, \ldots, P_k)$, where deg denotes the degree. The extended Euclidean algorithm calculates polynomials A_i and B_i, $i=3,\ldots,k$, such that $A_iF + B_iG = P_i$, $\deg(A_i)<\deg(G)-\deg(P_i)$, $\deg(B_i)<\deg(F)-\deg(P_i)$. According to the subresultant theory, each P_i, $i\geq3$, can be represented by a subresultant of F and G. This paper extends the subresultant theory on polynomial remainder sequence to include polynomials A_i and B_i and proves the applicability of the reduced PRS and the subresultant PRS algorithms to the calculation of A_i and B_i, $i=3,\ldots,k$. Furthermore, an order reduction method for determinants representing P_i, A_i and B_i is given.

§1. Introduction

Let F(x) and G(x) be polynomials with coefficients in an Euclidean domain I, such as the ring of integers or multivariate polynomials:

$$F(x) = f_\ell x^\ell + f_{\ell-1}x^{\ell-1} + \ldots + f_0, \quad f_\ell \neq 0, \quad f_i \in I,$$
$$G(x) = g_m x^m + g_{m-1}x^{m-1} + \ldots + g_0, \quad g_m \neq 0, \quad g_i \in I. \tag{1.1}$$

Assuming $\ell\geq m$, we can generate a polynomial remainder sequence (which is denoted as PRS in short), $P_1(x)$, $P_2(x)$, ..., $P_k(x)\neq0$, $P_{k+1}=0$, through the following formula:

$$P_1(x) = F(x),$$
$$P_2(x) = G(x), \tag{1.2}$$
$$\beta_i P_{i+1}(x) = \alpha_i P_{i-1}(x) - Q_i(x)P_i(x), \quad \deg(P_{i+1})<\deg(P_i),$$
$$i=2,3,\ldots,k-1,$$

where α_i, $\beta_i \in I$. The degree of P_{k+1} is defined to be zero: $\deg(P_{k+1})=0$. For each i, $3\leq i\leq k$, there exist unique polynomials A_i and B_i such that

$$A_iP_1 + B_iP_2 = P_i,$$
$$\deg(A_i) < \deg(P_2)-\deg(P_i), \tag{1.3}$$
$$\deg(B_i) < \deg(P_1)-\deg(P_i).$$

We call the sets of polynomials $(A_3,A_4,\ldots,A_k)$ and $(B_3,B_4,\ldots,B_k)$ associate polynomial sequences, or APS's in short. Note that, if $\deg(P_k)>0$, there exist A_{k+1} and B_{k+1} satisfying (1.3) with i=k+1. The A_{k+1} and B_{k+1} are, however, not unique. One solution to (1.3) with i=k+1 is $(A_{k+1}=P_2/D, B_{k+1}=P_1/D)$ with $D=\mathrm{GCD}(P_1,P_2)$.

The PRS plays an important role in algebra, for example, in calculating

polynomial greatest common divisors or as a Strum sequence. From the mathematical point of view, calculation of PRS is trivial. However, from the computational point of view, it is never trivial: simple algorithms for PRS calculation require large amounts of computation due to a phenomenon of coefficient growth[1,5]. Due to these reasons, PRS has drawn a strong attention of computer algebrists. The greatest advance on efficient PRS calculation was made by Collins[2] in 1966, who discovered the so-called reduced PRS algorithm. Collins' work was deepened by himself[3], by Brown and Traub[4], and by Brown[6], and elegant subresultant PRS algorithm was found. These algorithms are based on the subresultant theory on PRS, which is briefly surveyed in §2.

On the other hand, little attention was paid on the calculation of APS's which also are quite important in algebra. In §3, we show that the subresultant theory on PRS is easily extended to include APS's. That is, we give a determinant representation for each A_i and B_i, $3 \le i \le k$, and prove that the reduced PRS and the subresultant PRS algorithms are applicable to the APS calculation. In §4, we give another determinant representation for each A_i and B_i. In §5, we present a simple method for reducing the order of determinants representing A_i and B_i as well as P_i, $3 \le i \le k$. The reduced determinants are similar to Bezout's determinants, and they are quite convenient to calculate A_k or B_k separately without calculating APS's. Therefore, Hermite's algorithm for the partial fraction decomposition or the inversion of polynomials over an algebraic function field, for example, can be reduced to the determinant calculation.

