J. Symbolic Computation (1992) **13**, 395–411

Three New Algorithms for Multivariate Polynomial GCD

TATEAKI SASAKI AND MASAYUKI SUZUKI

The Institute of Physical and Chemical Research, Wako-shi, Saitama 351-01, *Japan*

(*Received* 7 *April* 1988)

Three new algorithms for multivariate polynomial GCD (greatest common divisor) are given. The first is to calculate a Gröbner basis with a certain term ordering. The second is to calculate the subresultant by treating the coefficients w.r.t. the main variable as truncated power series. The third is to calculate a PRS (polynomial remainder sequence) by treating the coefficients as truncated power series. The first algorithm is not important practically, but the second and third ones are efficient and seem to be useful practically. The third algorithm has been implemented naively and compared with the trial-division PRS algorithm and the EZGCD algorithm. Although it is too early to derive a definite conclusion, the PRS method with power series coefficients is very efficient for calculating low degree GCD of high degree non-sparse polynomials.

1. Introduction

Study of algorithm for multivariate polynomial GCD (greatest common divisor) has a long history. The idea of generalizing the Euclidean algorithm for integer GCD to polynomial GCD appeared in the 16th century. However, Collin's study (1967) is the first modern analysis of the Euclidean algorithm for multivariate polynomial GCD. Collin's algorithm, or the reduced-PRS (polynomial remainder sequence) algorithm, was soon improved to the subresultant-PRS algorithm by Brown & Traub (1971) and Brown (1978). Another improvement of the Euclidean algorithm is Hearn's trial-division algorithm (1979) which is practically efficient. For the GCD computation, modular algorithms are very important. Brown's algorithm (1971) is the first modular GCD algorithm. Subsequently, Moses & Yun (1973) and Wang (1980) presented the so-called EZGCD algorithm. This algorithm utilizes the generalized Hensel construction and is the best algorithm for large multivariate polynomials. For very sparse multivariate polynomials, Zippel's sparse modular algorithm is efficient (Zippel, 1979). Yet another modular algorithm has been presented by Char *et al.* (1984). This algorithm uses the integer GCD computation. Furthermore, an algorithm using Gröbner basis has been presented by Gianni & Trager (1985).

Since the GCD computation is one of the most important operations in computer algebra, we should search for the most efficient algorithm. In this paper, we propose three new algorithms for multivariate GCD. These algorithms are based on simple ideas. The first one calculates a Gröbner basis with a certain term ordering, but it is different from Gianni–Trager's algorithm. In the second and third algorithms, we treat the coefficients, w.r.t. the main variable, of polynomials as truncated power series. This device allows us to develop very efficient GCD algorithms for multivariate polynomials. Since the underlying ideas and the algorithms are very simple, we think that algorithms using truncated power series will be quite useful in actual computation.

We use the following notations in this paper.

P_1, P_2:	polynomials in $K[x, y, \dots, z]$ with K a number field, we assume that P_1 and P_2 are primitive w.r.t. x;
tdeg(P):	term-degree of P: if $T = cy^{e_y} \cdots z^{e_z}$ with $c \in K$ then tdeg(T) $= e_y + \cdots + e_z$, and tdeg(P) = max{tdeg(each term of P)};
deg(P):	degree (w.r.t. main variable x) of polynomial P;
$\deg_u(P)$:	degree w.r.t. sub-variable u of polynomial P;
lc(P):	leading coefficient (w.r.t. main variable x) of P;
pp(P):	primitive part (w.r.t. main variable x) of P;
cont(P):	GCD of the coefficients (w.r.t. main variable x) of P;
coef(P):	coefficients (w.r.t. main variable x) of P.

2. Gröbner Basis Method

Gianni & Trager (1985) proposed a method of using Gröbner basis for multivariate GCD computation. For Gröbner basis, see Buchberger (1985). We propose another algorithm in this section. Our algorithm calculates a Gröbner basis of the ideal (P_1, P_2) in $K[y, \dots, z][x]$, i.e. we regard P_1 and P_2 as polynomials in variable x with coefficients in $K[y, \dots, z]$. (The Gröbner basis in $K[y, \dots, z][x]$ is equivalent to Gröbner basis in $K[x, y, \dots, z]$ with the variable ordering $x > y, \dots, z$. The term ordering in $K[y, \dots, z]$ may be arbitrary.) Our algorithm is based on the following theorem.

THEOREM 1. *Let a Gröbner basis of the ideal* (P_1, P_2) *in* $K[y, \dots, z][x]$ *be* $\Gamma = \{P_1, P_2, \dots, P_s\}$. *Let* $\deg(P_i) = d_i$, $i = 1, \dots, s$, *and* d_k *be the minimum value among* $\{d_1, d_2, \dots, d_s\}$. *Then, there exists a polynomial* C, $C \in K[y, \dots, z]$, *such that* $P_k = C \cdot \mathrm{GCD}(P_1, P_2)$.

PROOF. Put $G = \mathrm{GCD}(P_1, P_2)$. Since $P_k \in (P_1, P_2)$, $P_k = A_k P_1 + B_k P_2$ for some A_k and B_k in $K[y, \dots, z][x]$, hence we have $P_k = CG$ with $C \in K[y, \dots, z][x]$. On the other hand, there exist polynomials A and B, $A, B \in K(y, \dots, z)[x]$, such that $G = AP_1 + BP_2$. Multiplying $\tilde{C}$ = least common multiple (denominators of A and B) to this equation, we see that $\tilde{C}G \in (P_1, P_2)$. Since $\deg(\tilde{C}G) = \deg(G)$ and $\tilde{C}G$ must be M-reduced to 0 by Γ, we have $\deg(P_k) \le \deg(\tilde{C}G) = \deg(G)$. Hence, $C \in K[y, \dots, z]$.

The above theorem gives us the following GCD algorithm.

ALGORITHM 1. (Gröbner basis method.)

```
Step 1. Calculate a Gröbner basis $\Gamma = \{P_1, P_2, \dots, P_s\}$ of the ideal $(P_1, P_2)$ in
        $K[y, \dots, z][x]$;
Step 2. Let $P_k$ be a minimum degree element, w.r.t. $x$, of $\Gamma$ and if $\deg(P_k) = 0$ then return
        1 else return $\mathrm{pp}(P_k)$.
```

NOTE 1. The $\mathrm{pp}(P_k)$ in Step 2 may be calculated efficiently as follows: calculate $g = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$ and construct $\tilde{P} = (gP_k)/\mathrm{lc}(P_k)$, then calculate $\mathrm{pp}(\tilde{P})$.

NOTE 2. The term ordering for terms in $K[y, \dots, z]$ may be arbitrary, but we recommend the term-degree ordering (so-called total-degree ordering) which orders terms according to their term-degrees primarily then lexicographically.

EXAMPLE 1.

$$P_1 = (x-y+2z)((2y+z)x-y^2+2y-3z),$$

$$P_2 = (x-y+2z)((y+2z)x-3y+2z^2).$$

The Gröbner basis of (P_1, P_2) in $Q[y, z][x]$, with the term-degree order for terms in $Q[y, z]$, is calculated as $\{P_3, P_1-2P_2, P_2\}$, where

$$P_3 = (y^3+2y^2z-8y^2+4yz^2-4yz+2z^3+6z^2)x$$
$$+(-y^4+8y^3-12y^2z+6yz^3-14yz^2+4z^4+12z^3).$$

The lowest degree element of the basis is P_3, and we obtain

$$\mathrm{GCD}(P_1, P_2) = \mathrm{pp}(P_3) = x-y+2z.$$

From the viewpoint of variable elimination, the Gröbner basis method is similar to the PRS method: the former uses the head term elimination, and the latter uses the leading term elimination, see Sasaki (1989). However, the PRS method causes intermediate expression growth: in the calculation of the pseudo-remainder, we multiply a power of leading coefficient of the divisor and it is factored out later. The Gröbner basis method does not cause this kind of expression growth, but it generates a number of polynomials and the reduction procedure is time-consuming.

