Floating-point Gröbner Base Computation with Ill-conditionedness Estimation *

Tateaki Sasaki †) and Fujio Kako ‡)

†) Institute of Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305-8571, Japan
sasaki@math.tsukuba.ac.jp
‡) 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 a long time; 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 large errors, in various cases. On the basis of this analysis, we propose a practical method for computing the 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 method of estimating the ill-conditionedness of the input system.

Key words and phrases: algebraic-numeric computation, approximate algebraic computation, cancellation error, floating-point Gröbner base, Gröbner base, instability, stabilization, symbolic-numeric computation.

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

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 [14]. 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 such that the coefficients of input polynomials are exact (algebraic numbers or real/complex numbers) but we approximate them by floating-point numbers for convenience, and the second kind is such that the coefficients are inexact hence we express them by floating-point numbers. This paper deals with the second kind.

The first kind floating-point Gröbner bases were studied by Shirayanagi and Sweedler [9, 10, 12]. The second kind floating-point Gröbner bases were studied by Stetter [13], Fortuna, Gianni and Trager [4], Traverso and Zanoni [17], Traverso [16], Weispfenning [18], Kondratyev, Stetter and Winkler [7], Gonzalez-Vega, Traverso and Zanoni [5], Stetter [15], Bodrato and Zanoni [1], Mourrain and his coworkers [8], and so on. It was, however, a serious problem for long years. A breakthrough was attained recently by [11], in which the authors clarified the origin of instability of computation and proposed a stable method.

According to [11], there are two origins of instability: one is main-term cancellation (for main terms, see the beginning of **2**), and the other is appearance of fully erroneous terms. The S-polynomial construction and the M-reduction can be formulated by matrices with entries of the numerical coefficients of polynomials concerned, as will be seen in **3**. As is well known, matrix elimination often causes very large cancellations. The same is true in the Gröbner base computation; in the subtraction of two polynomials, all of their main terms often cancel one another, causing large 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 the leading term, subsequent computation will be ruined completely.

In [11], the authors classified the term cancellation into two types, cancellation due to *self-reduction* and *intrinsic cancellation*. The 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 the self-reduction are removable, as we will explain below. The intrinsic cancellations are similar to numerical cancellations which occur in ill-conditioned matrices; see Example 1 in **2**. We want to know the amount of intrinsic cancellation. One reason is that the accuracy of floating-point Gröbner base is decreased by the amount of intrinsic cancellation. Another reason is that, as was shown in [11], it seems to be crucial for constructing *approximate Gröbner base*.

In [11], the authors proposed a method to overcome the instability of computation. As for the cancellation due to the self-reduction, they 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*. As for fully erroneous terms, they remove such terms by representing numeric coefficients by "effective floating-point numbers (*efloats*)"; we explain the efloat in **4**.

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 the 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 removing the errors due to the 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, the self-reduction is not avoided but we will show that it does not damage the accuracy of the Gröbner base computed. Furthermore, we propose a method to estimate the amount of intrinsic cancellation, by subtracting terms which will be canceled by the self-reduction as far as possible.

In **2**, we revisit the self-reduction and point out various kinds of self-reductions. In **3**, we analyze the main-term cancellation theoretically for the self-reductions pointed out in **2**. In **4**, we explain big-efloats and show that the self-reduction causes no problem in big-efloat computation, and we propose a new practical method. In **5**, we describe details of the implementation and show an example for estimating the intrinsic cancellation.

2 Instability due to the 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 $\|F\|$; in this paper, we employ the infinity norm, i.e., the maximum of the absolute values of the numerical coefficients of F. For notions on Gröbner base, we follow [3]. The 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 M-reduction of $\mathrm{lt}(F)$ by G, respectively; $\mathrm{Lred}(F, G)$ is often expressed as $F \xrightarrow{G} \tilde{F}$. By $F \xrightarrow{G}\!\!\!\!\rightarrow \tilde{F}$ we denote successive M-reductions of F by G so that $\mathrm{lt}(\tilde{F})$ is no more M-reducible by G.

We first explain the intrinsic cancellation by an example.

Example 1 Simple example which causes the intrinsic cancellation.