§2. Notations and survey

This section defines notations, surveys the extended Euclidean algorithm and the subresultant theory briefly, and presents formulas which are necessary in the following sections.

We assume that $F, G \in I[x]$, with I an Euclidean domain, as in §1. The leading coefficient of polynomial $P(x)$ is denoted as $lc(P)$. A polynomial P is primitive if GCD of all its coefficients is 1. Polynomials P and $\tilde{P}$ are similar to each other over I if there exist nonzero a and $\tilde{a} \in I$ such that $aP = \tilde{a}\tilde{P}$. The similarity is represented as $P \sim \tilde{P}$. We use the following abbreviations for each polynomial P_i in PRS:

$$\begin{aligned} c_i &= lc(P_i), \\ n_i &= \deg(P_i), \\ d_i &= n_i - n_{i+1}. \end{aligned} \tag{2.1}$$

In most textbooks on mathematics, PRS is generated by using the division operation. Then, the PRS is not guaranteed to be in $I[x]$. In computer algebra, it is rather common to use the pseudo-division defined below for the PRS generation:

$$lc(P_i)^{\deg(P_{i-1})-\deg(P_i)+1} P_{i-1} = Q_i P_i + \beta_i P_{i+1}, \quad \deg(P_{i+1}) < \deg(P_i). \tag{2.2}$$

That is, we choose α_i in (1.2) as

$$\alpha_i = c_i^{(d_{i-1}+1)}. \tag{2.3}$$

The choice of β_i changes from algorithm to algorithm.

The following algorithm, where β_i is not specified explicitly, calculates

A_i and B_i as well as P_i, i=3,4,...,k:

EXTENDED EUCLIDEAN ALGORITHM

INPUT : polynomials P_1 and P_2 s.t. $\deg(P_1) \geq \deg(P_2)$;

OUTPUT: PRS $(P_3,...,P_k)$, and APS's $(A_3,...,A_k)$, $(B_3,...,B_k)$;

$A_1 \leftarrow 1$; $A_2 \leftarrow 0$; $B_1 \leftarrow 0$; $B_2 \leftarrow 1$;

$i \leftarrow 2$;

while $\deg(P_i)>0$ do begin

$Q_i \leftarrow$ quotient$(\alpha_i P_{i-1}, P_i)$;

$\beta_i P_{i+1} \leftarrow \alpha_i P_{i-1} - Q_i P_i$;

$\beta_i A_{i+1} \leftarrow \alpha_i A_{i-1} - Q_i A_i$;

$\beta_i B_{i+1} \leftarrow \alpha_i B_{i-1} - Q_i B_i$;

$i \leftarrow i+1$; end

if $P_i=0$ then $i \leftarrow i-1$; (comment, here i=k)

return $((P_3,A_3,B_3), (P_4,A_4,B_4), ..., (P_i,A_i,B_i))$.

For example[1], let us apply the above algorithm with $\beta_i=1$, i=2,3,...,k, to the following polynomials over Z:

$$P_1(x) = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5,$$
$$P_2(x) = 3x^6 + 5x^4 - 4x^2 - 9x + 21. \qquad (2.4)$$

Then, we obtain the following sequences (for simplicity, only the coefficient vectors are written):

P_3 : (-15, 0, 3, 0, -9),
A_3 : (27),
B_3 : (-9, 0, 6),

P_4 : (15795, 30375, -59535),
A_4 : (-18225, 0, -34020),
B_4 : (6076, 0, 7290, 0, -10935),

P_5 : (1254542875143750, -1654608338437500),
A_5 : 134521003125×(-507, 975, -4631, 1820, -6087),
B_5 : 134521003125×(169, -325, 1431, -390, 906, 585, -2035),

P_6 : (125933387955007431009311419921875000),
A_6 : 966087637932149615733398437500×(13989, 18450, 40562, ...),
B_6 : 966087637932149615733398437500×(-4663, -6150, -10412, ...).