If we know the value of $d = \deg(\mathrm{GCD}(P_1, P_2))$, then we may stop the Gröbner basis construction when a polynomial of degree d is constructed. This device will make the Algorithm 1 quite efficient compared with the original version. However, our experience shows that the Gröbner basis method is quite inefficient when the sizes of P_1 and P_2 are large. This low efficiency can be understood by the fact that the lowest degree element P_k in Γ is a polynomial of almost the same size as $S^{(d)}(P_1, P_2)$, the subresultant of the dth order; for the subresultant, see section 4.

3. Terminology about Truncated Power Series

We use the truncated power series in our second and third algorithms, hence we introduce some terminology about the truncated power series and derive useful degree bounds. We denote the power series ring in the variables $y, \ldots, z$ by $K\{y, \ldots, z\}$, where K is a number field. We impose the reverse term-degree order $>$ for the terms in the power series. Hence, $1 > y > \cdots > z > y^2 > \cdots > yz > \cdots > z^2 > y^3 > \cdots$.

The addition and multiplication of truncated power series are the same as those for polynomials, except for the cut-off of terms whose term-degrees and variable exponents are higher than some prespecified values. With the reverse term-degree order $>$, the division of power series is performed as follows.

DIVISION OF POWER SERIES

Let $A, B, C \in K\{y, \ldots, z\}$. We express A and B as

$$A = A^{(a)} + A^{(a+1)} + A^{(a+2)} + \cdots,$$

$$B = B^{(b)} + B^{(b+1)} + B^{(b+2)} + \cdots,$$

Here $A^{(d)}$ denotes the sum of terms of term-degree d of A. Similarly, we define $B^{(d)}$ and $C^{(d)}$. Let $A=B\cdot C$, then we calculate the quotient $C=C^{(a-b)}+C^{(a-b+1)}+\cdots$ by the power series division of A and B as follows. First, since $A^{(a)}=B^{(b)}C^{(a-b)}$, we calculate $C^{(a-b)}$ by the polynomial division of $A^{(a)}$ by $B^{(b)}$, obtaining

$$A=C^{(a-b)}[B^{(b)}+B^{(b+1)}+\cdots]+[A^{(a+1)}-C^{(a-b)}B^{(b+1)}]$$
$$+[A^{(a+2)}-C^{(a-b)}B^{(b+2)}]+\cdots.$$

Second, we calculate $C^{(a-b+1)}$ by the polynomial division of $[A^{(a+1)}-C^{(a-b)}B^{(b+1)}]$ by $B^{(b)}$, and continue this procedure.

Note. This division makes sense only when B divides A evenly, of course.

Definition 1. (Exponent range and term-degree range.) Let

$$C=\sum_{i=1} c_i u^{e_i}\times(\text{monomial not containing } u),\qquad c_i\neq 0,\quad c_i\in K.$$

We define the exponent range of the variable u of C, abbreviated to $\mathrm{ran}_u(C)$, as $\mathrm{ran}_u(C)=(e_{\min}, e_{\max})$ where $e_{\min}=\min\{e_i\mid i=1,2,\ldots\}$ and $e_{\max}=\max\{e_i\mid i=1,2,\ldots\}$. Similarly, we define the term-degree range of C, abbreviated to $\mathrm{tran}(C)$, as $\mathrm{tran}(C)=(E_{\min}, E_{\max})$ where $E_{\min}(E_{\max})$ is the minimum (maximum) term-degree of terms of C.

Example. For $C=y+2y^2+3yz-y^3-3yz^2+3y^3z-4y^2z^2+5yz^3$, we have $\mathrm{ran}_y(C)=L(1,3)$, $\mathrm{ran}_z(C)=(0,3)$, and $\mathrm{tran}(C)=(1,4)$.

Lemma 1. For polynomials/finite power series C_1 and C_2, we have

$$\begin{cases}\mathrm{ran}_u(C_1C_2)=\mathrm{ran}_u(C_1)+\mathrm{ran}_u(C_2),\\ \mathrm{tran}(C_1C_2)=\mathrm{tran}(C_1)+\mathrm{tran}(C_2),\end{cases}\tag{1}$$

where the addition of numeric lists is $(m_1, n_1)+(m_2, n_2)=(m_1+m_2, n_1+n_2)$.

Proof. Denoting highest (lowest) degree terms of C as hterms(C) (lterms(C)), we have

$$\begin{cases}\mathrm{hterms}(C_1\cdot C_2)=\mathrm{hterms}(C_1)\cdot\mathrm{hterms}(C_2),\\ \mathrm{lterms}(C_1\cdot C_2)=\mathrm{lterms}(C_1)\cdot\mathrm{lterms}(C_2).\end{cases}\tag{2}$$

From these, (1) is obvious.

Definition 2. (Significant terms.) Let C and $\tilde{C}$ be in $K\{y,\ldots,z\}$ and such that high degree terms of C are cut-off to give $\tilde{C}$. Let $\mathrm{tran}(C)=(E, E+\mathit{some})$, let C agree with $\tilde{C}$ on all the terms having term-degree in the range $(0, E+\tilde{E})$ and u-degree in the range $(0, E+\tilde{e}_u)$ for every u in $\{y,\ldots,z\}$ and let $\tilde{C}$ have no term outside these ranges. Then, we say that $\tilde{C}$ is $(\tilde{E}, \tilde{e}_y,\ldots,\tilde{e}_z)$ significant w.r.t. C. All the terms of $\tilde{C}$ are called $(\tilde{E}, \tilde{e}_y,\ldots,\tilde{e}_z)$ significant terms of C. (In the following, "w.r.t. C" will be omitted.)

Note. Apparently, we have $\tilde{e}_u\leq\tilde{E}$. Hence, if $\tilde{E}'$ and $\tilde{e}'_u$ are estimated values of $\tilde{E}$ and $\tilde{e}_u$, respectively, such that $\tilde{e}'_u\geq\tilde{E}'$, we may omit the exponent bound for u.

NOTATION

We express the relation between C and $\tilde{C}$ in Definition 2 as

$$C \equiv \tilde{C} \qquad (\text{upto } (\tilde{E}, \tilde{e}_y, \dots, \tilde{e}_z)).$$

We use the same terminology for $P \in K\{y, \dots, z\}[x]$ also, where the tran and ran are defined for the coefficients of P. Let Op be a rational operation on power series and F, $G \in K\{y, \dots, z\}[x]$. If the Op is applied to significant terms of F and G, and the operation is performed with $(E, e_y, \dots, e_z)$ significant terms, then we express this calculation as

$$\mathrm{Op}(F, G) \qquad (\text{upto } (E, e_y, \dots, e_z)).$$

(The above relation is nothing but the congruence relation. So, if we introduce a variable representing the term-degree, for example the variable t such that $y^{e_y} \cdots z^{e_z} \to t^{e_y + \cdots + e_z} y^{e_y} \cdots z^{e_z}$, then the above relation can be represented by "mod". However, "upto" is more convenient than "mod" in our case, because the term-degree range will change during the calculation of PRS.)

