Floating-Point Gröbner Basis Computation with Ill-conditionedness Estimation*

Tateaki Sasaki[1] and Fujio Kako[2]

[1] Institute of Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305-8571, Japan
sasaki@math.tsukuba.ac.jp
[2] Department of Comp. Sci., Nara Women's University
Nara-shi, Nara 630-8506, Japan
kako@ics.nara-wu.ac.jp

Abstract. Computation of Gröbner bases of polynomial systems with coefficients of floating-point numbers has been a serious problem in computer algebra for many years; the computation often becomes very unstable and people did not know how to remove the instability. Recently, the present authors clarified the origin of instability and presented a method to remove the instability. Unfortunately, the method is very time-consuming and not practical. In this paper, we first investigate the instability much more deeply than in the previous paper, then we give a theoretical analysis of the term cancellation which causes loss of accuracy in various cases. On the basis of this analysis, we propose a practical method for computing Gröbner bases with coefficients of floating-point numbers. The method utilizes multiple precision floating-point numbers, and it removes the drawbacks of the previous method almost completely. Furthermore, we present a practical method of estimating the ill-conditionedness of the input system.

1 Introduction

Algebraic computation of polynomials with floating-point numbers is a recent hot theme in computer algebra, and many works have been done on the approximate GCD (greatest common divisor), on the approximate polynomial factorization, and so on [15]. However, computation of Gröbner bases with floating-point numbers (*floating-point Gröbner bases*, in short) is just at the beginning of research, although it is a very important theme in approximate algebraic computation (*approximate algebra*). There are two kinds of floating-point Gröbner bases: the first kind is where the coefficients of input polynomials are exact (algebraic numbers or real/complex numbers) but we approximate them by floating-point numbers for some reasons, and the second kind is where the coefficients are inexact hence we express them by floating-point numbers. This paper deals with the second kind.

* Work supported in part by Japan Society for the Promotion of Science under Grants 19300001.

D. Kapur (Ed.): ASCM 2007, LNAI 5081, pp. 278–292, 2008.

The first kind of floating-point Gröbner bases were studied by Shirayanagi and Sweedler [11], [12], [13]. The second kind of floating-point Gröbner bases were studied by Stetter [14], Fortuna, Gianni and Trager [5], Traverso and Zanoni [18], [17], Weispfenning [19], Kondratyev, Stetter and Winkler [8], Gonzalez-Vega, Traverso and Zanoni [6], Stetter [16], Bodrato and Zanoni [2], Mourrain and his coworkers [9], and so on. How to compute floating-point Gröbner bases stably was, however, an open problem for many years. A breakthrough was attained recently by [10], in which the authors clarified the origin of instability of computation and proposed a stable method.

According to [10], there are two origins of instability: one is main-term cancellation (for main terms, see the beginning of Subsect. 2.1), and the other is the appearance of fully erroneous terms (the leading digit is an error). In the computation of Gröbner bases, the main terms of two polynomials sometimes cancel one another in the subtraction, causing large numerical errors. The main-term cancellation is often exact, and exact cancellation with floating-point numbers usually yields a fully erroneous term. If a fully erroneous term appears as a leading term, subsequent computation will be fully wrong.

In [10], the authors classified the main-term cancellation into two types, cancellation due to *self-reduction* and *intrinsic cancellation*. Self-reduction is caused by a polynomial with small or large leading term, just as the elimination by a small pivot row causes large cancellations in Gaussian elimination. The numerical errors due to self-reduction are avoidable, as we will explain later. The intrinsic cancellation is similar to cancellation which occurs in ill-conditioned numerical matrix; see Example 1 in Sect. 2. We want to know the amounts of intrinsic cancellations. One reason is that the accuracy of floating-point Gröbner basis is reduced by the amounts. Another reason is that knowing the amounts seems to be crucial for computing *approximate Gröbner bases*; see [10].

In [10], in order to remove the instability of computation due to self-reduction, the authors proposed to replace each small leading coefficient by an independent symbol and, in the case of large leading term, multiply a symbol to the terms other than the leading term. We call this method *symbolic coefficient method*. They remove fully erroneous terms by representing numeric coefficients by "effective floating-point numbers (*efloats*)"; we explain the efloat in Subsect. 4.2. The efloats work quite well. However, the symbolic coefficient method has two serious drawbacks: 1) it is very time-consuming because we must handle polynomials with symbolic coefficients, and 2) it cannot completely remove the errors due to self-reduction, because even a leading term of relative magnitude 0.3, say, may cause considerable errors.

In this paper, we propose a new method for avoiding the errors due to self-reduction. The new method does not introduce any symbol but it employs multiple precision effective floating-point numbers (*big-efloats*), hence the method is much more efficient than the symbolic coefficient method. In the new method, self-reduction is not avoided but we will show that it does not reduce the accuracy of the Gröbner basis computed. Furthermore, we propose a method to estimate the amount of intrinsic cancellation.

2 Instability Due to Self-reduction

First of all, we emphasize that **we compute Gröbner bases by successive eliminations of leading terms**. This is crucial in the following arguments.

By F, G, etc., we denote multivariate polynomials with coefficients of floating-point numbers. The norm of polynomial F is denoted by $\|F\|$; we employ the infinity norm, i.e., the maximum of the absolute values of numerical coefficients of F. For notions on Gröbner bases, we follow [4]. A power product is a term with no coefficient. By $\mathrm{lt}(F)$, $\mathrm{lc}(F)$ and $\mathrm{rt}(F)$ we denote the leading term, the leading coefficient and the reductum, respectively, of F, w.r.t. a term order $\succ$: $F = \mathrm{lt}(F) + \mathrm{rt}(F)$ with $\mathrm{lt}(F) \succ \mathrm{rt}(F)$. By $\mathrm{Spol}(F, G)$ and $\mathrm{Lred}(F, G)$ we denote the S-polynomial of F and G and the reduction of leading term of F by G, respectively. By reduction of F by G, we mean $\mathrm{Lred}(F, G)$. $\mathrm{Lred}(F, G)$ is often expressed as $F \xrightarrow{G} \tilde{F}$. By $F \xrightarrow{G}\!\!\!\!\rightarrow \tilde{F}$ we denote successive reductions of F by G so that $\mathrm{lt}(\tilde{F})$ is no more reducible by G.

We explain intrinsic cancellation by an example. In order to ease the reader to check our computation of examples, we construct examples by converting rational number coefficients into double precision floating-point numbers.