This shows the seriousness of coefficient growth for not only P_i but also A_i and B_i. We can suppress the coefficient growth by making PRS primitive:

P_3 : (5, 0, -1, 0, 3),
A_3 : (-9),
B_3 : (3, 0, -2),

P_4 : (13, 25, -49),
A_4 : (-15, 0, -28),
B_4 : (5, 0, 6, 0, -9),

P_5 : (4663, -6150),
A_5 : (1/2)×(-507, 975, -4631, 1820, -6087),
B_5 : (1/2)×(169, -325, 1431, -390, 906, 585, -2035),

P_6 : (1),
A_6 : (1/130354)×(13989, 18450, 40562, 67125, 5149, -9737),
B_6 : (1/130354)×(-4663, -6150, -10412, -18275, 9888, 21579, ...).

As we see, nontrivial choice of β_i does not guarantee the APS's to be in Z[x] although the PRS is so. Hence, the question arises: how can we calculate APS's efficiently over I?

According to (1.2), we have

$$\beta_i(P_i,P_{i+1}) = (\beta_i P_i,\ \alpha_i P_{i-1} - Q_i P_i) = (P_{i-1},P_i)\begin{pmatrix} 0 & \alpha_i \\ \beta_i & -Q_i \end{pmatrix}.$$

Continuing this reduction, we obtain

$$(P_i,P_{i+1}) = (P_1,P_2)\prod_{r=2}^{i}\left\{\begin{pmatrix} 0 & \alpha_r \\ \beta_r & -Q_r \end{pmatrix}/\beta_r\right\}. \tag{2.5}$$

(Note the order of matrices in the product: $\Pi_{r=2}^{i}M_i$ is $M_2M_3...M_i$ and not $M_i...M_3M_2$.) Comparison of (1.3) and (2.5) gives

$$\begin{pmatrix} A_i & A_{i+1} \\ B_i & B_{i+1} \end{pmatrix} = \prod_{r=2}^{i}\left\{\begin{pmatrix} 0 & \alpha_r \\ \beta_r & -Q_r \end{pmatrix}/\beta_r\right\}. \tag{2.6}$$

The j-th degree subresultant S_j of F and G is defined by the following determinant of order $\ell+m-2j$:

$$S_j(F,G) = \begin{vmatrix} f_\ell & f_{\ell-1} & \cdots & f_{2j+2-m} & x^{m-j-1}F \\ & f_\ell & f_{\ell-1} & \cdots & f_{2j+3-m} & x^{m-j-2}F \\ & & \cdots & & \cdots & \cdots \\ & & & f_\ell & \cdots & f_{j+1} & x^{0}F \\ g_m & g_{m-1} & \cdots & g_{2j+2-\ell} & x^{\ell-j-1}G \\ & g_m & g_{m-1} & \cdots & g_{2j+3-\ell} & x^{\ell-j-2}G \\ & & \cdots & & \cdots & \cdots \\ & & & g_m & \cdots & g_{j+1} & x^{0}G \end{vmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}m-j\ \text{rows} \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}\ell-j\ \text{rows} \end{matrix}, \tag{2.7}$$

where $f_i=g_i=0$ for $i<0$. Note that S_j can be defined only for such j that $0\leq j\leq m-1$. The S_j is a polynomial of degree j or less. The most important results of the subresultant theory are the following equalities:

$$P_i = \rho_i S_{n_{i-1}-1}(F,G) = \lambda_i S_{n_i}(F,G), \quad i=3,4,\ldots,k, \tag{2.8}$$