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

We convert P_1, P_2, P_3 into erroneous polynomials by converting their coefficients into double precision floating-point numbers, and compute a Gröbner base with 30-digit floating-point numbers, obtaining the following unreduced Gröbner base (underlined figures are correct).

$$\left\{\begin{array}{rcl} & & P_1,\ P_2,\ P_3 \ \text{ are unchanged,} \\ P_6 &=& y^2z^2 - \underline{2.995436947732552644538319700370}\,xy^2 \\ & & \quad - \underline{1.002078216512374825767495109674}0\,y^3 \\ & & \quad + \underline{1.998325469173724514019288562156}0\,xy + \bullet\bullet\bullet\,, \\ P_7 &=& xz^2 - \underline{1.764316342370426661429391997320}\mathrm{e}{-3}\,yz^2 \\ & & \quad - \underline{9.947232450186805419457332443380}\mathrm{e}{-1}\,xy \\ & & \quad + \underline{1.767982973726193638564792753148}0\mathrm{e}{-3}\,y^2 + \bullet\bullet\bullet\,. \end{array}\right.$$

We see that the accuracy has been decreased by $O(10^4)$, which is the same in the computation with double precision floating-point numbers. Later, we will see that the errors due to the self-reduction will disappear if we increase the precision. □

We next explain the self-reduction. In the following, we use notations $F \approx G$ 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 te $\|G\| < 0.2 \times \|F\|$.) We call a term T of F a *main term* if $\|T\| = O(\|F\|)$. Let F_1 and F_2 be *normal* polynomials, i.e., $|\mathrm{lc}(F_i)| = O(\|\mathrm{rt}(F_i)\|)$ $(i = 1, 2)$, and let G be a polynomial with small leading term, $\|\mathrm{lc}(G)\| \ll \|G\|$. Suppose that F_1 and F_2 are M-reduced by G as

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

Then, so long as $|\mathrm{lc}(F_1)|/\|F_1\| \gg |\mathrm{lc}(G)|/\|G\|$, we have

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

where M_1 and M_2 are monomials. In this case, we call $\tilde{F}_1$ and $\tilde{F}_2$ *clones* of G and represent as $\tilde{F}_i = \mathrm{clone}(G)$ $(i = 1, 2)$. Next, we assume that

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

We consider what happens in $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$; 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 $\mathrm{lt}(\tilde{F}_2) \mid \mathrm{lt}(\tilde{F}_1)$ and $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2) = -\mathrm{Lred}(\tilde{F}_2, \tilde{F}_1)$ if $\mathrm{lt}(\tilde{F}_1) \mid \mathrm{lt}(\tilde{F}_2)$. 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. With condition (2.3), we have $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2) \approx \tilde{M}_1 M_1\,\mathrm{rt}(G) - \tilde{M}_2 M_2\,\mathrm{rt}(G)$ and condition (2.4) tells us $\|\tilde{M}_1 M_1\,\mathrm{rt}(G) - \tilde{M}_2 M_2\,\mathrm{rt}(G)\| \ll \|\tilde{M}_1 M_1\,\mathrm{rt}(G)\|$. This means that all the main terms of $\tilde{M}_1 M_1\,\mathrm{rt}(G)$ and $\tilde{M}_2 M_2\,\mathrm{rt}(G)$ cancel each other; the cancellation is exact if

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

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 cancellation of all the main terms in clones was called "self-reduction" in [11]. We see that the self-reduction is caused by polynomials with small leading terms.

Definition 1 (likeness of clone) *Let F and G be polynomials of norm 1. Let $\tilde{F}$ be a clone of G: $\tilde{F} = \mathrm{clone}(G)$. We call $\|\tilde{F}\|/\|\mathrm{rt}(G)\|$* likeness *of the clone.*

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 $\mathrm{Lred}(F_2, G)$. If T_2 is contained in leading terms of both $\tilde{F}_1$ and $\tilde{F}_2$ then $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$ does not cause the self-reduction; both the large leading terms of $\tilde{F}_1$ and $\tilde{F}_2$ cancel each other. The self-reduction occurs only when $|\mathrm{lc}(\tilde{F}_1)|/\|\tilde{F}_1\| \approx |\mathrm{lc}(\tilde{F}_2)|/\|\tilde{F}_2\|$ and $\mathrm{lt}(\tilde{F}_2)\tilde{T}_1 = \mathrm{lt}(\tilde{F}_1)\tilde{T}_2$ up to a constant, where $\tilde{T}_i$ is the large term in $\tilde{F}_i$ $(i = 1, 2)$, which is extremely rare to occur. We must notice, however, that G generates a polynomial with one large term. If the large term is the leading term then it may cause the self-reduction, as we will explain below. Even if the large term is not the leading term, subsequent M-reductions may generate a polynomial with large leading term.