Example. Let $C = y + 2y^2 + 3yz - y^3 - 3yz^2 + 3y^3z - 4y^2z^2 + 5yz^3$, and $\tilde{C} = y + 2y^2 + 3yz - 3yz^2$, then $C \equiv \tilde{C}$ (upto $(2, 1, 2)$).

Let P_1 and P_2 be power series such that P_i is $(E_i, e_{iy}, \dots, e_{iz})$ significant, $i = 1, 2$. Let $P = \mathrm{Op}(P_1, P_2)$, with Op an arithmetic operation on power series, and let P be $(E, e_y, \dots, e_z)$ significant. For multiplication and division, we have

$$E = \min\{E_1, E_2\}, \qquad e_u = \min\{e_{1u}, e_{2u}\}, \qquad u = y, \dots, z.$$

For addition and subtraction, we must be careful. Let $\mathrm{tran}(P_i) = (L_i, H_i)$, $i = 1, 2$, and $\mathrm{tran}(P) = (L, H)$. Then, $L = \min\{L_1, L_2\}$ if $L_1 \neq L_2$, but $L \geq \min\{L_1, L_2\}$ if $L_1 = L_2$. If $L \neq \min\{L_1, L_2\}$, i.e. all the lowest term-degree terms of P_1 and P_2 cancel, then we call the phenomenon *accuracy decreasing*. With this in mind, we have

$$E = \min\{E_1 + L_1 - L, E_2 + L_2 - L\},$$

$$e_u = \min\{e_{1u} + L_1 - L, e_{2u} + L_2 - L\}, \qquad u = y, \dots, z.$$

Note that the cancellation of higher degree terms does not cause the accuracy decreasing, so long as the lowest term-degree term survives.

Example. Let P_1 and P_2 be

$$P_1 = 1 + (y + 2z) + (3y^2 - 4yz + 5z^2),$$

$$P_2 = -1 - (y + 2z) + (5y^2 + 4yz + 3z^2),$$

which are $(2, 2, 2)$ significant. However,

$$P = P_1 + P_2 = 0 + 0 + (8y^2 + 8z^2),$$

which is only $(0, 0, 0)$ significant.

Lemma 2. *Let C_1, C_2, D be in $K\{y, \dots, z\}$ and satisfy $C_1 = C_2 D$. Let $\mathrm{tran}(D) = (E_l, E_h)$ and $\mathrm{ran}_u(D) = (\text{some}, e_u)$ for $u = y, \dots, z$. Then, in order to calculate D from C_1 and C_2 by the power series division, we need only $(E_h - E_l, e_y - E_l, \dots, e_z - E_l)$ significant terms of C_1 and C_2.*

Proof. Obvious from the above division operation for power series.

From now on, by tdeg(coef(P)) and $\deg_u$(coef(P)) with $P \in K\{y, \ldots, z\}[x]$, we mean the maximum of the term-degrees and u-degrees, respectively, of coefficients, w.r.t. x, of P.

Lemma 3. *For P_1 and P_2 in $K\{y, \ldots, z\}[x]$, put $G = \mathrm{GCD}(P_1, P_2)$. Let*

$$E = \mathrm{tdeg}(\mathrm{coef}(G)), \qquad E'' = \mathrm{tdeg}(\mathrm{lc}(G)),$$
$$E_i = \mathrm{tdeg}(\mathrm{coef}(P_i)), \qquad E'_i = \mathrm{tdeg}(\mathrm{lc}(P_i)), \qquad i = 1, 2.$$

Furthermore, for each variable u in $\{y, \ldots, z\}$, let

$$e_u = \deg_u(\mathrm{coef}(G)), \qquad e''_u = \deg_u(\mathrm{lc}(G)),$$
$$e_{ui} = \deg_u(\mathrm{coef}(P_i)), \qquad e'_{ui} = \deg_u(\mathrm{lc}(P_i)), \qquad i = 1, 2.$$

Then, we have

$$E \le \min\{E_i - E'_i + E'' \mid i = 1, 2\}, \tag{3}$$
$$e_u \le \min\{e_{ui} - e'_{ui} + e''_u \mid i = 1, 2\}. \tag{4}$$

Proof. Consider the equality $P_i = G \cdot (P_i/G)$, $i = 1, 2$. Let $\mathrm{tdeg}(\mathrm{lc}(P_i/G)) = \tilde{E}_i$, then $E'_i = \tilde{E}_i + E''$ because $\mathrm{lc}(P_i) = \mathrm{lc}(P_i/G) \times \mathrm{lc}(G)$. Since

$$\mathrm{tdeg}(\mathrm{coef}(P_i)) = \mathrm{tdeg}(\mathrm{coef}(G)) + \mathrm{tdeg}(\mathrm{coef}(P_i/G)),$$
$$\mathrm{tdeg}(\mathrm{coef}(P_i/G)) \ge \mathrm{tdeg}(\mathrm{lc}(P_i/G)),$$

we obtain (3). Similarly, we can derive (4).

Note that Lemma 3 is useless in actual GCD computation because we do not know lc(G) in advance. However, the following Lemma 4 is useful.

Lemma 4. *For P_1 and P_2 in $K\{y, \ldots, z\}[x]$, put $G = \mathrm{GCD}(P_1, P_2)$, $g = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$, $\gamma = g/\mathrm{lc}(G)$, and $\tilde{P} = \gamma G$. Let*

$$E = \mathrm{tdeg}(\mathrm{coef}(\tilde{P})), \qquad E'' = \mathrm{tdeg}(g),$$
$$E_i = \mathrm{tdeg}(\mathrm{coef}(P_i)), \qquad E'_i = \mathrm{tdeg}(\mathrm{lc}(P_i)), \qquad i = 1, 2.$$

Furthermore, for each variable u in $\{y, \ldots, z\}$, let

$$e_u = \deg_u(\mathrm{coef}(\tilde{P})), \qquad e''_u = \deg_u(g),$$
$$e_{ui} = \deg_u(\mathrm{coef}(P_i)), \qquad e'_{ui} = \deg_u(\mathrm{lc}(P_i)), \qquad i = 1, 2.$$

Then, we have

$$E \le \min\{E_i - E'_i + E'' \mid i = 1, 2\}, \tag{5}$$
$$e_u \le \min\{e_{ui} - e'_{ui} + e''_u \mid i = 1, 2\}. \tag{6}$$

Proof. This is an immediate consequence of Lemma 3, because E and E'' here are obtained from E and E'' in Lemma 3 by multiplying γ to G.

4. Subresultant Method with Power Series Coefficient

Let P_1 and P_2 be represented as

$$\begin{cases} P_1 = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_0, & a_m \neq 0, \\ P_2 = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_0, & b_n \neq 0, \end{cases}$$

where we assume $m \geq n$. We define the polynomial $S^{(j)}$ as

$$S^{(j)} = D_j^{(j)}x^j + D_{j-1}^{(j)}x^{j-1} + \cdots + D_0^{(j)}, \tag{7}$$

where $D_i^{(j)}$, $i = j, j-1, \ldots, 0$, are the following determinants:

$$D_i^{(j)} = \begin{vmatrix} a_m & a_{m-1} & \cdots & \cdots & \cdots & a_{2j+2-n} & a_{i+j+1-n} \\ & a_m & a_{m-1} & \cdots & \cdots & a_{2j+3-n} & a_{i+j+2-n} \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & a_m & \cdots & a_{j+1} & a_i \\ b_n & b_{n-1} & \cdots & \cdots & \cdots & b_{2j+2-m} & b_{i+j+1-m} \\ & b_n & b_{n-1} & \cdots & \cdots & b_{2j+3-m} & b_{i+j+2-m} \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & b_n & \cdots & b_{j+1} & b_i \end{vmatrix} \tag{8}$$

Here, we mean $a_i = b_i = 0$ if $i < 0$. The $S^{(j)}$ is the jth order subresultant of P_1 and P_2, and $\deg(S^{(j)})$ is usually j.