$$\rho_i = c_{i-1}^{(d_{i-1}-1)}\prod_{r=2}^{i-1}\{(\alpha_r/\beta_r)^{(n_r-n_{i-1}+1)}c_r^{-(d_{r-1}+d_r)} \times(-1)^{(n_{r-1}-n_{i-1}+1)(n_r-n_{i-1}+1)}\}, \tag{2.9}$$

$$\lambda_i = c_i^{(1-d_{i-1})}\prod_{r=2}^{i-1}\{(\alpha_r/\beta_r)^{(n_r-n_i)}c_r^{-(d_{r-1}+d_r)} \times(-1)^{(n_{r-1}-n_i)(n_r-n_i)}\}, \tag{2.10}$$

$$S_j(F,G) = 0 \quad \text{for } 0\leq j<n_k \text{ or } n_i<j<n_{i-1}-1,\ i=3,\ldots,k. \tag{2.11}$$

Note that we can define ρ_i for i=k+1 if $n_k>0$, while λ_{k+1} cannot be defined because $c_{k+1}=0$.

As an example of choosing β_i, let us set $\beta_i=1$, $i\geq 2$, with α_i given by (2.3). Then, we see from (2.9) or (2.10) that the expression sizes of ρ_i and λ_i grow exponentially as i increases. This coefficient growth can be suppressed drastically by choosing β_i suitably. In the rest of this section, we describe two famous choices of β_i, the reduced PRS algorithm and the subresultant PRS algorithm.

The reduced PRS algorithm determines β_i as

$$\beta_2 = 1, \qquad \beta_i = \alpha_{i-1} = c_{i-1}^{(d_{i-2}+1)}, \quad i=3,4,\ldots,k. \tag{2.12}$$

Then, according to (2.8) and (2.9), we find that $P_i = \tilde{\rho}_i S_{n_{i-1}-1}(P_1,P_2)$ with $\tilde{\rho}_i \in I$, $i=3,4,\ldots,k$.

The subresultant PRS algorithm calculates PRS so that $\rho_i = 1$ for every $i\geq 3$, that is, $P_i = S_{n_{i-1}-1}(P_1,P_2) \in I[x]$. According to ref.4, the algorithm determines β_i as

$$\beta_2 = (-1)^{(d_1+1)},$$
$$\beta_i = -c_{i-1}\zeta_i^{d_{i-1}}, \quad i=3,4,\ldots,k, \tag{2.13}$$

where ζ_i is calculated successively as

$$\zeta_2 = -1,$$
$$\zeta_3 = -c_2^{d_1},$$
$$\zeta_i = (-c_{i-1})^{d_{i-2}}\zeta_{i-1}^{(1-d_{i-2})}, \quad i=4,\ldots,k. \tag{2.14}$$

Brown[6] proved that ζ_i, $i\geqq 4$, is in I.

§3. Determinant representations for APS's

We introduce the following determinants $T_j^{(A)}$ and $T_j^{(B)}$:

$$T_j^{(A)}(F,G) = \begin{vmatrix} f_\ell & f_{\ell-1} & \cdots & f_{2j+2-m} & x^{m-j-1} \\ & f_\ell & f_{\ell-1} & \cdots & f_{2j+3-m} & x^{m-j-2} \\ & & \cdots & & \cdots & \\ & & & f_\ell & \cdots & f_{j+1} & x^0 \\ g_m & g_{m-1} & \cdots & g_{2j+2-\ell} & 0 \\ & g_m & g_{m-1} & \cdots & g_{2j+3-\ell} & 0 \\ & & \cdots & & \cdots & \\ & & & g_m & \cdots & g_{j+1} & 0 \end{vmatrix} \begin{matrix} \\ \\ \\ \\ \} m-j \text{ rows} \\ \\ \\ \\ \} \ell-j \text{ rows} \\ \\ \\ \end{matrix}, \tag{3.1}$$