Example 1. Simple example which exhibits intrinsic cancellation.

$$\left\{\begin{array}{l} P_1 = 57/56\, x^2y + 68/67\, xz^2 - 79/78\, xy + 89/88\, x \\ P_2 = \qquad\qquad xyz^3 - \quad xy^2z + \quad xyz \\ P_3 = 56/57\, xy^2 - 67/68\, yz^2 + 78/79\, y^2 - 88/89\, y \end{array}\right\} \tag{2.1}$$

We convert P_1, P_2, P_3 into erroneous polynomials by converting their coefficients into double precision floating-point numbers. Then, we compute a Gröbner basis w.r.t. the total-degree order with $x \succ y \succ z$, using 30-digit floating-point numbers. We obtain the following unreduced Gröbner basis (correct figures are underlined).

$$\left\{\begin{array}{l} \qquad P_1,\ P_2,\ P_3 \ \text{ are unchanged,} \\ P_6 = y^2z^2 - \underline{2.99543694773255264453831970037}0\, xy^2 \\ \qquad\quad - \underline{1.00207821651237482}57674951096740\, y^3 \\ \qquad\quad + \underline{1.99832546917}37245140192885621560\, xy + \ \cdots\ , \\ P_7 = xz^2 - \underline{1.76431634237}04266614293919973\mathrm{20e}{-3}\, yz^2 \\ \qquad\quad - \underline{9.94723245018}680541945733244338\mathrm{0e}{-1}\, xy \\ \qquad\quad + \underline{1.76798297372}619363856479275314\mathrm{80e}{-3}\, y^2 + \ \cdots\ . \end{array}\right.$$

We see that some relative errors have been increased by about 10^4. □

2.1 Clones and Self-reduction Caused by Small Leading Terms

We use notation $F \approx G$ if $\|F - G\| \ll \|G\|$ and $\|F\| = O(\|G\|)$ if $\eta < \|F\|/\|G\| < 1/\eta$, where η is a positive number less than 1 but not much less than 1. (In our computer program, we set $\eta = 0.2$ and specify $\|G\| \ll \|F\|$ to be $\|G\| < 0.2\,\|F\|$.) We call a polynomial F *normal* if $|\mathrm{lc}(F)| = O(\|\mathrm{rt}(F)\|)$. We call a term T of F a *main term* if $\|T\| = O(\|F\|)$.

Definition 1 (clone). *Let R be either* $\mathrm{Spol}(F,G)$, $\mathrm{Lred}(F,G)$ *or* $F \xrightarrow{G}\!\!\!\!\rightarrow R$. *If* $R \approx M\,\mathrm{rt}(G)$, *with* M *a monomial, then* R *is called a* clone *of* G *and denoted by* $\mathrm{clone}(G)$. *Let* $\|F\| = \|G\| = 1$. *We call* $\|R\|/\|\mathrm{rt}(G)\|$ likeness *of the clone.*

Let F_1 and F_2 be normal polynomials and G be a polynomial with small leading term, $|\mathrm{lc}(G)| \ll \|G\|$. Suppose that F_1 and F_2 are reduced by G as

$$F_1 \xrightarrow{G}\!\!\!\!\rightarrow \tilde{F}_1, \quad F_2 \xrightarrow{G}\!\!\!\!\rightarrow \tilde{F}_2 \qquad (F_1 \neq \tilde{F}_1,\ F_2 \neq \tilde{F}_2)\,. \tag{2.2}$$

Then, so long as $|\mathrm{lc}(F_i)|/\|F_i\| \gg |\mathrm{lc}(G)|/\|G\|$ $(i=1,2)$, we usually have

$$\tilde{F}_1 \approx M_1\,\mathrm{rt}(G) \quad \text{and} \quad \tilde{F}_2 \approx M_2\,\mathrm{rt}(G)\,, \tag{2.3}$$

where M_1 and M_2 are the monomial multipliers in the last reductions, hence $\tilde{F}_1$ and $\tilde{F}_2$ are clones of G. We consider $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2)$; we do not consider $\mathrm{Lred}(\tilde{F}_1,\tilde{F}_2)$ or $\mathrm{Lred}(\tilde{F}_2,\tilde{F}_1)$, because $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2) = \mathrm{Lred}(\tilde{F}_1,\tilde{F}_2)$ if $\mathrm{lt}(\tilde{F}_2)\,|\,\mathrm{lt}(\tilde{F}_1)$ and $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2) = -\mathrm{Lred}(\tilde{F}_2,\tilde{F}_1)$ if $\mathrm{lt}(\tilde{F}_1)\,|\,\mathrm{lt}(\tilde{F}_2)$. Let $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2) = \tilde{M}_1\tilde{F}_1 - \tilde{M}_2\tilde{F}_2$, where $\tilde{M}_1$ and $\tilde{M}_2$ are monomials. Note that we may have $\mathrm{lt}(\tilde{F}_i) \succ M_i\,\mathrm{rt}(G)$ $(i \in \{1,2\})$. In order to avoid this case, we assume that

$$\mathrm{lt}(\tilde{F}_1) \approx \mathrm{lt}(M_1\,\mathrm{rt}(G)) \quad \text{and} \quad \mathrm{lt}(\tilde{F}_2) \approx \mathrm{lt}(M_2\,\mathrm{rt}(G))\,. \tag{2.4}$$

Under condition (2.4), we have $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2) \approx \tilde{M}_1M_1\,\mathrm{rt}(G) - \tilde{M}_2M_2\,\mathrm{rt}(G)$, hence we have $\|\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2)\| \approx \|\tilde{M}_1M_1\,\mathrm{rt}(G) - \tilde{M}_2M_2\,\mathrm{rt}(G)\| \ll \|\tilde{M}_1M_1\,\mathrm{rt}(G)\|$. This means that all the main terms of $\tilde{M}_1M_1\,\mathrm{rt}(G)$ and $\tilde{M}_2M_2\,\mathrm{rt}(G)$ nearly cancel each other; the cancellation is exact if

$$\mathrm{lt}(\tilde{F}_1) = \mathrm{lt}(M_1\,\mathrm{rt}(G)) \quad \text{and} \quad \mathrm{lt}(\tilde{F}_2) = \mathrm{lt}(M_2\,\mathrm{rt}(G))\,. \tag{2.5}$$

Obviously, the above argument is valid for the case of $\tilde{F}_1 = \mathrm{Spol}(F_1,G)$ and/or $\tilde{F}_2 = \mathrm{Spol}(F_2,G)$. The above near cancellation of all the main terms in clones was called "self-reduction" in [10].

We must be careful in treating binomials with small leading terms. Let F_1 and F_2 be normal polynomials as given above, and let the reducer G be a binomial with small leading term: $G = g_1T_1 + g_2T_2$ with $|g_1| \ll |g_2|$, where T_1 and T_2 are power products. Then, $\mathrm{Lred}(F_1,G)$ becomes a polynomial with one large term, and so is $\mathrm{Lred}(F_2,G)$. Let $\tilde{F}_i = \mathrm{Lred}(F_i,G) \approx M_iT_2$ $(i=1,2)$, where M_i is a monomial. If $\mathrm{lt}(\tilde{F}_i) \approx M_iT_2$ $(i=1,2)$ then $\mathrm{Spol}(\tilde{F}_1,\tilde{F}_2)$ does not cause self-reduction. Self-reduction occurs only when $\mathrm{lt}(\tilde{F}_i) \succ M_iT_2$ $(i=1,2)$, $\mathrm{lt}(\tilde{F}_1)M_2 \approx \mathrm{lt}(\tilde{F}_2)M_1$ and $|\mathrm{lc}(\tilde{F}_1)|/\|\tilde{F}_1\| \approx |\mathrm{lc}(\tilde{F}_2)|/\|\tilde{F}_2\|$, which is unlikely to occur. We must notice, however, that G generates a polynomial with one large term. If the large term is the leading term then self-reduction may occur later, as we will explain below. Even if the large term is not the leading term, subsequent reductions may generate a polynomial with large leading term.