THEOREM 2. *Let* $\deg(\mathrm{GCD}(P_1, P_2)) = d$ *and* $g = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$. *Then,*

$$g \mid D_i^{(j)}, \qquad i = j-1, \ldots, 0, \tag{9}$$

$$[D_d^{(d)}/g] \mid D_i^{(d)}, \qquad i = d, \ldots, 0. \tag{10}$$

PROOF. Expanding the determinant in Eq. (8) w.r.t. the first column, we see $\mathrm{GCD}(a_m, b_n) \mid D_i^{(j)}$. This proves (9). Next, we note that $S^{(d)}$ is a multiple of $G = \mathrm{GCD}(P_1, P_2)$ (see, for example, Brown & Traub, 1971). Hence, $\tilde{P} = S^{(d)}/[D_d^{(d)}/g]$ is a polynomial in x such that $\tilde{P} = CG$ with $C = g/\mathrm{lc}(G)$. Since $\mathrm{lc}(G) \mid g$, we see $\tilde{P} \in K[x, y, \ldots, z]$. This proves (10).

Suppose we know the value of $d = \deg(\mathrm{GCD}(P_1, P_2))$. We can use a modular method to calculate d cheaply: calculate $\mathrm{GCD}(P_1(x, n_y, \ldots, n_z), P_2(x, n_y, \ldots, n_z))$ for several sets of numbers $(n_y, \ldots, n_z)$ and set d to the lowest degree of the GCDs calculated. (The value d thus obtained may not be correct, but the possibility of encountering an unlucky case is practically very small.) Then, we construct the subresultant $S^{(d)}$ by calculating the determinants $D_i^{(d)}$, $i = d, d-1, \ldots, 0$, defined by Eq. (8), and we can obtain $G = \mathrm{GCD}(P_1, P_2)$ as $G = \mathrm{pp}(S^{(d)})$. This method has been known for many years, but it is not efficient (in practice).

We have seen above that g divides $\mathrm{lc}(S^{(d)})$ and

$$\tilde{P} = S^{(d)}/[\mathrm{lc}(S^{(d)})/g] \tag{11}$$

is a multiple of G, hence $G = \mathrm{pp}(\tilde{P})$. Representing $\tilde{P}$ as

$$\tilde{P} = g_d x^d + g_{d-1}x^{d-1} + \cdots + g_0, \tag{12}$$

we see that

$$g_d = g, \qquad g_i = D_i^{(d)}/[D_d^{(d)}/g], \qquad i = d-1, \ldots, 0. \tag{13}$$

Eqs (12) and (13) show that what we need are g_d and g_i, $i = d-1, \ldots, 0$, and not $D_d^{(d)}$ and $D_i^{(d)}$ themselves. The sizes of g_i, $i = d, \ldots, 0$, are usually much smaller than those of $D_d^{(d)}$ and $D_i^{(d)}$. The calculation of g_i by Eq. (13) does not require all the terms of $D_d^{(d)}$ and $D_i^{(d)}$ but we need only some lower exponent terms (or higher exponent terms). Exploiting this fact, we may discard the unnecessary terms in the calculation of determinants $D_d^{(d)}$ and $D_i^{(d)}$, which will make the calculation fairly efficient. We discard the unnecessary terms systematically by treating the coefficients of P_1 and P_2 as truncated power series.

Let E, $e_y, \ldots, e_z$ be defined as in Lemma 4. Lemma 2 shows that, in order to calculate $\tilde{P}$, we have only to calculate the determinants $D_i^{(d)}$, $i = d, d-1, \ldots, 0$, up to $(E, e_y, \ldots, e_z)$ significant terms. Note that, since P_1 and P_2 are primitive, we have

$$\begin{cases} \operatorname{ran}_u(P_i) = (0, \mathit{some}),\ i = 1, 2, \\ \operatorname{ran}_u(G) = (0, \mathit{some}), \end{cases} \tag{14}$$

for each variable $u \in \{y, \ldots, z\}$. Thus, we obtain the following algorithm.

Algorithm 2. (Subresultant method with power series coefficient.)

```
Step 1. Estimate $d = \deg(\mathrm{GCD}(P_1, P_2))$ by a modular method (hence, the estimated
        value is an upper bound);
        If $d = 0$ then return 1
          else $g \leftarrow \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$,
            $E \leftarrow \min\{E_i - E_i' + E'' \mid i = 1, 2\}$,
            $e_u \leftarrow \min\{e_{ui} - e_{ui}' + e_u'' \mid i = 1, 2\}$,   $u \in \{y, \ldots, z\}$,
            where $E_i$ etc. are defined in Lemma 4;
Step 2. Construct determinants $D_i^{(d)}$, $i = d, d-1, \ldots, 0$, of the $d$th order subresultant
        $S^{(d)}$, and calculate the determinants $D_i^{(d)}$ up to $(E, e_y, \ldots, e_z)$ significant terms
        for $(y, \ldots, z)$;
Step 3. Calculate $\tilde{P} = \sum_{i=0}^{d} x^i D_i^{(d)}/[D_d^{(d)}/g]$ by the power series division up to
        $(E, e_y, \ldots, e_z)$ significant terms;   $G \leftarrow \mathrm{pp}(\tilde{P})$;
Step 4. If $G \mid P_1$ and $G \mid P_2$ then return $G$
          else $d \leftarrow d-1$, and go to Step 2.
```

The determinant $D_i^{(d)}$ is of order $m+n-2j$. In the Appendix, we give a method of reducing the order of determinant to $m-j$ (note that $m \geq n$).

Example 2.

$$\begin{aligned} P_1 &= yx^4 + (-y^2 - yz + 2y + z)x^3 + (-2y^2 - 2yz - z^2 + y + z)x^2 \\ &\quad + (y^3 + y^2 z - y^2 - z^2)x + (yz^2 + z^3 - z^2), \\ P_2 &= zx^4 + (-yz - z^2 + z - 3)x^3 + (4y + 3z - 3)x^2 \\ &\quad + (-y^2 - yz + 2y - z)x + (-y^2 + z^2 + y - z). \end{aligned}$$

We see $g = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2)) = \mathrm{GCD}(y, z) = 1$. Furthermore, $\operatorname{ran}_y(P_1) = (0, 3)$, $\operatorname{ran}_z(P_1) = (0, 3)$, $\operatorname{tran}(P_1) = (1, 3)$, $\operatorname{ran}_y(P_2) = (0, 2)$, $\operatorname{ran}_z(P_2) = (0, 2)$, $\operatorname{tran}(P_2) = (0, 2)$. Hence,

$$e_y \leq \min\{3 - 1 + 0, 2 - 0 + 0\} = 2,$$

$$e_z \leq \min\{3-0+0, 2-1+0\} = 1,$$

$$E \leq \min\{3-1+0, 2-1+0\} = 1.$$

Therefore, we have only to calculate the subresultant up to $(1, 1, 1)$ significant terms. Suppose we found that $\deg(G) = 1$ by a modular method, so we calculate $D_1^{(1)}$ and $D_0^{(1)}$ up to $(1, 1, 1)$ significant terms. The result is

$$\begin{aligned} D_1^{(1)} = {} & (3y^5 - 24y^3z^2 + 48yz^4) \\ & + (-3y^4z^2 - 8y^2z^4 + 16yz^5 + 16z^6) \\ & + (\text{terms of term-degree} \geq 7), \end{aligned}$$

$$\begin{aligned} D_0^{(1)} = {} & (3y^5 - 24y^3z^2 + 48yz^4) \\ & + (-3y^6 - 3y^5z + 21y^4z^2 + 24y^3z^3 - 56y^2z^4 - 32yz^5 + 16z^6) \\ & + (\text{terms of term-degree} \geq 7). \end{aligned}$$