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

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

Then, we have $(i = 1, 2)$

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

where M_1 and M_2 are power products. Therefore, $\mathrm{Lred}(F_i, G)$ is a clone of G, and the self-reduction may occur in $\mathrm{Spol}(\mathrm{Lred}(F_1, G), \mathrm{Lred}(F_2, G))$. Note that the self-reduction requires two polynomials with large leading terms. Therefore, the self-reduction by polynomials with large leading terms is less frequent than that by polynomials with small leading terms. Note further that the M-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 $(|\mathrm{lc}(F)|/\|\mathrm{rt}(F)\|) \cdot (\|G\|/|\mathrm{lc}(G)|)$.

We have a more complicated case of self-reduction which seldom occurs. Let F_1 and F_2 be normal polynomials as above, and let F_1 and F_2 be M-reduced, respectively, by G_1 and G_2 which are polynomials with small leading terms: $F_i \xrightarrow{G_i} \tilde{F}_i = F_i - M_i G_i$ $(i = 1, 2)$. Then, we have $\tilde{F}_1 \approx M_1 \,\mathrm{rt}(G_1)$ and $\tilde{F}_2 \approx M_2 \,\mathrm{rt}(G_2)$. Consider $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2) \stackrel{\mathrm{def}}{=} \tilde{M}_1 \tilde{F}_1 - \tilde{M}_2 \tilde{F}_2$, where $\tilde{M}_1$ and $\tilde{M}_2$ are monomials. We assume that the following relations hold,

$$\mathrm{lt}(\tilde{F}_i) \succ M_i \,\mathrm{rt}(G_i) \ (i = 1, 2), \quad \mathrm{rt}(\tilde{M}_1 \tilde{F}_1) \succ \mathrm{rt}(\tilde{M}_2 \tilde{F}_2) \tag{2.8}$$

and that we have polynomials $\tilde{G}_1 \approx N_1 \,\mathrm{rt}(G_1)$ and $\tilde{G}_2 \approx N_2 \,\mathrm{rt}(G_2)$, with N_1 and N_2 monomials, satisfying $\mathrm{lt}(\tilde{G}_i) \mid \mathrm{lt}(\tilde{M}_i \,\mathrm{rt}(\tilde{F}_i))$ $(i = 1, 2)$. By assumption, $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$ contains terms $\tilde{M}_1 M_1 \,\mathrm{rt}(G_1)$ and $\tilde{M}_2 M_2 \,\mathrm{rt}(G_2)$, hence there is a possibility that the main terms of $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$ are canceled by successive M-reductions by $\tilde{G}_1$ and $\tilde{G}_2$. We call this self-reduction *paired self-reduction*. The paired self-reduction requires additional rather severe conditions: the M-reduction by $\tilde{G}_1$ does not change the leading term of $\tilde{M}_2 \tilde{F}_2$ and that $\mathrm{lt}(\tilde{M}_2 \tilde{F}_2)$ is the leading term of $\mathrm{Lred}(\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2), \tilde{G}_1)$.

The polynomial F may be M-reduced by $G_1, \ldots, G_m$ successively, where $G_1, \ldots, G_m$ are polynomials with small leading terms: $F \xrightarrow{G_1} \cdots \xrightarrow{G_m} \tilde{F}$. In this case, we call $\tilde{F}$ a *multiple clone*, and represent it as $\mathrm{clone}(G_1, \ldots, G_m)$. We will encounter double clones just below.

Example 2 Simple system causing large errors (an example shown in [11]).

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

We compute a Gröbner base w.r.t. the total-degree order, with double precision floating-point numbers, just as we compute a Gröbner base over $\mathbf{Q}$. We show about two-thirds of the whole

steps.

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

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

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

We explain why so large cancellations occur in Example 2. The $\mathrm{clone}(P_1,P_4)$ in the third line is a double clone generated by single M-reductions by P_1 and P_4, and so is the double clone in the fifth line. Following Theorem 1 in the next section, one may think that the amount of cancellation caused by the 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 of this superficial discrepancy is that, before the M-reduction by P_1, the polynomial concerned has been M-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 [11] shows this symbolically.

3 Analysis of self-reductions given in 2

In [11], we analyzed only the typical self-reduction by single clones. In this section, we analyze self-reductions given in **2**, in particular, the self-reduction by multiple clones.