2.2 Self-reduction in Three Other Cases

Particularly large leading terms can also cause self-reductions, but the situation is pretty different. Let F_1 and F_2 be polynomials with large leading terms, and G be a normal polynomial:

$$|\mathrm{lc}(F_i)| \ll \|\mathrm{rt}(F_i)\| \ \ (i=1,2)\ , \qquad |\mathrm{lc}(G)| = O(\|\mathrm{rt}(G)\|)\ . \tag{2.6}$$

Then, we can express $\mathrm{Lred}(F_i, G)$ $(i=1,2)$ as follows:

$$\mathrm{Lred}(F_i, G) = F_i - \mathrm{lc}(F_i)/\mathrm{lc}(G)\cdot T_i G \approx -\mathrm{lc}(F_i)/\mathrm{lc}(G)\cdot T_i\,\mathrm{rt}(G)\ , \tag{2.7}$$

where T_1 and T_2 are power products. Therefore, $\mathrm{Lred}(F_i, G)$ is a clone of G, and self-reduction may occur in $\mathrm{Spol}(\mathrm{Lred}(F_1, G), \mathrm{Lred}(F_2, G))$. Note that self-reduction requires two polynomials with large leading terms. Therefore, self-reduction by polynomials with large leading terms is not frequent. Note further that the reduction of a polynomial F with a large leading term by a polynomial G with a small leading term generates a clone of very large likeness: the likeness is $(|\mathrm{lc}(F)|/\|\mathrm{rt}(F)\|)\cdot(\|G\|/|\mathrm{lc}(G)|)$.

Polynomial F may be reduced by $G_1, \ldots, G_m$ successively: $F \xrightarrow{G_1}\!\!\!\!\rightarrow \cdots \xrightarrow{G_m}\!\!\!\!\rightarrow \tilde{F}$. Here, $G_1, \ldots, G_m$ are polynomials with small leading terms and the reduction by each G_j $(1 \le j \le m)$ generates a clone(G_j). In this case, we call $\tilde{F}$ an m *multiple clone*, and represent it as clone$(G_1, \ldots, G_m)$.

We have a more complicated self-reduction which we call *paired self-reduction*. Let normal polynomials F_1 and F_2 be reduced, respectively, by G_1 and G_2 which are polynomials with small leading terms: $F_i \xrightarrow{G_i} \tilde{F}_i$ $(i=1,2)$. There may occur self-reduction in $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$, if F_1, F_2, G_1 and G_2 satisfy several conditions which are seldom satisfied. Because of the page limit, we omit the explanation of paired self-reduction.

Example 2. Simple system causing large errors (an example given in [10]).

$$\left\{\begin{array}{l} P_1 = x^3/10.0 + 3.0x^2y + 1.0y^2 \\ P_2 = 1.0x^2y^2 - 3.0xy^2 - 1.0xy \\ P_3 = y^3/10.0 + 2.0x^2 \end{array}\right\}$$

We compute a Gröbner basis w.r.t. the total-degree order with $x \succ y \succ z$, using double precision floating-point numbers, just as we compute a Gröbner basis over $\mathbb{Q}$. We show about two-thirds of the steps.

$$\begin{array}{ll}
1: & \mathrm{Spol}(P_3, P_2) \xrightarrow{P_1}\xrightarrow{P_1}\xrightarrow{P_2}\xrightarrow{P_3}\boxed{\xrightarrow{P_1}}\boxed{P_4} \qquad /*\ P_4 = \mathrm{clone}(P_1) \\
2: & P_4 = x^2y + 29.8\cdots xy^2 + 3.33\cdots y^3 + 10.0xy + 0.333\cdots y^2 \\
3: & P_2 \xrightarrow{P_4}\xrightarrow{P_3}\boxed{\xrightarrow{P_1}}\boxed{\xrightarrow{P_4}}\boxed{P_2'} \qquad /*\ P_2' = \mathrm{clone}(P_1, P_4) \\
4: & P_2' = xy^2 + 0.111\cdots y^3 + 0.334\cdots xy - 0.000041\cdots y^2 \\
5: & \mathrm{Spol}(P_3, P_2') \xrightarrow{P_3}\boxed{\xrightarrow{P_1}}\boxed{\xrightarrow{P_4}}\boxed{\xrightarrow{P_2'}}\xrightarrow{P_3} P_5 \qquad /*\ \textbf{self-reduction} \\
6: & P_5 = x^2 + 7.14\cdots xy + 0.573\cdots y^2 \\
7: & P_4 \xrightarrow{P_5}\xrightarrow{P_2'}\xrightarrow{P_3}\boxed{\xrightarrow{P_5}}\boxed{P_4'} \qquad /*\ P_4' = \mathrm{clone}(P_5) \\
8: & P_4' = xy + 0.0844\cdots y^2 \\
9: & P_2' \xrightarrow{P_4'}\xrightarrow{P_3}\boxed{\xrightarrow{P_5}}\boxed{\xrightarrow{P_4'}} P_2'' \qquad /*\ \textbf{self-reduction}
\end{array}$$

Here, the polynomials boxed show clones and reducers which generate clones; the clones and self-reductions are commented in the right column. The above computation causes a very large cancellation: self-reductions in Steps 5 and 9 cause cancellations of $O(10^8)$ and $O(10^2)$, respectively. Other steps of computation cause almost no cancellation.

In Step 1, $\mathrm{Spol}(P_3, P_2)$ is a polynomial with large leading term and two reductions by P_1 give a clone of very large likeness, but it is erased by the subsequent reduction by P_2; P_3 is a binomial but the reduction by P_3 does not generate a polynomial with a large term, so we do not mind the reduction; the final reduction by P_1 gives a clone, i.e., $P_4 = \mathrm{clone}(P_1)$. In Step 3, the first reduction by P_4 gives a clone but the clone is erased by the subsequent reduction by P_3; the reduction by P_1 gives a clone, and the clone is reduced by P_4 having a small leading term, hence P_2' is a double clone. In Step 5, reductions by P_1 and P_4 give a double clone, and the double clone is reduced by another double clone P_2', hence there occurs self-reduction between double clones. □

We explain why such large cancellations occur in Example 2. P_2' in Step 3 is a double clone generated by successive reductions by P_1 and P_4, and so is the clone(P_1, P_4) obtained in Step 5. Following Theorem 1 in the next section, one may think that the amount of cancellation caused by self-reduction is $O((\|P_1\|/|\mathrm{lc}(P_1)|)(\|P_4\|/|\mathrm{lc}(P_4)|))$. Actually, we encounter a much larger cancellation. The reason for this superficial discrepancy is that, before the reduction by P_1, the polynomial concerned has been reduced by a binomial P_3 with a small leading term. Hence, $\mathrm{Lred}(\mathrm{Lred}(\mathrm{Lred}(\star, P_3), P_1), P_4)$ becomes a polynomial of very large likeness. The analysis in the next section shows that the actual amount of cancellations occurred is $O((\|P_1\|/|\mathrm{lc}(P_1)|)^2 (\|P_3\|/|\mathrm{lc}(P_3)|)^2)$. In fact, the symbolic coefficient computation in [10] shows this symbolically.

3 Analysis of Self-reductions Given in Sect. 2

In [10], we analyzed only the typical self-reduction by single clones. In this section, we analyze the self-reductions given in Sect. 2, in particular, self-reduction by multiple clones.