$$T_j^{(B)}(F,G) = \begin{vmatrix} f_\ell & f_{\ell-1} & \cdots & f_{2j+2-m} & 0 \\ & f_\ell & f_{\ell-1} & \cdots & f_{2j+3-m} & 0 \\ & & \cdots & & \cdots & \\ & & & f_\ell & \cdots & f_{j+1} & 0 \\ g_m & g_{m-1} & \cdots & g_{2j+2-\ell} & x^{\ell-j-1} \\ & g_m & g_{m-1} & \cdots & g_{2j+3-\ell} & x^{\ell-j-2} \\ & & \cdots & & \cdots & \\ & & & g_m & \cdots & g_{j+1} & x^0 \end{vmatrix} \begin{matrix} \\ \\ \\ \} m-j \text{ rows} \\ \\ \\ \\ \} \ell-j \text{ rows} \\ \\ \\ \end{matrix}, \tag{3.2}$$

where $f_i=g_i=0$ for $i<0$. We define $T_j^{(A)}$ and $T_j^{(B)}$ only for such j that $0\leqq j\leqq m-1$. The degree of $T_j^{(A)}$ is m−j−1 or less and that of $T_j^{(B)}$ is ℓ−j−1 or less, respectively. Comparing (2.7) with (3.1) and (3.2), we directly obtain

$$T_j^{(A)}(F,G)\cdot F + T_j^{(B)}(F,G)\cdot G = S_j(F,G). \tag{3.3}$$

This equality implies that

$$A_i \sim T_{n_i}^{(A)}(F,G) \sim T_{n_{i-1}-1}^{(A)}(F,G), \tag{3.4}$$

$$B_i \sim T_{n_i}^{(B)}(F,G) \sim T_{n_{i-1}-1}^{(B)}(F,G), \tag{3.4'}$$

suggesting applicability of the reduced PRS and subresultant PRS algorithms to the APS calculation. The main purpose of this paper is to investigate

the determinants $T_j^{(A)}$ and $T_j^{(B)}$ in details and to prove the applicability. The discussion goes parallelly with that in ref.4.

Lemma 1. Let F and G be polynomials defined by (1.1) with $\deg(F) = \ell \geqq m = \deg(G)$, and let

$$F = QG + H, \quad \deg(H) = n < m, \tag{3.5}$$

that is,

$$\begin{aligned} H &= h_n x^n + h_{n-1} x^{n-1} + \cdots + h_0, \\ Q &= q_{\ell-m} x^{\ell-m} + \cdots + q_0. \end{aligned} \tag{3.6}$$

Then, we have the following equalities:

when $0 \leqq j < n$

$$T_j^{(A)}(F,G) = (-1)^{(\ell-j)(m-j)} g_m^{(\ell-n)} T_j^{(B)}(G,H), \tag{3.7}$$

$$T_j^{(B)}(F,G) = (-1)^{(\ell-j)(m-j)} g_m^{(\ell-n)} \{T_j^{(A)}(G,H) - Q T_j^{(B)}(G,H)\}, \tag{3.7'}$$

when $j=n$

$$T_n^{(A)}(F,G) = (-1)^{(\ell-n)(m-n)} g_m^{(\ell-n)} h_n^{(m-n-1)}, \tag{3.8}$$

$$T_n^{(B)}(F,G) = (-1)^{(\ell-n)(m-n)} g_m^{(\ell-n)} h_n^{(m-n-1)} (-Q), \tag{3.8'}$$

when $n < j < m-1$

$$T_j^{(A)}(F,G) = T_j^{(B)}(F,G) = 0, \tag{3.9}$$

and when $j=m-1$

$$T_{m-1}^{(A)}(F,G) = (-1)^{(\ell-m+1)} g_m^{(\ell-m+1)}, \tag{3.10}$$

$$T_{m-1}^{(B)}(F,G) = (-1)^{(\ell-m+1)} g_m^{(\ell-m+1)} (-Q). \tag{3.10'}$$

(Proof) Representing (3.5) in the form of coupled equations in the coefficients of F, G, Q, and H, we have

$$\begin{aligned} &(1 \ -q_{\ell-m} \ \cdots \ -q_0) \begin{bmatrix} f_\ell & f_{\ell-1} & \cdots & \cdots & f_0 \\ g_m & g_{m-1} & \cdots & g_0 & \\ & \ddots & & & \ddots \\ & & g_m & g_{m-1} & \cdots \ g_0 \end{bmatrix} \Big\} \ell-m+2 \text{ rows} \\ &= (\overbrace{0 \ \cdots \ 0}^{\ell-n} \ h_n \ \cdots \ h_0). \end{aligned}$$