Calculating $D_0^{(1)}/D_1^{(1)}$ up to $(1, 1, 1)$ significant terms, we find

$$D_0^{(1)}/D_1^{(1)} = (1-y-z) + (\text{terms of term-degree} \geq 2).$$

Therefore, we have $\tilde{P} = 1 \cdot x + (1-y-z)$ as a multiple of GCD. Since $\tilde{P}$ is already primitive, we have

$$G = \mathrm{pp}(\tilde{P}) = x + (1-y-z).$$

EXAMPLE 3. Let P_1 and P_2 be defined in Example 2, and

$$P_1 = [y \to y+1, z \to z-1](P_1),$$

$$P_2 = [y \to y+1, z \to z-1](P_2).$$

In this case also, we have only to calculate the subresultant up to $(1, 1, 1)$ significant terms, but the calculation is much simpler than that in Example 2 because each coefficient of P_1 and P_2 contains a constant term. We have

$$D_1^{(1)} = 23 - 38y - 149z + (\text{terms of term-degree} \geq 2),$$

$$D_0^{(1)} = 23 - 61y - 172z + (\text{terms of term-degree} \geq 2).$$

After the power series division of $D_0^{(1)}$ by $D_1^{(1)}$, we have

$$D_0^{(1)}/D_1^{(1)} = (1-y-z) + (\text{terms of term-degree} \geq 2).$$

Therefore, we have

$$\tilde{P} = 1 \cdot x + (1-y-z).$$

NOTE. If we calculate the determinants $D_1^{(1)}$ and $D_0^{(1)}$ fully, we obtain polynomials of 21 and 38 terms, respectively, for P_1 and P_2 in Example 2 and polynomials of 45 and 57 terms, respectively, for P_1 and P_2 in Example 3. Therefore, if we use the subresultant GCD algorithm with x as the main variable, we will face a large expression growth in the above examples.

It is important to note that, although Examples 2 and 3 are essentially the same, the computational efficiency is considerably different in Examples 2 and 3. This is due to a simple fact on power series: the more the term-degree is, the more different terms we have. For example, in $Q\{y, z\}$, we have three different terms y^2, yz, z^2 of term-degree $=2$, while we have five different terms y^4, y^3z, y^2z^2, yz^3, z^4 of term-degree $=4$. We have calculated power series up to $(1, 1, 1)$ significant terms in both Examples 2 and 3, but we handled more terms in Example 2 than in Example 3. This shows that, although we may choose higher degree terms as necessary terms in calculating GCD, we had better handle lower degree terms. This is the reason why we utilize truncated power series for discarding unnecessary terms. Furthermore, it indicates the importance of preprocessing which generates many constants in the coefficients of P_1 and P_2.

5. PRS Method with Power Series Coefficient

As we have seen in section 4, when we calculate a small-sized polynomial by applying rational operations to large polynomials, we can often obtain the answer efficiently by treating the polynomials as power series and cutting off the higher degree terms. We can apply this idea to PRS algorithms also.

Suppose we calculate the PRS $(P_1, P_2, \ldots, P_k \neq 0, P_{k+1} = 0)$ by cutting off the higher degree terms in the coefficients systematically and calculate $\tilde{P}_k = gP_k/\mathrm{lc}(P_k)$ up to $(E, e_y, \ldots, e_z)$ significant terms, where $E, e_y, \ldots, e_z$ are defined in Lemma 4, then Lemma 2 tells that $\tilde{P} = \gamma G$ where $G = \mathrm{GCD}(P_1, P_2)$ and $\gamma = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))/\mathrm{lc}(G)$. The PRS can be calculated by the conventional formula:

$$\begin{cases} \beta_i P_{i+1} \equiv \alpha_i P_{i-1} - Q_i P_i & (\text{upto } (E, e_y, \ldots, e_z)), \\ \alpha_i = \mathrm{lc}(P_i)^{\delta+1}, & \delta = \deg(P_{i-1}) - \deg(P_i), \end{cases} \tag{15}$$

where β_i is determined by the reduced-PRS or the subresultant PRS algorithms (Collins, 1967; Brown, 1978). One may think that, since the higher degree terms are discarded, we may choose $\beta_i = 1$ without causing the expression growth. However, this is not true as we have mentioned at the end of section 4.

The only one problem in the above-mentioned method is the treatment of accuracy decreasing. Note that, even if the accuracy is not decreased in P_i, the accuracy decreasing in $\mathrm{lc}(P_i)$ will cause the accuracy decreasing of the PRS. This is because the accuracy decreasing in $\mathrm{lc}(P_i)$ decreases the accuracy of α_i in Eq. (15) hence that of P_{i+1}. Conversely, if there is no accuracy decreasing in $\mathrm{lc}(P_i)$, $i = 3, \ldots, k$, then the accuracy of PRS is not decreased. We set the following rule on the PRS calculation.

RULE OF PRS CALCULATION. Regardless of whether the accuracy decreasing happens or not, we calculate PRS by preserving all the significant terms and discarding insignificant terms.

LEMMA 5. *Let P_1 and P_2 be in $K\{y, \ldots, z\}[x]$ and $(\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_k \neq 0, \tilde{P}_{k+1} = 0)$ be a PRS such that $P_i \equiv \tilde{P}_i$* (upto $(E, e_y, \ldots, e_z)$), $i = 1, 2$, *and $\tilde{P}_k$ is generated by formula* (15) *with the above rule. Suppose $\tilde{P}_k$ is $(E', e'_y, \ldots, e'_z)$ significant, hence $E - E' = e_y - e'_y = \cdots = e_z - e'_z \geq 0$. Let G be* $\mathrm{GCD}(P_1, P_2)$ *in $K[x, y, \ldots, z]$, then*

$$G \mid \tilde{P}_k \quad (\text{upto } (E', e'_y, \ldots, e'_z)).$$

PROOF. Divisibility is obvious because of modular congruence. Since G is correct upto $(\infty, \infty, \ldots, \infty)$, the division $G \mid \tilde{P}_k$ is correct upto $(E', e'_y, \ldots, e'_z)$.