Following Collins [3], we introduce the concept of *associated polynomial*. Let polynomials P_i $(i = 1, \ldots, n)$ be expressed as $P_i = c_{i1}T_1 + \cdots + c_{im}T_m$, where $T_1, \ldots, T_m$ are power products, and let $M = (c_{ij})$ be an $n \times m$ matrix, where $n \le m$. The polynomial associated with M, which we denote by $\mathrm{assP}(M)$, is defined as follows.

$$\mathrm{assP}\begin{pmatrix} c_{11} & \cdots & c_{1n} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} & \cdots & c_{nm} \end{pmatrix} \stackrel{\mathrm{def}}{=} \sum_{i=0}^{m-n} \begin{vmatrix} c_{11} & \cdots & c_{1,n-1} & c_{1,n+i} \\ \vdots & \ddots & \vdots & \vdots \\ c_{n1} & \cdots & c_{n,n-1} & c_{n,n+i} \end{vmatrix} T_{n+i}\,. \qquad (3.1)$$

3.1 Analysis of Self-reduction by Double Clones

Let polynomials F and F' be expressed as $F = f_1S_1 + f_2S_2 + \cdots + f_mS_m$ and $F' = f'_1S'_1 + f'_2S'_2 + \cdots + f'_mS'_m$, where S_i and S'_i $(i \geq 1)$ are power products satisfying $S_i \succ S_{i+1}$, $S_i = SS'_i$ for some power product S, and $f_1f'_1 \neq 0$ (some of f_j or f'_j $(j > 1)$ may be 0). Let polynomials G and G' be $G = g_1T_1 + g_2T_2 + \cdots + g_nT_n$ and $G' = g'_1T'_1 + g'_2T'_2 + \cdots + g'_nT'_n$, where T_i and T'_i $(i \geq 1)$ are power products satisfying $S_i = TT_i$ and $S'_i = T'T'_i$ for some power products T and T', and $g_1g'_1 \neq 0$ (some of g_j or g'_j $(j > 1)$ may be 0). We consider the case that both F and F' are reduced k times by G and then reduced k' times by G': $F \xrightarrow{G} \cdots \xrightarrow{G}\xrightarrow{G'} \cdots \xrightarrow{G'} \tilde{F}$ and $F' \xrightarrow{G} \cdots \xrightarrow{G}\xrightarrow{G'} \cdots \xrightarrow{G'} \tilde{F}'$, hence $\tilde{F}$ and $\tilde{F}'$ are double clones of G and G'. The next lemma is well known; we can easily prove it by mathematical inductions on k and k' (cf. [3]).

Lemma 1 (well known). *Let F, G and G' be defined as above. Suppose F is reduced k times by G then reduced k' times by G' (only the leading terms are reduced), then the resulting polynomial $\tilde{F}$ can be expressed as (we discard a numerical multiplier)*

$$\tilde{F} \;=\; \mathrm{assP}\begin{pmatrix} f_1 & f_2 & \cdots & f_n & f_{n+1} & \cdots & \cdots \\ g_1 & g_2 & \cdots & g_n & & & \\ & \ddots & \ddots & \cdots & \ddots & & \\ & & g'_1 & g'_2 & \cdots & g'_n & \\ & & & \ddots & \ddots & \cdots & \ddots \end{pmatrix}, \tag{3.2}$$

where the numbers of $(\cdots g_1 \cdots g_n \cdots)$-rows and $(\cdots g'_1 \cdots g'_n \cdots)$-rows are k and k', respectively. Here, polynomials F, G and G' are padded suitably by zero-coefficient terms so that the elements in each column of the matrix (3.2) correspond to the same term, as in (3.1).

Theorem 1. *Let F, F', $\tilde{F}$ and $\tilde{F}'$ be as above, and assume that $\mathrm{lt}(\tilde{F})/\mathrm{lc}(\tilde{F}) = S\,\mathrm{lt}(\tilde{F}')/\mathrm{lc}(\tilde{F}')$, with S a power product. Let $\tilde{F}$ and $\tilde{F}'$ be expressed as in* (3.2) *(for $\tilde{F}'$, we must replace the top row by $(f'_1\ f'_2 \cdots f'_n \cdots)$). Then, $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$ can be factored as*

$$\begin{vmatrix} g_1 & \cdots & g_k & \cdots & g_{k+k'} \\ & \ddots & \ddots & \cdots & \vdots \\ & & g'_1 & \cdots & g'_{k'} \\ & & & \ddots & \vdots \\ & & & & g'_1 \end{vmatrix} \times \mathrm{assP}\begin{pmatrix} f_1 & f_2 & \cdots & f_n & f_{n+1} & \cdots & \cdots \\ f'_1 & f'_2 & \cdots & f'_n & f'_{n+1} & \cdots & \cdots \\ g_1 & g_2 & \cdots & g_n & & & \\ & \ddots & \ddots & \cdots & \ddots & & \\ & & g'_1 & g'_2 & \cdots & g'_n & \\ & & & \ddots & \ddots & \cdots & \ddots \end{pmatrix}, \tag{3.3}$$

where the numbers of $(\cdots\ g_1\ \cdots\ g_n\ \cdots)$-rows and $(\cdots\ g'_1\ \cdots\ g'_n\ \cdots)$-rows in the above matrix are k and k', respectively.

Proof . The coefficient of $S_{k+k'+i}$ term $(i \geq 2)$ in $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$ is

$$\begin{vmatrix} f'_1 & \cdots & f'_k & \cdots & f'_{k+k'} & f'_{k+k'+1} \\ g_1 & \cdots & g_k & \cdots & g_{k+k'} & g_{k+k'+1} \\ & \ddots & \vdots & \ddots & \vdots & \\ & & g'_1 & \cdots & g'_{k'} & g'_{k'+1} \\ & & & \ddots & \vdots & \\ & & & & g'_1 & g'_{1+1} \end{vmatrix} \cdot \begin{vmatrix} f_1 & \cdots & f_k & \cdots & f_{k+k'} & f_{k+k'+i} \\ g_1 & \cdots & g_k & \cdots & g_{k+k'} & g_{k+k'+i} \\ & \ddots & \vdots & \ddots & \vdots & \\ & & g'_1 & \cdots & g'_{k'} & g'_{k'+i} \\ & & & \ddots & \vdots & \\ & & & & g'_1 & g'_{1+i} \end{vmatrix} \tag{3.4}$$

$$- \begin{vmatrix} f_1 & \cdots & f_k & \cdots & f_{k+k'} & f_{k+k'+1} \\ g_1 & \cdots & g_k & \cdots & g_{k+k'} & g_{k+k'+1} \\ & \ddots & \vdots & \ddots & \vdots & \\ & & g'_1 & \cdots & g'_{k'} & g'_{k'+1} \\ & & & \ddots & \vdots & \\ & & & & g'_1 & g'_{1+1} \end{vmatrix} \cdot \begin{vmatrix} f'_1 & \cdots & f'_k & \cdots & f'_{k+k'} & f'_{k+k'+i} \\ g_1 & \cdots & g_k & \cdots & g_{k+k'} & g_{k+k'+i} \\ & \ddots & \vdots & \ddots & \vdots & \\ & & g'_1 & \cdots & g'_{k'} & g'_{k'+i} \\ & & & \ddots & \vdots & \\ & & & & g'_1 & g'_{1+i} \end{vmatrix}.$$