Taking out left $\ell+m-2j-i$ columns of the vectors of these equations, with $1 \leqq i \leqq m-j$, and adding x^{m-j-i} from the right side, we obtain

$$\begin{aligned} &(\overbrace{0 \ \cdots \ 0}^{i-1} \ f_\ell \ f_{\ell-1} \ \cdots \ f_{2j+i+1-m} \ x^{m-j-i}) \\ &= \sum_{r=0}^{\ell-m} q_r (\overbrace{0 \ \cdots \ 0}^{i+r-1} \ g_m \ g_{m-1} \ \cdots \ g_{2j+i+1-\ell+r} \ 0) \\ &\quad + (0 \ \cdots \ 0 \ h_n \ h_{n-1} \ \cdots \ h_{2j+i+1-m} \ x^{m-j-i}). \end{aligned} \tag{3.11}$$

Similarly, since

$$x^{m-j-i} Q = \sum_{r=0}^{\ell-m} q_{\ell-m-r} x^{\ell-j-i-r},$$

eq.(3.5) yields

$$(\overbrace{0 \ \cdots \ 0}^{i-1} \ f_\ell \ f_{\ell-1} \ \cdots \ f_{2j+i+1-m} \ 0) \tag{3.12}$$

$$= \sum_{r=0}^{\ell-m} q_{\ell-m-r}\,(\overbrace{0\ \cdots\ 0}^{i+r-1}\ g_m\ g_{m-1}\ \cdots\ g_{2j+i+1-\ell+r}\ x^{\ell-j-i-r})$$

$$+\ (0\ \cdots\cdots\ 0\ h_n\ h_{n-1}\ \cdots\ h_{2j+i+1-m}\ -x^{m-j-i}Q).$$

Replacing the upper m-j rows of the matrices in (3.1) and (3.2) by (3.11) and (3.12), respectively, only the vectors constructed from coefficients of H (i.e., the last vectors in (3.11) and (3.12)) contribute to the determinants, because every vector constructed from coefficients of G in (3.11) or (3.12) leads to a determinant with identical rows. Hence the resulting determinants can be reduced easily. Classifying the resulting determinants by different values of j, $0\leq j\leq m-1$, we obtain (3.7) ~ (3.10').//

Lemma 2. Let ($P_1=F$, $P_2=G$, P_3, ..., P_k) be a PRS generated through (1.2). Then, we have the following equalities:

when $0\leq j<n_{i+1}$, $2\leq i\leq k-1$,

$$\alpha_i^{(n_i-j)}\begin{pmatrix} T_j^{(A)}(P_{i-1},P_i) \\ T_j^{(B)}(P_{i-1},P_i)\end{pmatrix} = (-1)^{(n_{i-1}-j)(n_i-j)} \times c_i^{(d_{i-1}+d_i)}\beta_i^{(n_i-j)}(1/\beta_i)\begin{pmatrix} 0 & \alpha_i \\ \beta_i & -Q_i\end{pmatrix}\begin{pmatrix} T_j^{(A)}(P_i,P_{i+1}) \\ T_j^{(B)}(P_i,P_{i+1})\end{pmatrix}, \tag{3.13}$$

when $0\leq j=n_{i+1}$, $2\leq i\leq k-1$,

$$\alpha_i^{d_i}\begin{pmatrix} T_{n_{i+1}}^{(A)}(P_{i-1},P_i) \\ T_{n_{i+1}}^{(B)}(P_{i-1},P_i)\end{pmatrix} = (-1)^{(d_{i-1}+d_i)d_i} \times c_i^{(d_{i-1}+d_i)}c_{i+1}^{(d_i-1)}\beta_i^{d_i}(1/\beta_i)\begin{pmatrix} 0 & \alpha_i \\ \beta_i & -Q_i\end{pmatrix}\begin{pmatrix} 0 \\ 1\end{pmatrix}, \tag{3.14}$$