Following Collins [2], we introduce *associated polynomial*. Let $P_i = c_{i1}T_1 + \cdots + c_{im}T_m$ $(i = 1,\ldots,n)$ be polynomials where $T_1,\ldots,T_m$ are power products, and $M = (c_{ij})$ be an $n \times m$

matrix, $n < m$. The polynomial associated with M, which we denote by $\mathrm{assP}(M)$, is defined as follows.

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

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 are power products satisfying $S_1 \succ S_2 \succ \cdots \succ S_n$ and $S'_1 \succ S'_2 \succ \cdots \succ S'_n$, $S_i = SS'_i$ $(1 \leq i \leq m)$ for some power product S and $f_1f'_1 \neq 0$ (some of f_i or f'_i 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 are power products, $S_i = TT_i$ and $S'_i = T'T'_i$ $(1 \leq i \leq m)$ for some power products T and T', and $g_1g'_1 \neq 0$ (some of g_i or g'_i may be 0). We consider the case that both F and F' are k times M-reduced by G and then k' times M-reduced 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 following lemma is well known; we can easily prove it by mathematical inductions on k and k' (cf. [2]).

Lemma 1 (well known) *Let F, G and G' be defined as above. Suppose F is k times M-reduced by G then k' times M-reduced by G' (only the leading terms are M-reduced), then the resulting polynomial $\tilde{F}$ can be expressed as (we discard a constant 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 added suitably by zero-coefficient terms so that the elements in each column in the above matrix correspond to the same monomial.

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 $(f'_1f'_2 \cdots f'_n \cdots)$). 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 are k and k', respectively.

Proof The coefficient of $S_{k+k'+i}$ $(i \geq 2)$ term is

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

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

The Sylvester identity allows us to factor the above expression as

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

This proves the theorem. □

Remark 1 We may define the self-reduction to be the main-term cancellation which occurs in the computation from (3.4) to (3.5). A possible main-term cancellation which occurs in the computation of the right determinant in (3.5) is the intrinsic cancellation. □

Remark 2 Consider the case that F is k_1 times M-reduced by G then k'_1 times M-reduced by G' and F' is k_2 times M-reduced by G then k'_2 times M-reduced by G'. If $k_1 > k_2$, for example, then we put $k = k_2$ and treat $k_1 - k_2$ times M-reduction 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'. □

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 do 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 $\mathrm{lc}(\tilde{F}')\tilde{F} - \mathrm{lc}(\tilde{F})S\tilde{F}'$.

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_1 and F_2 be as in* (3.6)*. Let reducers G and G' be polynomials with coefficients such that*

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

Claim1: 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. 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 such $\widetilde{D_{1i}}$ such that $f'_{k+k'+i} \neq 0$, we see that $\widetilde{D_{1i}} = O(1)$. Then, determinants in (3.5) lead us to Claim 1. The determinants also leads 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 the self-reduction caused by polynomials with large leading terms and the 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_1 and F_2 be polynomials with large leading term such that $|\mathrm{lc}(F_i)| \gg \|\mathrm{rt}(F_i)\|$ $(i = 1, 2)$, and let G be a normal polynomial. Put $\tilde{F}_i = \mathrm{Lred}(F_i, G)$ $(i = 1, 2)$. Then, in the computation of $\mathrm{Spol}(\tilde{F}_1, \tilde{F}_2)$, there occurs main-term cancellation of magnitude $\min(|\mathrm{lc}(F_1)|/\|\mathrm{rt}(F_1)\|, |\mathrm{lc}(F_2)|/\|\mathrm{rt}(F_2)\|)$.*

We next consider the paired self-reduction. We put $F_1 = F, F_2 = F', G_1 = G$ and $G_2 = G'$. For simplicity, we consider the case of $\tilde{F} = \mathrm{Lred}(F, G)$ and $\tilde{F}' = \mathrm{Lred}(F', G')$; the case of multiple M-reductions can be treated similarly. The relations in (2.8) tell us that $\tilde{F}$ and $\tilde{F}'$ can be expressed as

$$\tilde{F} = \mathrm{assP}\begin{pmatrix} f_1 & \hat{f}_1 & f_2 & \cdots \\ g_1 & 0 & g_2 & \cdots \end{pmatrix}, \qquad \tilde{F}' = \mathrm{assP}\begin{pmatrix} f'_1 & \hat{f}'_1 & f'_2 & \cdots \\ g'_1 & 0 & g'_2 & \cdots \end{pmatrix},$$