The Sylvester identity allows us to factor the above expression as

$$\Rightarrow \begin{vmatrix} g_1 & \cdots & g_k & \cdots & g_{k+k'} \\ & \ddots & \vdots & \ddots & \vdots \\ & & g'_1 & \cdots & g'_{k'} \\ & & & \ddots & \vdots \\ & & & & g'_1 \end{vmatrix} \cdot \begin{vmatrix} f_1 & \cdots & f_k & \cdots & f_{k+k'} & f_{k+k'+1} & f_{k+k'+i} \\ f'_1 & \cdots & f'_k & \cdots & f'_{k+k'} & f'_{k+k'+1} & f'_{k+k'+i} \\ g_1 & \cdots & g_k & \cdots & g_{k+k'} & g_{k+k'+1} & g_{k+k'+i} \\ & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & g'_1 & \cdots & g'_{k'} & g'_{k'+1} & g'_{k'+i} \\ & & & \ddots & \vdots & \vdots & \vdots \\ & & & & g'_1 & g'_{1+1} & g'_{1+i} \end{vmatrix}. \tag{3.5}$$

This proves the theorem. □

Remark 1. Consider the case that F is reduced k_1 times by G then reduced k'_1 times by G' and F' is reduced k_2 times by G then reduced k'_2 times by G'. If $k_1 > k_2$, for example, then we put $k = k_2$ and treat the result of $k_1 - k_2$ reductions of F as a new F. If $k'_1 \neq k'_2$ then $\tilde{F}$ and $\tilde{F}'$ are not double clones but we must treat them as single clones of G'. □

Remark 2 . Theorem 1 can be generalized easily to the case of multiple clones: $F \xrightarrow{G_1} \cdots \xrightarrow{G_1} \cdots \xrightarrow{G_j} \cdots \xrightarrow{G_j} \tilde{F}$ and $F' \xrightarrow{G_1} \cdots \xrightarrow{G_1} \cdots \xrightarrow{G_j} \cdots \xrightarrow{G_j} \tilde{F}'$, where $\tilde{F} = \mathrm{clone}(G_1, \ldots, G_j)$ and $\tilde{F}' = \mathrm{clone}(G_1, \ldots, G_j)$. □

The above theorem is valid for any G and G', regardless of the magnitudes of leading terms of G and G'. The theorem tells us that term cancellations occur frequently: all the terms that are not proportional to $g_1^k {g'_1}^{k'}$ cancel one another. This cancellation does not cause large errors usually. If $|\mathrm{lc}(G)| \ll \|G\|$ and/or $|\mathrm{lc}(G')| \ll \|G'\|$, however, the term cancellation is the main-term cancellation and it causes large errors. Below, we order-estimate the amount of term cancellation occurring in $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$ in a simple case.

3.2 Estimation of Amount of Main-Term Cancellation

By $\widetilde{D}_1, \widetilde{D}'_1, \widetilde{D}_i$ and $\widetilde{D}'_i$, we denote the determinants representing $\mathrm{lc}(\tilde{F})$, $\mathrm{lc}(\tilde{F}')$, the coefficient of $S_{k+k'+i}$ term of $\tilde{F}$, and the coefficient of $S_{k+k'+i}$ term of $S\tilde{F}'$, respectively, hence the first expression in the proof of Theorem 1 is $\widetilde{D}'_1\widetilde{D}_i - \widetilde{D}_1\widetilde{D}'_i$. Furthermore, by $\widetilde{D}_{1i}$, we denote the determinant of order $k+k'+2$ in the r.h.s. of (3.5). The magnitudes of $\widetilde{D}_1$ etc. change complicatedly as the situation changes, so we assume that the coefficients of F and F' are as follows.

$$f_1 = f'_1 = 1, \qquad f_i = 0 \text{ or } O(1), \quad f'_i = 0 \text{ or } O(1) \quad (i \geq 2). \tag{3.6}$$

Corollary 1. *Let the coefficients of F and F' be as in* (3.6)*. Let reducers G and G' be polynomials with coefficients such that*

$$\begin{aligned} &|g_1| \ll 1, \quad g_2 = \cdots = g_{l-1} = 0, \ |g_l| = O(1), \quad |g_{l+i}| = O(1) \text{ or } 0, \\ &|g'_1| \ll 1, \quad g'_2 = \cdots = g'_{l'-1} = 0, \ |g'_{l'}| = O(1), \quad |g'_{l'+i}| = O(1) \text{ or } 0. \end{aligned} \tag{3.7}$$

Claim 1: when $l = l' = 2$ (hence $g_2 = O(1)$ and $g'_2 = O(1)$), there occurs cancellation of amount $O((1/g_1)^k (1/g'_1)^{k'})$ in the computation of $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$. Claim 2: when $l \geq 3$ and/or $l' \geq 3$ (hence $g_2 = 0$ and/or $g'_2 = 0$), let $|\widetilde{D}_1| = O((g_1)^{\kappa_1}(g'_1)^{\kappa'_1})$, $|\widetilde{D}_i| = O((g_1)^{\kappa_i}(g'_1)^{\kappa'_i})$ and $|\widetilde{D}_{1i}| = O((g_1)^{\tilde{\kappa}}(g'_1)^{\tilde{\kappa}'})$, then there occurs cancellation of amount $O((1/g_1)^{k-\kappa_1-\kappa_i+\tilde{\kappa}}(1/g'_1)^{k'-\kappa'_1-\kappa'_i+\tilde{\kappa}'})$ in the computation of $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$.

Proof. When $l = l' = 2$, consider $\widetilde{D}_1$ for example, which is the determinant constructed from the leftmost $k+k'+1$ columns of matrix in (3.2). The product of diagonal elements gives the main term of $\widetilde{D}_1$, because other terms contain at least one g_1 or g'_1. Similarly, if we consider those $\widetilde{D}_{1i}$ for which $f'_{k+k'+i} \neq 0$, we see that $\widetilde{D}_{1i} = O(1)$. Then, determinants in (3.5) lead us to Claim 1. The determinants also lead us to Claim 2, because the main terms of $\widetilde{D}'_1\widetilde{D}$ and $\widetilde{D}_1\widetilde{D}'_i$ must be of the same order. □

Determination of $\tilde{\kappa}_1, \tilde{\kappa}'_1, \tilde{\kappa}_i, \tilde{\kappa}'_i, \tilde{\kappa}$ and $\tilde{\kappa}'$ in the general case of $l \geq 3$ and/or $l' \geq 3$ is messy. Because of the page limit, we omit the determination.

Theorem 1 allows us to analyze self-reduction caused by polynomials with large leading terms and paired self-reduction, too. For the case of large leading terms, we put $F_1 = F$, $F_2 = F'$ and $G' = G$, and assume that the leading terms of F and F' are large. Then, estimating the magnitudes of determinants in (3.5), we obtain the following corollary which can be easily generalized to the case that F_1 and/or F_2 contain several large terms at their heads.