Let $(E, e_y, \ldots, e_z)$ be determined by Lemma 4, and suppose that we have calculated PRS $(\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_k, \tilde{P}_{k+1} \doteq 0)$ as mentioned in Lemma 5. Since $(E, e_y, \ldots, e_z)$ is often an over-estimate, we can calculate G correctly from $\tilde{P}_k$ even if the accuracy decreases, so long as all the terms of G are contained in $(E', e'_y, \ldots, e'_z)$ significant terms of $\tilde{P}_k$. Except in the following very unlucky case, we can check the adquateness of $\tilde{P}_k$ by the trial-division: calculate $\tilde{P} \equiv (g\tilde{P}_k)/\mathrm{lc}(\tilde{P}_k)$ up to as many significant terms as possible, and test if $\mathrm{pp}(\tilde{P})$ divides both P_1 and P_2 in $K[x, y, \ldots, z]$. The unlucky case is that $\deg(\tilde{P}_k) < \deg(G)$ and $\mathrm{pp}(\tilde{P}) \mid G$ in $K[x, y, \ldots, z]$; this case may happen if the higher degree terms disappear in $\tilde{P}_k$ due to accuracy decreasing. Note that this case does not occur if the sequence $(\tilde{P}_1, \tilde{P}_2, \tilde{P}_3, \ldots)$ is *normal*, i.e. $\deg(\tilde{P}_{i+1}) = \deg(\tilde{P}_i) - 1$ for every $i = 2, 3, \ldots$.

We can easily avoid the unlucky case mentioned above by setting a lower bound on the significant terms.

THEOREM 3. *Let* $G = \mathrm{GCD}(P_1, P_2)$, $g = \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$ *and*

$$E_{\min} = \min\{\text{term-degrees of terms of } g\}. \tag{16}$$

Let $(\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_k \neq 0, \tilde{P}_{k+1} = 0)$ *be the PRS defined in Lemma* 5. *Let* $\mathrm{lc}(\tilde{P}_k)$ *be* $(\tilde{E}, \tilde{e}_y, \ldots, \tilde{e}_z)$ *significant and let* $\tilde{P}$ *be* $(g\tilde{P}_k)/\mathrm{lc}(\tilde{P}_k)$ (upto $(\tilde{E}, \tilde{e}_y, \ldots, \tilde{e}_z)$). *If* $\tilde{E} \geq E_{\min}$ *and* $\mathrm{pp}(\tilde{P})$ *divides* P_1 *and* P_2 *in* $K[x, y, \ldots, z]$, *then* $G = \mathrm{pp}(\tilde{P})$.

PROOF. Since $\mathrm{lc}(G) \mid g$ and $\tilde{E} \geq E_{\min}$, Eq. (16) tells that $\mathrm{lc}(G)$ contains non-zero terms in $(\tilde{E}, \tilde{e}_y, \ldots, \tilde{e}_z)$ significant terms of G. Hence, $\deg(G) \leq \deg(\tilde{P})$ because $\deg(\tilde{P}_k) = \deg(\tilde{P})$ and Lemma 5 tells that $G \mid \tilde{P}_k$ (upto $(\tilde{E}, \ldots)$). On the other hand, assumption in theorem means $\mathrm{pp}(\tilde{P}) \mid G$. Hence, $\deg(\tilde{P}) = \deg(G)$ and we see $\mathrm{pp}(\tilde{P}) = G$ because P_1 and P_2 are primitive.

Thus, we obtain the following algorithm.

ALGORITHM 3. (PRS method with power series coefficient.)
```
Step 1. $g \leftarrow \mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))$;
        $\hat{E} \leftarrow E \leftarrow \min\{E_i - E'_i + E'' \mid i = 1, 2\}$;
        $\hat{e}_u \leftarrow e_u \leftarrow \min\{e_{ui} - e'_{ui} + e''_u \mid i = 1, 2\}$,   $u \in \{y, \ldots, z\}$;
        ($E_i$ etc. are defined in Lemma 4.)
        $E_{\min} \leftarrow \min\{\text{term-degrees of terms of } g\}$;
Step 2. $\tilde{P}_i \leftarrow P_i$ (upto $(E, e_y, \ldots, e_z)$), $i = 1, 2$; /* cut-off higher degree terms */
        Calculate PRS $(\tilde{P}_3, \ldots, \tilde{P}_k \neq 0, \tilde{P}_{k+1} \equiv 0)$ up to as many significant terms as
        possible;
          Let $\mathrm{lc}(\tilde{P}_k)$ be $(\tilde{E}, \tilde{e}_y, \ldots, \tilde{e}_z)$ significant;
Step 3. If $\deg(\tilde{P}_k) = 0$ then return 1
          else $\tilde{P} \leftarrow g\tilde{P}_k/\mathrm{lc}(\tilde{P}_k)$ (upto $(\tilde{E}, \tilde{e}_y, \ldots, \tilde{e}_z)$);
        $G \leftarrow \mathrm{pp}(\tilde{P})$ and if not $(G \mid P_1$ and $G \mid P_2)$ then go to Step 4;
        If the PRS is normal or $\tilde{E} \geq E_{\min}$ then return $G$;
Step 4. $\hat{E} \leftarrow E \leftarrow 2\hat{E} - \tilde{E}$, $\hat{e}_u \leftarrow e_u \leftarrow \hat{e}_u + \hat{E} - \tilde{E}$ for each $u \in \{y, \ldots, z\}$;
        go to Step 2.
```

NOTE 1. Resetting of $\hat{E}$ and $\hat{e}_u$ in Step 4 is made to increase these values by $\hat{E} - \tilde{E}$ which is the amount of accuracy decreasing.

NOTE 2. A bound on $(E, e_y, \ldots, e_z)$ is given also by lowest degree terms, w.r.t. x, of P_1 and P_2. Hence, if the bound $(E, e_y, \ldots, e_z)$ determined by Lemma 4 is too high, it had better investigate the lowest degree terms of P_1 and P_2.

EXAMPLE 4. P_1 and P_2 given in Example 3, that is

$$P_1=(1+y)x^4+(1+y-y^2-yz)x^3+(-1-y+z-2y^2-2yz-z^2)x^2$$
$$+(-2-y+3z+y^2+2yz-z^2+y^3+y^2z)x+(-1+y+3z-2yz-3z^2+yz^2+z^3),$$
$$P_2=(-1+z)x^4+(-4+y+2z-yz-z^2)x^3+(-2+4y+3z)x^2$$
$$+(3+y-2z-y^2-yz)x+(2-y-3z-y^2+z^2).$$

Lemma 4 gives $(E, e_y, e_z)=(1,2,1)\to(1,1,1)$. Hence, we calculate a PRS by handling only constants, *constant* $\times y$ terms, and *constant* $\times z$ terms. We use the formula (18) with $\beta_i=1$ for simplicity.

$$P_3=(-1+z)\cdot P_1-(1+y)\cdot P_2$$
$$\equiv(3+2y-z)x^3+(3-y-5z)x^2+(-1-3y-3z)x+(-1-2y-z),$$
$$P_4=(3+2y-z)^2\cdot P_2-Q_3(x)\cdot P_3$$
$$\equiv(6+10y-17z)x^2+(15+4y-60z)x+(9-9y-46z),$$
$$P_5=(6+10y-17z)^2\cdot P_3-Q_4(x)\cdot P_4$$
$$\equiv3\{(-69+22y+631z)x+(-69+91y+700z)\},$$
$$P_6=(-69+22y+631z)^2\cdot P_4-Q_5(x)\cdot P_5/3\equiv0.$$

Since $g=\mathrm{GCD}(\mathrm{lc}(P_1), \mathrm{lc}(P_2))=1$, we calculate $\tilde{P}$ as

$$\tilde{P}\equiv P_5/\mathrm{lc}(P_5)\equiv1\cdot x+(1-y-z)\quad(\text{upto }(1,1,1)).$$

Since there is no accuracy decreasing, we obtain $G=x+(1-y-z)$.

NOTE. Since degrees of P_1 and P_2 w.r.t. y or z are smaller than $\deg(P_1)$ and $\deg(P_2)$ in the above example, many GCD programs will calculate $\mathrm{GCD}(P_1, P_2)$ by treating y or z as main variable. The above example shows that our algorithm does not cause expression growth even if the PRS becomes a long sequence.