and we can express $\mathrm{Spol}(\tilde{F}, \tilde{F}') = \tilde{M}\tilde{F} - \tilde{M}'\tilde{F}' = \mathrm{rt}(\tilde{M}\tilde{F}) - \mathrm{rt}(\tilde{M}'\tilde{F}')$ as

$$\tilde{M} \cdot \mathrm{assP}\begin{pmatrix} f_1 & f_2 & f_3 & \cdots \\ g_1 & g_2 & g_3 & \cdots \end{pmatrix} - \tilde{M}' \cdot \mathrm{assP}\begin{pmatrix} f'_1 & f'_2 & f'_3 & \cdots \\ g'_1 & g'_2 & g'_3 & \cdots \end{pmatrix}.$$

The existence of $\tilde{G}$ and $\tilde{G}'$ satisfying $\tilde{G} \approx N\,\mathrm{rt}(G)$ and $\tilde{G}' \approx N'\mathrm{rt}(G')$, respectively, means that $\tilde{G}$ and $\tilde{G}'$ are obtained by M-reducing H and H', say, by G and G', respectively: $\tilde{G} = \mathrm{Lred}(H, G)$, $\tilde{G}' = \mathrm{Lred}(H', G')$. Hence, we can express $\tilde{G}$ and $\tilde{G}'$ as

$$\tilde{G} = \mathrm{assP}\begin{pmatrix} h_1 & h_2 & h_3 & \cdots \\ g_1 & g_2 & g_3 & \cdots \end{pmatrix}, \qquad \tilde{G}' = \mathrm{assP}\begin{pmatrix} h'_1 & h'_2 & h'_3 & \cdots \\ g'_1 & g'_2 & g'_3 & \cdots \end{pmatrix}.$$

Furthermore, occurrence of the paired self-reduction means that $\mathrm{rt}(\tilde{M}\tilde{F})$ and $\mathrm{rt}(\tilde{M}'\tilde{F}')$ are M-reducible by $\tilde{G}$ and $\tilde{G}'$, respectively. Therefore, applying Theorem 1, with $k = k' = 1$, to $\mathrm{Lred}(\mathrm{rt}(\tilde{M}\tilde{F}), \tilde{G})$ and $\mathrm{Lred}(\mathrm{rt}(\tilde{M}'\tilde{F}'), \tilde{G}')$ separately, we obtain the following corollary.

Corollary 3 *Let F, F', H and H' be normal polynomials and G and G' be polynomials with small leading terms, and let $\tilde{G} = \mathrm{Lred}(H, G)$ and $\tilde{G}' = \mathrm{Lred}(H', G')$. If the paired self-reduction occurs on $\mathrm{Spol}(\tilde{F}, \tilde{F}')$ by $\tilde{G}$ and $\tilde{G}'$, then the paired self-reduction causes the main-term cancellation of magnitude* $\min(\|F\|/|\mathrm{lc}(G)|, \|H\|/|\mathrm{lc}(G)|, \|F'\|/|\mathrm{lc}(G')|, \|H'\|/|\mathrm{lc}(G')|)$.

4 New method of stabilization

We propose a new method of stabilization. The method does not introduce any symbol but utilizes big-efloats, hence it is practical and much more efficient than the previous one.

First, we explain the efloats briefly. The efloat was proposed by the present authors in 1997 [6] 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.

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

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

The efloats allow us not only to estimate total amount of cancellation occurred on each coefficient but also to remove the fully erroneous terms as follows. Let ϵ_{m} be the *machine epsilon* of the floating-point numbers (the smallest mantissa of the floating-point numbers). We set the error-part of each efloat coefficient to about $5\epsilon_{\mathrm{m}}\times$ value-part (for the big-efloat, we set the error-part larger). In our algebra system named GAL, the efloat $\#\mathrm{E}[f, e]$ with $|f| < e$ is automatically set to 0. Therefore, GAL sets fully erroneous terms to 0, unless the rounding errors accumulate to $5\epsilon_{\mathrm{m}}$ or more, which is extremely rare in practice.