Corollary 2. *Let F and F' be polynomials with large leading terms and G be a normal polynomial. Put $\tilde{F} = \mathrm{Lred}(F, G)$ and $\tilde{F}' = \mathrm{Lred}(F', G)$. Then, in the computation of $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$, there occurs main-term cancellation of magnitude $\min(|\mathrm{lc}(F)|/\|\mathrm{rt}(F)\|, |\mathrm{lc}(F')|/\|\mathrm{rt}(F')\|)$.*

4 New Method of Stabilization

We consider that the coefficients of input system of polynomials are inexact. If the largest relative error in the coefficients is ε then we say the *accuracy* of the system is ε. Below, by ϵ_{m} we denote the *machine epsilon* (= the difference between 1 and the smallest representable number greater than 1) of double precision floating-point numbers (double-floats). If the coefficients of input system are given by double-floats, we have $\varepsilon \geq \epsilon_{\mathrm{m}}$.

4.1 Supporting Theorem

We will compute the Gröbner basis by converting each input coefficient into a multiple precision floating-point number (big-float). We assume that each big-float is a p-digit decimal number satisfying $10^{-p} \ll \varepsilon$, and put $\epsilon_{\mathrm{M}} = 10^{-p}$.

Theorem 2. *As far as self-reductions treated in Sect. 3 are concerned, the main-term cancellation due to self-reduction ruins only tail figures of the coefficients concerned.*

Proof. We note that, although the big-floats in our case contain relative errors which are much larger than ϵ_{M}, *the errors are introduced initially and the coefficients are treated as definite numbers of the full precision throughout the computation*. On the other hand, Theorem 1 implies that, in the self-reductions considered, the coefficients of main terms cancel exactly within the precision. Hence the self-reductions ruin only tail figures of the coefficients. □

Remark 3. One may think that all the cancellation errors can be avoided if we increase the precision. This is, however, wrong as Example 1 shows. In the self-reductions we have considered, the main terms cancel exactly, which is the key of Theorem 2. □

4.2 Effective Floating-Point Numbers

In actual computation, we must remove the fully erroneous terms and estimate the amount of accuracy loss. Thus, we utilize *multiple precision effective floating-point numbers* (big-efloats), instead of big-floats.

We explain the efloats briefly. The efloat was proposed by the present authors in 1997 [7] so as to detect the cancellation errors automatically. The efloat is a pair of two floating-point numbers and expressed as $\#\mathrm{E}[f, e]$; we call f and e *value-part* and *error-part*, respectively. The arithmetic of efloats is as follows.

$$\begin{aligned}
\#\mathrm{E}[f_a, e_a] + \#\mathrm{E}[f_b, e_b] &\Longrightarrow \#\mathrm{E}[f_a + f_b,\ \max\{e_a, e_b\}],\\
\#\mathrm{E}[f_a, e_a] - \#\mathrm{E}[f_b, e_b] &\Longrightarrow \#\mathrm{E}[f_a - f_b,\ \max\{e_a, e_b\}],\\
\#\mathrm{E}[f_a, e_a] \times \#\mathrm{E}[f_b, e_b] &\Longrightarrow \#\mathrm{E}[f_a \times f_b,\ \max\{|f_b e_a|, |f_a e_b|\}],\\
\#\mathrm{E}[f_a, e_a] \div \#\mathrm{E}[f_b, e_b] &\Longrightarrow \#\mathrm{E}[f_a \div f_b,\ \max\{|e_a/f_b|, |f_a e_b/f_b^2|\}].
\end{aligned} \tag{4.1}$$

Thus, the value-part of efloat is nothing but the conventional floating-point value. On the other hand, the error-part of efloat represents the cancellation error approximately; the rounding errors are neglected in determining the error-part. Similarly, we neglect the rounding errors throughout the following arguments.

The big-efloat is expressed as $\#\mathrm{BE}[f, e]$, where f is a big-float, and it is processed by the same arithmetic as efloat. We set the error-part e to $10^{-p+2}\,|f|$.

We explain how the fully erroneous terms are removed (we explain only for the case of efloats). We set the error-part of each efloat coefficient to $5\varepsilon|f|$ (in the examples, we set to about $5\epsilon_{\mathrm{m}}|f|$). In our algebra system named GAL, the efloat $\#\mathrm{E}[f, e]$ with $|f| < e$ is automatically set to 0 (not $\#\mathrm{E}[0, 0]$). Therefore, GAL removes fully erroneous terms unless the rounding errors accumulate to $5\epsilon_{\mathrm{m}}$ or more, which is extremely rare in practice.

Example 3. Check Theorem 2 by the system in Example 2.

We convert the coefficients into double-floats, and compute a Gröbner basis with big-efloats of 30 decimal precision. For reference, we show the initial polynomials; if all the figures from 17th to the last decimal places are 0, our system outputs only one 0. Note that the rounding errors appear at the 17th decimal places.

$$\begin{cases} P_1 = + \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-2},\ 2.0\mathrm{e}{-28}]\, x^3 + x^2 y \\ \qquad + \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, y^2, \\ P_2 = + \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, x^2 y^2 - x y^2, \\ \qquad - \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, xy \\ P_3 = + \#\mathrm{BE}[5.0000000000000000\mathrm{e}{-2},\ 3.9\mathrm{e}{-28}]\, y^3 + x^2. \end{cases}$$

The $\mathrm{Spol}(P_3, P_1)$, for example, is reduced and normalized as follows; we see that 17th to 30th figures of xy^3 term are contaminated by rounding errors.

$$x^4 + \#\mathrm{BE}[1.5000000000000001665334536937720\mathrm{e}{-1},\ 3.9\mathrm{e}{-28}]\, xy^3 \\ + \#\mathrm{BE}[5.0000000000000000\mathrm{e}{-2},\ 2.0\mathrm{e}{-28}]\, xy^2.$$

We obtain the following unreduced Gröbner base.

$$\begin{cases} P_2'' = y^2, \\ P_4' = xy + \#\mathrm{BE}[\underline{8.4402255045219586}7629311654600\mathrm{e}{-2},\ 3.3\mathrm{e}{-21}]\, y^2, \\ P_5 = x^2 + \#\mathrm{BE}[\underline{7.1484968974627}0700636549331894 0,\ 4.2\mathrm{e}{-19}]\, xy \\ \qquad + \#\mathrm{BE}[\underline{5.7371613952464572}25742044589410\mathrm{e}{-1},\ 2.6\mathrm{e}{-20}]\, y^2. \end{cases}$$

Here, underlines show correct figures. We see that, although large cancellations have occurred, the accuracy loss in the Gröbner basis is only slight. □

4.3 Description of New Method

Now, we describe our new method which is based on Theorem 2 crucially. The method is composed of the following three devices.

Device 1: Convert the numeric coefficients of input polynomials into big-efloats of a suitably determined initial precision, say $p = 30$, and compute the Gröbner basis by using only the leading-term reduction and the S-polynomial construction.

Device 2: Monitor the error-parts of big-efloat coefficients during the computation, and if the amount of the largest cancellation accumulated, let it be C, becomes large satisfying $\epsilon_{\mathrm{M}} C > 10^{-5}\varepsilon$, say, then increase the precision of big-efloats and retry the computation.