6. Empirical Study of PC-PRS Method

We call the above Algorithm 3 PC-PRS method (Power series Coefficient PRS method). We have implemented the PC-PRS method on the Japanese algebra system GAL. The implementation is rather preliminary without tuning the program. However, we can see the efficiency of our method approximately.

In the actual program, we have utilized GAL's facility of power series arithmetic and introduced a new variable to represent the term-degree of each term when the number of variables is three or more. The higher degree terms are cut-off by only the bound on term-degree, hence only one variable representing the term-degree is treated as power series variable. The program calculates GCDs of polynomials of one variable by the conventional PRS method, and those of two or more variables by the reduced-PRS method. The program preprocesses input polynomials so that they have many constant terms in their coefficients. Furthermore, the program chooses a variable containing the highest exponent as the main variable. If $\mathrm{lc}(P_1) = \text{constant}$ or $\mathrm{lc}(P_2) = \text{constant}$ then the bound $(E, e_y, \ldots, e_z)$ is determined by the lowest degree terms, w.r.t. the main variable, of P_1 and P_2.

We have compared our algorithm with the trial-division PRS and EZGCD algorithms equipped in REDUCE (Hearn, 1985). Tables 1 to 4 show timing data (in milli-second) by these three algorithms, as well as the size of input polynomials and their GCD. As we will see below, the trial-division PRS algorithm is too bad for large-sized polynomials, so we compare PC-PRS with EZGCD mostly.

Let us explain briefly the test problems and the result of comparison.

Problem 1 (Table 1). This problem is an extension of Example 4 in section 5; the variable u, $u \in \{x, y, z\}$, in Example 4 is replaced by $\sum_{i=1}^{m} u^i$. The number of variables is three. Table 1 shows that the theoretical estimation of exponent bound of the GCD is quite good, which makes the PC-PRS method quite effective for this problem.

Problem 2 (Table 2). This problem is taken from Hearn (1979). The degree of GCD is low. Polynomials P_1 and P_2 are very sparse (the number of terms in P_i is n^2). The theoretical estimation of exponent bound is good for this problem also, and PC-PRS is quite effective.

Problem 3 (Table 3). Polynomials G, P_1 and P_2 have higher degree terms and are denser than those of problem 2. The degree of GCD is about half of degrees of P_1 and P_2. The number of variables is n, the term-degree of $P_1(P_2)$ is $2n(2n-1)$, and the number of terms in P_i grows up in proportion to n^4. Although the estimation of term-degree bound is good for this problem, PC-PRS is much inferior to EZGCD. We have two reasons for this result. The first reason is that the bound calculation took a lot of time; in fact, more than 70% of the time is spent for this calculation. (Note that $\mathrm{lc}(P_1) = \text{constant}$, so the bound is calculated by using constant terms of P_1 and P_2, which requires the computation of GCD of large polynomials in $n-1$ variables.) The second reason is that the problem itself is quite favourable to EZGCD algorithm, because factors of P_1 and P_2 are of simple structures.

Problem 4 (Table 4). This problem is constructed arbitrarily so that polynomials have rather high degree terms and are considerably dense, hence the problem is the largest among the four. As Table 4 shows, the bound estimation is not good when $n \geq 3$; the estimated bound is twice as large as the true upper bound. This makes PC-PRS much worse than EZGCD.

In the first column of each Table, the numbers i_n and i_m in $X-(i_n, i_m)$ mean the values of n and m, respectively, for the problem X. The #term and deg mean respectively, the number of terms and term-degree of either P_1, P_2, or G. E is the estimated upper bound by using formula (5)

Table 1. Problem 1. For P_1 and P_2 of Example 4, replace the variables as follows

$$x \to \sum_{i=0}^{m} x^i, \qquad y \to \sum_{i=0}^{m} y^i, \qquad z \to \sum_{i=0}^{m} z^i$$

				P_1		P_2		G		
$(3, m)$	PC-PRS	EZGCD	Trial	#term	deg	#term	deg	#term	deg	E
I-(3, 1)	12	46	45	27	5	20	5	4	1	1
I-(3, 2)	78	132	673	112	10	80	10	6	2	2
I-(3, 3)	230	338	2889	277	15	198	15	9	3	3
I-(3, 4)	571	717	—	544	20	392	20	12	4	4
I-(3, 5)	1321	1582	—	940	25	680	25	15	5	5

Table 2. Problem 2

$$G = \sum_{i=1}^{n} x_i + 1, \qquad P_1 = G\left(\sum_{i=1}^{n} x_i^n - 2\right), \qquad P_2 = G\left(\sum_{i=1}^{n} x_i^{n-1} + 2\right)$$

				P_1		P_2		G		
(n)	PC-PRS	EZGCD	Trial	#term	deg	#term	deg	#term	deg	E
II-(2)	3	16	5	9	3	6	2	3	1	1
II-(3)	11	18	12	16	4	16	3	4	1	1
II-(4)	22	32	44	25	5	25	4	5	1	1
II-(5)	39	41	199	36	6	36	5	6	1	1
II-(6)	62	59	953	49	7	49	6	7	1	1

Table 3. Problem 3

$$G = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i^j + 1, \qquad P_1 = G\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^j - 2\right), \qquad P_2 = G\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^{j-1} + 2\right)$$

				P_1		P_2		G		
(n)	PC-PRS	EZGCD	Trial	#term	deg	#term	deg	#term	deg	E
III-(2)	11	21	7	11	4	10	3	5	2	2
III-(3)	60	58	43	43	6	40	5	10	3	3
III-(4)	359	175	860	125	8	119	7	17	4	4
III-(5)	2 089	442	—	296	10	286	9	26	5	5
III-(6)	12 694	1007	—	607	12	592	11	37	6	6

Table 4. Problem 4

$$G=\left(\sum_{i=1}^{n} x_i+1\right)^m,\qquad P_1=G\left(\sum_{i=1}^{n}\sum_{j\neq i}^{n}\{x_i(x_j-1)\}^m\right),\qquad P_2=G\left(\sum_{i=1}^{n}\sum_{j\neq i}^{n}\{x_i(x_j^2+1)\}^m\right)$$

				P_1		P_2		G		
(n, m)	PC-PRS	EZGCD	Trial	#term	deg	#term	deg	#term	deg	E
IV-(2, 2)	22	40	66	16	6	25	8	6	2	2
IV-(2, 3)	57	83	1 023	29	9	50	12	10	3	3
IV-(2, 4)	124	153	10 704	51	12	81	16	15	4	4
IV-(2, 5)	404	255	—	72	15	122	20	21	5	5
IV-(3, 2)	129	165	—	55	6	100	8	10	2	4
IV-(3, 3)	591	441	—	136	9	270	12	20	3	6
IV-(3, 4)	2 193	990	—	268	12	535	16	35	4	8
IV-(3, 5)	7 658	1992	—	505	15	969	20	56	5	10
IV-(4, 2)	327	447	—	152	6	280	8	15	2	4
IV-(4, 3)	2 184	1488	—	430	9	956	12	35	3	6
IV-(4, 4)	11 956	3878	—	1132	12	2214	16	70	4	8
IV-(4, 5)	51 693	8833	—	2272	15	4688	20	126	5	10
IV-(5, 2)	726	1036	—	315	6	635	8	21	2	4
IV-(5, 3)	7 106	4239	—	1245	9	2620	12	56	3	6

TOWARDS A BETTER IMPLEMENTATION OF PC-PRS METHOD

Our experience shows that the efficiency of the PC-PRS method depends strongly on the bound $(E, e_y, \ldots, e_z)$. If the bound is twice as large as the true bound, say, then the computation time may increase by an order of magnitude for large-sized polynomials. Considering this, we may say that the formulas (5) and (6) in Lemma 4 are not good practically, and we can improve the PC-PRS method largely if we find a good estimation of the exponent bound. In fact, such an improvement has been made recently, to give a very efficient PC-PRS GCD algorithm (Suzuki & Sasaki, 1990).