when $n_{i+1}<j<n_i-1$, $2\leq i\leq k-1$, or when $0\leq j<n_k-1$

$$\begin{pmatrix} T_j^{(A)}(P_{i-1},P_i) \\ T_j^{(B)}(P_{i-1},P_i)\end{pmatrix} = 0, \tag{3.15}$$

and when $0\leq j=n_i-1$, $2\leq i\leq k$,

$$\alpha_i\begin{pmatrix} T_{n_i-1}^{(A)}(P_{i-1},P_i) \\ T_{n_i-1}^{(B)}(P_{i-1},P_i)\end{pmatrix} = (-1)^{(d_{i-1}+1)} \times c_i^{(d_{i-1}+1)}\beta_i(1/\beta_i)\begin{pmatrix} 0 & \alpha_i \\ \beta_i & -Q_i\end{pmatrix}\begin{pmatrix} 0 \\ 1\end{pmatrix}. \tag{3.16}$$

(Proof) Definitions (3.1) and (3.2) of $T_j^{(A)}$ and $T_j^{(B)}$ yield

$$T_j^{(A)}(aF,bG) = a^{(m-j-1)}b^{(\ell-j)}T_j^{(A)}(F,G), \tag{3.17}$$

$$T_j^{(B)}(aF,bG) = a^{(m-j)}b^{(\ell-j-1)}T_j^{(B)}(F,G). \tag{3.17'}$$

Putting $F=\alpha_iP_{i-1}$, $G=P_i$, $Q=Q_i$, $H=\beta_iP_{i+1}$ in (3.7) ~ (3.10'), and using (3.17) and (3.17'), we obtain (3.13) ~ (3.16).//

Theorem 1. The $T_j^{(A)}$ and $T_j^{(B)}$ satisfy the following equalities:

$$(A_i,B_i) = \rho_i(T_{n_{i-1}-1}^{(A)}(P_1,P_2),T_{n_{i-1}-1}^{(B)}(P_1,P_2)), \quad i=3,4,\ldots,k, \tag{3.18}$$

$$(A_i,B_i) = \lambda_i(T_{n_i}^{(A)}(P_1,P_2),T_{n_i}^{(B)}(P_1,P_2)), \quad i=3,4,\ldots,k, \tag{3.19}$$

$$(A_{k+1},B_{k+1}) \sim \rho_{k+1}(T_{n_k-1}^{(A)}(P_1,P_2),T_{n_k-1}^{(B)}(P_1,P_2)), \quad \text{if } n_k>0, \tag{3.18'}$$

$$T_j^{(A)}(P_1,P_2) = T_j^{(B)}(P_1,P_2) = 0 \quad \text{for other values of } j. \tag{3.20}$$

(Proof) If $j<n_i$, then $j<n_i<n_{i-1}<\ldots<n_1$ and we can perform the reduction (3.13) successively obtaining

$$\left(\prod_{r=2}^{i-1}\alpha_r^{(n_r-j)}\right)\begin{pmatrix}T_j^{(A)}(P_1,P_2)\\ T_j^{(B)}(P_1,P_2)\end{pmatrix}=\begin{pmatrix}T_j^{(A)}(P_{i-1},P_i)\\ T_j^{(B)}(P_{i-1},P_i)\end{pmatrix} \tag{3.21}$$

$$\times\prod_{r=2}^{i-1}\left\{(-1)^{(n_{r-1}-j)(n_r-j)}c_r^{(d_{r-1}+d_r)}\beta_r^{(n_r-j)}\begin{pmatrix}0 & \alpha_r\\ \beta_r & -Q_r\end{pmatrix}\Big/\beta_r\right\}.$$