Device 3: Monitor the clone generation of likeness greater than 5, say, and self-reduction by such clones. We explain the device for single reduction; for multiple reductions, see Sect. 5. Suppose that self-reduction occurs in the subtraction $\tilde{F}_1 - \tilde{F}_2$ in computing $\mathrm{Spol}(\mathrm{Lred}(F_1, G), \mathrm{Lred}(F_2, G))$ etc. Here, $\tilde{F}_1 = \mathrm{clone}(G)$ and $\tilde{F}_2 = \mathrm{clone}(G)$, hence $\tilde{F}_i = -c_i T \mathrm{rt}(G) + (\text{small-terms})$ $(i = 1, 2)$, where c_1 and c_2 are numbers such that $c_1 \approx c_2$ and T is a power product. Then, we "subtract" $-\mathrm{rt}(G)$ from both $\tilde{F}_1$ and $\tilde{F}_2$ as $\tilde{F}_1' := \tilde{F}_1 + cT\mathrm{rt}(G)$ and $\tilde{F}_2' := \tilde{F}_2 + cT\mathrm{rt}(G)$, where c will be determined in Subsect. 5.2 ($c \approx c_1 \approx c_2$). We call this operation *reducer subtraction*. Regard the possible cancellation occurring in $\tilde{F}_1' - \tilde{F}_2'$ as the intrinsic cancellation.

The number 5 in Device 3 is determined from the following reason: Example 2 shows that we must monitor clones of likeness 10 or more, and it is impractical to monitor clones of likeness 2 or less. With Devices 1 and 2, we can protect the accuracy of the system from self-reduction completely; the number 5 for specifying the clone in Device 3 is irrelevant to this protection. Device 3 is for estimating the intrinsic cancellation; we explain the details in Sect. 5.

As for the intrinsic cancellation, authors of [2] and [10] defined the cancellation in terms of syzygies. The computation of syzygies is quite costly in practice. On the other hand, the reducer subtraction is not a costly operation (see Sect. 5 for implementation), hence our method is practical.

5 Implementation Details

Although our ideas given above are simple, actual implementation of the Device 3 requires various detailed considerations.

5.1 Representation of Clones

In our current program, each input polynomial or S-polynomial generated is numbered uniquely, say F_i $(i \in \mathbb{N})$, and the numbering is not changed if the polynomial is reduced; if F_i is reduced to 0 then F_i is removed from the memory. Suppose a polynomial F_i is reduced by G_j to become a clone of G_j. It is not enough to save the index j to specify the clone; we must save the current G_j because G_j itself might change later during the computation. Let the reduction be $\tilde{F}_i := F_i - c_j T_j G_j$, where $c_j \in \mathbb{C}$ and T_j is a power product. The multiplier c_j changes from the reduction to reduction, hence we must save the multipliers, too. Therefore, we represent clones generated from G_j as follows.

1. Normalize G_j so that its leading coefficient is 1.
2. Represent each clone by a triplet $\langle j, c_j, T_j G_j \rangle$ which we call *clone-triplet*. Construct a clone-triplet each time F_i is reduced.

3. Save the clone-triplets for F_i into a list and attach the list to F_i. For example, if $F_i \xrightarrow{G_j} \xrightarrow{G_{j'}} \cdots$, then the list is $(\cdots \langle j', c'_{j'}, T'_{j'}G_{j'}\rangle \langle j, c_j, T_jG_j\rangle)$.

We normalize not only clones but also each polynomial appearing in the computation so that its leading coefficient is 1, which makes the programming easy. The normalization is made after each reduction (and S-polynomial generation): $\tilde{F}_i := F_i - c_jT_jG_j \longrightarrow \tilde{F}_i := \tilde{F}_i/\mathrm{lc}(\tilde{F}_i)$. Just after this normalization, all the multipliers in the clone-triplet list for F_i must be changed as $\langle j, c_j, T_jG_j\rangle \rightarrow \langle j, c_j/\mathrm{lc}(\tilde{F}_i), T_jG_j\rangle \ \ (j=1,2,\ldots)$.

5.2 Reducer Subtraction

The reducer subtraction is performed as follows. Let $\tilde{F}_i = F_i - c_iTG$ $(i = 1, 2)$, and suppose that self-reduction occurs in $\tilde{F}_1 - \tilde{F}_2$, as in Device 3 (self-reduction occurs actually in $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$ or $\mathrm{Lred}(\tilde{F}_1, \tilde{F}_2)$, but we simplified the situation by multiplying suitable power products to $\tilde{F}_1$ and $\tilde{F}_2$). By computing c as

$$c = \begin{cases} c_1 & \text{if } |c_1| \le |c_2|, \\ c_2 & \text{if } |c_1| > |c_2|, \end{cases} \tag{5.1}$$

we subtract $-\mathrm{rt}(G)$ from $\tilde{F}_i$ $(i = 1, 2)$ as $\tilde{F}_i := \tilde{F}_i + cT\mathrm{rt}(G)$.

The above subtraction is for the reducer used at the last reduction (the leftmost reducer in the clone-triplet list of F_i). For other reducers, the subtraction is made as follows; we explain only for the case that equalities in (2.5) hold. Suppose F_1 is reduced by $G_1, \ldots, G_j$ as $F_1 \xrightarrow{G_1} \cdots \xrightarrow{G_j} \tilde{F}_1$ and we have

$$\tilde{F}_1 := (\cdots (F_1 - c_{11}T_1G_1)\cdots) - c_{1j}T_jG_j.$$

In this case, the leading terms of $\mathrm{rt}(G_1)$ may be eliminated and $\tilde{F}_1$ may contain only $\mathrm{rt}(\cdots \mathrm{rt}(G_1)\cdots)$. Therefore, we scan terms of $\tilde{F}_1$ and $\mathrm{rt}(G_1)$ from highest to lowest order, and determine which $\mathrm{rt}(\cdots \mathrm{rt}(G_1)\cdots)$ is contained in $\tilde{F}_1$. Then, we subtract a suitable multiple of that $\mathrm{rt}(\cdots \mathrm{rt}(G_1)\cdots)$ from $\tilde{F}_1$ (and $\tilde{F}_2$).

5.3 Estimating the Amount of Intrinsic Cancellation

The actual term cancellation is the sum of the intrinsic cancellations and cancellations due to self-reductions. Therefore, if we remove all the cancellations due to self-reductions, then the rest must be the sum of intrinsic cancellations. It should be mentioned that Device 3 will fail to remove small amounts of cancellations due to small self-reductions, because we neglect the clones of likeness < 5. Therefore, the method explained in Device 3 will over-estimate the amount of intrinsic cancellation.

To illustrate our technique we show the estimation of the intrinsic cancellation in Example 2, in particular at the reduction step $\mathrm{Spol}(P_3, P'_2) \xrightarrow{P_3} \xrightarrow{P_1} \xrightarrow{P_4} \cdots$, where self-reduction by double clones occurs and we encounter main-term cancellation of $O(10^{10})$.

Example 4. Intrinsic cancellation in Step 5 of Example 2.