References

Brown, W. S. (1971). On Euclid's algorithm and the computation of polynomial greatest common divisor. *J. ACM* **18**, 478–503.

Brown, W. S. (1978). The subresultant PRS algorithm. *ACM Trans. Math. Soft.* 4, 237–249.

Brown, W. S., Traub, J. F. (1971). On Euclid's algorithm and the theory of subresultants. *J. ACM* **18**, 505–514.

Buchberger, B. (1985). Gröbner bases: an algorithmic method in polynomial ideal theory. In: (Bose, R., ed.) *Multidimensional Systems Theory*. Reidel Publishing.

Char, B. W., Geddes, K. O., Gonnet, G. H. (1984). GCDHEU: heuristic polynomial GCD algorithm based on integer GCD computation. *Lecture Notes in Computer Science* **174**, 285–296.

Collins, G. E. (1967). Subresultants and reduced polynomial remainder sequences. *J. ACM* **14**, 128–142.

Gianni, P., Trager, B. (1985). GCDs and factoring multivariate polynomials using Gröbner bases. *Lecture Notes in Computer Science* **204**, 409–410.

Hearn, A. C. (1979). Non-modular computation of polynomial GCDs using trial division. *Lecture Notes in Computer Science* **72**, 227–239.

Hearn, A. C. (1985). *REDUCE User's Manual, version 3.2*, The Rand Corporation, U.S.A.

Moses, J., Yun, D. Y. Y. (1973). The EZ GCD algorithm. *Proceedings of ACM'73* pp. 159–166.

Sasaki, T. (1982). Extended Euclidean Algorithm and Determinants. *Preprint of ICPR*.

Sasaki, T. (1989). Some algebraic algorithms based on head term elimination over polynomial rings. *Lecture Notes in Computer Science* **378**, 348–354.

Suzuki, M., Sasaki, T. (1990). Improvements of PC-PRS GCD algorithm. *Preprint of IPCR.*
Wang, P. (1980). The EEZ-GCD algorithm. *SIGSAM Bulletin* **14**, 50–60.
Zippel, R. (1979). Probabilistic algorithms for sparse polynomials. *Lecture Notes in Computer Science* **72**, 216–226.

Appendix

In Sasaki (1982), a method of reducing the order of determinants of the form in Eq. (8) was described. Since the paper has not been published, we describe the method below. The determinant $D_i^{(j)}$ in Eq. (8) is of order $m+n-2j$, and the method reduces it to a determinant of order $m-j$ (note that $m \geq n$).

At the first step of the reduction, the determinant in Eq. (8) is modified as

$$\left|\begin{array}{ccccccccc}
1 &&&&&&&& \\
& \cdot &&&&&&& \\
&& 1 &&&&&& \\
&&& a_m & a_{m-1} & \cdots & \cdots & a_{2j+2-n} & a_{i+j+1-n} \\
&&&& \cdots & \cdots & \cdots & \cdots & \cdots \\
&&&&& a_m & \cdots & a_{j+1} & a_i \\
0 & \cdot & 0 & b_n & b_{n-1} & \cdots & \cdots & b_{2j+2-m} & b_{i+j+1-m} \\
&&&& \cdots & \cdots & \cdots & \cdots & \cdots \\
&&&&& b_n & \cdots & b_{j+1} & b_i
\end{array}\right|. \tag{A1}$$

At the second step, we move the bottom $m-n$ rows of (A1) to the middle:

$$\left|\begin{array}{cccccccccccc}
1 &&&&&&&&&&& \\
& \cdot &&&&&&&&&& \\
&& 1 &&&&&&&&& \\
&&& a_m & \cdots & a_{m+j+1-n} & a_{m+j-n} & \cdots & \cdots & \cdots & a_{2j+2-n} & a_{i+j+1-n} \\
&&&& \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
&&&&& a_m & a_{m-1} & \cdots & \cdots & \cdots & a_{j+1} & a_i \\
0 &&&&&& b_n & \cdots & \cdots & \cdots & b_{j+2-m+n} & b_{i+1-m+n} \\
& \cdot &&&&&& \cdots & \cdots & \cdots & \cdots & \cdots \\
&& 0 &&&&&& b_n & \cdots & b_{j+1} & b_i \\
&&& b_n & \cdots & b_{j+1} & b_j & \cdots & \cdots & \cdots & b_{2j+2-m} & b_{i+j+1-m} \\
&&&& \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
&&&&& b_n & b_{n-1} & \cdots & \cdots & \cdots & b_{j+1-m+n} & b_{i-m+n}
\end{array}\right| \tag{A2}$$

Writing this determinant as

$$\begin{vmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{vmatrix},$$

where M_{ij}, $i=1,2, j=1,2$, are $(m-j)\times(m-j)$ square matrices, we see $M_{11}M_{21}=M_{21}M_{11}$. Hence, Schur's theorem leads us to the last step of the reduction:

$$D_i^{(j)} = (-1)^{(m-n)(n-j)}\,|M_{11}M_{22}-M_{21}M_{12}|. \tag{A3}$$

This is our reduction formula. Below, we rewrite (A3) slightly. We define matrices $\tilde{M}_{11}$, $\tilde{M}_{12}$, $\tilde{M}_{21}$, $\tilde{M}_{22}$ as follows:

$$\tilde{M}_{11} = \begin{bmatrix} a_m & a_{m-1} & \cdots & a_{m+j+1-n} \\ & \cdots & \cdots & \cdots \\ & & a_m & a_{m-1} \\ & & & a_m \end{bmatrix},$$

$$\tilde{M}_{21} = \begin{bmatrix} b_n & b_{n-1} & \cdots & b_{j+1} \\ & \cdots & \cdots & \cdots \\ & & b_n & b_{n-1} \\ & & & b_n \end{bmatrix},$$

$$\tilde{M}_{12} = \begin{bmatrix} a_{m+j-n} & \cdots & \cdots & a_{2j+2-n} & a_{i+j+1-n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m-1} & \cdots & \cdots & a_{j+1} & a_i \end{bmatrix},$$

$$\tilde{M}_{21} = \begin{bmatrix} b_j & \cdots & \cdots & b_{2j+2-m} & b_{j+1-m+n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b_{n-1} & \cdots & \cdots & b_{i+j+1-m} & b_{i-m+n} \end{bmatrix}.$$

The $\tilde{M}_{11}$, $\tilde{M}_{12}$, $\tilde{M}_{21}$, $\tilde{M}_{22}$ are submatrices of M_{11}, M_{12}, M_{21}, M_{22}, respectively. Then, the determinant in (A3) can be written as

$$\begin{vmatrix} b_n & \cdots & \cdots & \cdots & b_{j+2-m+n} & b_{i+1-m+n} \\ & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & b_n & \cdots & b_{j+1} & b_i \\ & & & \tilde{M}_{11}\tilde{M}_{22} - \tilde{M}_{21}\tilde{M}_{12} & & \end{vmatrix} \tag{A4}$$