When $0\leq j=n_i-1$, $i=2,3,\ldots,k-1$, (3.21) and (3.16) yield

$$\rho_{i+1}\begin{pmatrix}T_{n_i-1}^{(A)}(P_1,P_2)\\ T_{n_i-1}^{(B)}(P_1,P_2)\end{pmatrix}=\prod_{r=2}^{i}\left\{\begin{pmatrix}0 & \alpha_r\\ \beta_r & -Q_r\end{pmatrix}\Big/\beta_r\right\}\begin{pmatrix}0\\ 1\end{pmatrix}. \tag{3.22}$$

According to (2.6), the r.h.s. of this equality is $\begin{pmatrix}A_{i+1}\\ B_{i+1}\end{pmatrix}$, hence we obtain (3.18). Similarly, when $0\leq j=n_{i+1}$, $i=2,3,\ldots,k-1$, (3.21), (3.14), and (2.6) yield (3.19). If $n_k>0$, (3.16) with $i=k$ tells us that $T_{n_k-1}^{(A)}$ and $T_{n_k-1}^{(B)}$ are nonzero. Since $S_{n_k-1}(P_1,P_2)=0$, we see from (3.3) that

$$T_{n_k-1}^{(A)}(P_1,P_2)\cdot P_1+T_{n_k-1}^{(B)}(P_1,P_2)\cdot P_2=0.$$

Hence, (3.18') is obtained. Eq.(3.20) is obtained immediately from (3.15) and (3.21).//

Theorem 1 implies that the reduced PRS and the subresultant PRS algorithms are applicable to the APS calculation. That is, we can choose β_i, $i=2,3,\ldots,k$, in the extended Euclidean algorithm as (2.12) or (2.13), leaving the APS's $(A_3, A_4, \ldots, A_k)$ and $(B_3, B_4, \ldots, B_k)$ in $I[x]$.

As an example, we calculate APS's for polynomials (2.4) by applying the subresultant PRS algorithm:

P_3 : (15, 0, -3, 0, 9),
A_3 : (-27),
B_3 : (9, 0, -6),

P_4 : (65, 125, -245),
A_4 : (-75, 0, -140),
B_4 : (25, 0, 30, 0, -45),

P_5 : (9326, -12300),
A_5 : (-507, 975, -4631, 1820, -6087),
B_5 : (169, -325, 1431, -390, 906, 585, -2035),

P_6 : (260708),
A_6 : 2×(13989, 18450, 40562, 67125, 5149, -9737),
B_6 : 2×(-4663, -6150, -10412, -18275, 9888, 21579, 3820, 3889).

§4. Different determinant representations for APS's

紙面の制約上，勝手ながら省略させて頂きます。

§5. Reduction of S_j, $T_j^{(A)}$ and $T_j^{(B)}$

紙面の制約上，勝手ながら省略させて頂きます。

References

1. D. E. Knuth, The art of computer programming, Vol.2: Seminumerical algorithms, §4.6, 1969, Addison-Wesley.
2. G. E. Collins, Polynomial remainder sequences and determinants, Amer. Math. Mon. 73, No.7, p.708, 1966.
3. G. E. Collins, Subresultants and reduced polynomial remainder sequences, J. ACM 14, No.1, p.128, 1967.
4. W. S. Brown and J. F. Traub, On Euclid's algorithm and the theory of subresultants, J. ACM 18, No.4, p.505, 1971.
5. W. S. Brown, On Euclid's algorithm and the computation of polynomial greatest common divisors, J. ACM 18, No.4, p.478, 1971.
6. W. S. Brown, The subresultant PRS algorithm, ACM Trans. Math. Soft. 4, No.3, p.237, 1978.
7. T. Sasaki, Constructing Bezout's Determinants from Sylvester's determinants, preprint of The Inst. Phys. and Chem. Research, 1982; see, also, T. Sasaki, Y. Kanada and S. Watanabe, Calculation of discriminants of high degree equations, Tokyo J. Math. 4, No.2, p.493, 1981.