Put $Q_1 = \mathrm{Lred}(\mathrm{Lred}(\mathrm{Lred}(\mathrm{Spol}(P_3, P_2'), P_3), P_1), P_4)$ and let $\mathrm{Lred}(Q_1, P_2') = Q_1 - Q_2$, where $P_2' = \mathrm{clone}(P_1, P_4)$. Below, underlines show figures which are same in both Q_1 and Q_2 (or Q_1' and Q_2', or Q_1'' and Q_2'').

$$\begin{aligned} Q_1 = &+ \#\mathrm{BE}[\underline{1.1152418136}7893093558453405171\mathrm{e}{-1},\ 8.6\mathrm{e}{-28}]\, y^3 \\ &+ \#\mathrm{BE}[\underline{3.3457253}7116428064158048010 40\mathrm{e}{-1},\ 3.7\mathrm{e}{-27}]\, xy \\ &- \#\mathrm{BE}[\underline{4.1613}506289664168782840449950\mathrm{e}{-5},\ 1.1\mathrm{e}{-28}]\, y^2, \\ Q_2 = &+ \#\mathrm{BE}[\underline{1.1152418132}0025535431698179200\mathrm{e}{-1},\ 8.6\mathrm{e}{-28}]\, y^3 \\ &+ \#\mathrm{BE}[\underline{3.3457254}3960076062950945376 00\mathrm{e}{-1},\ 3.7\mathrm{e}{-27}]\, xy \\ &- \#\mathrm{BE}[\underline{4.1612}957039749613089747630908\mathrm{e}{-5},\ 1.1\mathrm{e}{-28}]\, y^2. \end{aligned}$$

A multiple of $-\mathrm{rt}(P_4)$ is subtracted from Q_1 and Q_2; $Q_1' \leftarrow Q_1$ and $Q_2' \leftarrow Q_2$:

$$\begin{aligned} Q_1' = &+ \#\mathrm{BE}[\underline{2.3}290837408651847149108379154\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\ &- \#\mathrm{BE}[\underline{1.1194031}4101705996407177741 30\mathrm{e}{-2},\ 1.1\mathrm{e}{-28}]\, y^2, \\ Q_2' = &+ \#\mathrm{BE}[\underline{2.2}812159995976324552013022754\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\ &+ \#\mathrm{BE}[6.843647998792897365603906826 4\mathrm{e}{-9},\ 3.7\mathrm{e}{-27}]\, xy \\ &+ \#\mathrm{BE}[\underline{1.1194030}8609206850850246813 10\mathrm{e}{-2},\ -1.1\mathrm{e}{-28}]\, y^2. \end{aligned}$$

A multiple of $-\mathrm{rt}(P_1)$ is subtracted from Q_1' and Q_2'; $Q_1'' \leftarrow Q_1'$ and $Q_2'' \leftarrow Q_2'$:

$$\begin{aligned} Q_1'' = &+ \#\mathrm{BE}[\underline{2.3}290837408651847149108379154\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3, \\ Q_2'' = &+ \#\mathrm{BE}[\underline{2.2}812159995976324552013022754\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\ &+ \#\mathrm{BE}[6.843647998792897365603906826 4\mathrm{e}{-9},\ 3.7\mathrm{e}{-27}]\, xy \\ &+ \#\mathrm{BE}[5.492499145556930928190424214 8\mathrm{e}{-10},\ 1.1\mathrm{e}{-28}]\, y^2. \end{aligned}$$

We see $O(10^2)$ cancellation occurs in $Q_1'' - Q_2''$ which we regard as the intrinsic cancellation. □

6 Concluding Remarks

For the page limit of LNCS, several parts were omitted in this paper. See [1] for the omitted parts.

We showed that, restricting the reductions to leading-term reductions, we are able to describe local steps of Gröbner basis computation by matrices and analyze self-reduction and intrinsic cancellation in terms of determinants (Theorem 1). Furthermore, we showed that the main-term cancellation due to self-reduction causes no problem if we utilize big-efloats, as far as the self-reductions investigated in Sect. 2 are concerned (Theorem 2). We are now trying to prove that any self-reduction causes no problem.

Our analysis suggests us that the cancellation errors will be decreased largely if self-reduction is avoided as far as possible. We are now developing a program package based on this suggestion.

Finally, the authors acknowledge anonymous referees for valuable comments.

References

1. Sasaki, T., Kako, F.: Floating-point Gröber Basis Computation with Ill-conditionedness Estimation. Technical Report of Univ. of Tsukuba, in (December 2007), http://www.math.tsukuba.ac.jp/~sasaki/papers/ASCM2007
2. Bodrato, M., Zanoni, A.: Intervals, syzygies, numerical Gröbner bases: a mixed study. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2006. LNCS, vol. 4194, pp. 64–76. Springer, Heidelberg (2006)
3. Collins, J.E.: Subresultant and reduced polynomial remainder sequence. J. ACM 14, 128–142 (1967)
4. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer, New York (1997)
5. Fortuna, E., Gianni, P., Trager, B.: Degree reduction under specialization. J. Pure Appl. Algebra 164, 153–164 (2001)
6. Gonzalez-Vega, L., Traverso, C., Zanoni, A.: Hilbert stratification and parametric Gröber bases. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2005. LNCS, vol. 3718, pp. 220–235. Springer, Heidelberg (2005)
7. Kako, F., Sasaki, T.: Proposal of "effective" floating-point number. Preprint of Univ. Tsukuba (May 1997) (unpublished)
8. Kondratyev, A., Stetter, H.J., Winkler, S.: Numerical computation of Gröbner bases. In: Proceedings of CASC 2004 (Computer Algebra in Scientific Computing), St. Petersburg, Russia, pp. 295–306 (2004)
9. Mourrain, B.: Pythagore's dilemma, symbolic-numeric computation, and the border basis method. In: Symbolic-Numeric Computations (Trends in Mathematics), pp. 223–243. Birkhäuser Verlag, Basel (2007)
10. Sasaki, T., Kako, F.: Computing floating-point Gröbner base stably. In: Proceedings of SNC 2007 (Symbolic Numeric Computation), London, Canada, pp. 180–189 (2007)
11. Shirayanagi, K.: An algorithm to compute floating-point Gröbner bases. In: Mathematical Computation with Maple V. Ideas and Applications, pp. 95–106. Birkhäuser, Basel (1993)
12. Shirayanagi, K.: Floating point Gröbner bases. Mathematics and Computers in Simulation 42, 509–528 (1996)
13. Shirayanagi, K., Sweedler, M.: Remarks on automatic algorithm stabilization. J. Symb. Comput. 26, 761–765 (1998)
14. Stetter, H.J.: Stabilization of polynomial systems solving with Gröbner bases. In: Proceedings of ISSAC 1997 (Intern'l Symposium on Symbolic and Algebraic Computation), pp. 117–124. ACM Press, New York (1997)
15. Stetter, H.J.: Numerical Polynomial Algebra. SIAM Publ., Philadelphia (2004)
16. Stetter, H.J.: Approximate Gröbner bases – an impossible concept? In: Proceedings of SNC 2005 (Symbolic-Numeric Computation), Xi'an, China, pp. 235–236 (2005)
17. Traverso, C.: Syzygies, and the stabilization of numerical Buchberger algorithm. In: Proceedings of LMCS 2002 (Logic, Mathematics and Computer Science), RISC-Linz, Austria, pp. 244–255 (2002)
18. Traverso, C., Zanoni, A.: Numerical stability and stabilization of Gröbner basis computation. In: Proceedings of ISSAC 2002 (Intern'l Symposium on Symbolic and Algebraic Computation), pp. 262–269. ACM Press, New York (2002)
19. Weispfenning, V.: Gröbner bases for inexact input data. In: Proceedings of CASC 2003 (Computer Algebra in Scientific Computing), Passau, Germany, pp. 403–411 (2003)