The big-efloat is expressed as $\#\mathrm{BE}[f, e]$, where f is a multiple precision floating-point number, and it is processed by the same arithmetic as efloat. We denote the smallest mantissa of big-efloats by ϵ_{M}. We consider that the input coefficients are inexact in general; let the relative error of a coefficient be ϵ. In the floating-point Gröbner base computation, we should assume that ϵ is not less than ϵ_{m}: $\epsilon \geq \epsilon_{\mathrm{m}} \gg \epsilon_{\mathrm{M}}$. We will convert each coefficient of the input

polynomials into a big-efloat. We say that a big-efloat is of accuracy $1/\epsilon$ if it contains an error of relative magnitude ϵ. Then, one may think that cancellations of main terms by self-reduction will decrease the accuracy of big-efloat coefficients. Surprisingly, in all the cases we have investigated in **3**, the cancellation due to the self-reduction does not decrease the accuracy, as we will show just below.

Theorem 2 *So long as the self-reductions treated in* **3** *are concerned, the main-term cancellation due to the self-reduction ruins only tail figures of the coefficients concerned; if the amount of cancellation is* $O(10^{\kappa})$ *then tail* κ *figures of coefficients concerned are ruined.*

Proof We note that, although the big-efloats in our case contain relative errors which are much larger than ϵ_{M}, *the errors are represented correctly within the precision and treated as definite numbers given initially.* Furthermore, Theorem 1 and Corollaries $1 \sim 3$ imply that the main terms cancel exactly in the self-reduction, hence the self-reduction ruins only tail figures of the coefficients concerned. □

Example 3 Check Theorem 2 by the system in Example 2.
We convert the coefficients into double precision floating-point numbers, and compute the Gröbner base with big-efloats of 30 decimal precision. For reference, we show the initial polynomials; note that the rounding errors appear at the 17th decimal places.

$$\left\{\begin{array}{rcl} P_1 &=& +\ \#\mathrm{BE}[3.3333333333333330\mathrm{e}{-2},\ 2.0\mathrm{e}{-28}]\, x^3 + x^2 y \\ && +\ \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, y^2, \\ P_2 &=& +\ \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, x^2 y^2 - x y^2, \\ && -\ \#\mathrm{BE}[3.3333333333333310\mathrm{e}{-1},\ 3.2\mathrm{e}{-27}]\, xy \\ P_3 &=& +\ \#\mathrm{BE}[5.0000000000000000\mathrm{e}{-2},\ 3.9\mathrm{e}{-28}]\, y^3 + x^2. \end{array}\right.$$

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

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

We obtain the following unreduced Gröbner base (underlines show correct figures).

$$\left\{\begin{array}{rcl} P_2'' &=& y^2, \\ P_4' &=& xy + \#\mathrm{BE}[\underline{8.4402255045219586762893116546}00\mathrm{e}{-2},\ 3.3\mathrm{e}{-21}]\, y^2, \\ P_5 &=& x^2 + \#\mathrm{BE}[\underline{7.1484968974627}0700636549331894 0,\ 4.2\mathrm{e}{-19}]\, xy \\ && +\ \#\mathrm{BE}[\underline{5.7371613952464572}2574204458941 0\mathrm{e}{-1},\ 2.6\mathrm{e}{-20}]\, y^2. \end{array}\right.$$

Although large cancellations have occurred, the accuracy decrease is only a little. □

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 each input polynomial into big-efloats of a suitably determined initial precision, and compute the Gröbner base as usual.

Device 2: Monitor the error-parts of big-efloat coefficients during the computation, and if the relative error-parts become too large then increase the precision of big-efloats and retry the computation.

Device 3: Monitor the clone generation of likeness greater than 5.0, say. If the self-reduction occurs in the subtraction $\tilde{F}_1 - \tilde{F}_2$, where $\tilde{F}_1 = \text{clone}(G)$ and $\tilde{F}_2 = \text{clone}(G)$, say, then we subtract G from both $\tilde{F}_1$ and $\tilde{F}_2$ as $\tilde{F}'_1 := \tilde{F}_1 - G$ and $\tilde{F}'_2 := \tilde{F}_2 - G$, and compute $\tilde{F}'_1 - \tilde{F}'_2$. We call this operation *reducer subtraction*. Regard the possible cancellation occurring in the subtraction $\tilde{F}'_1 - \tilde{F}'_2$ as the intrinsic cancellation.

With Devices 1 and 2, we can remove the cancellation errors due to the self-reduction completely; the number 5.0 for specifying the clone in Device 3 is irrelevant to this removal. As for the intrinsic cancellation, authors of [1] and [11] 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 **5** for implementation), hence our method is practical. However, it should be mentioned that Device 3 will miss to remove small amounts of cancellations due to small self-reductions, because we neglect the clones of likeness ≤ 5. Therefore, the above method will over-estimate the amount of intrinsic cancellation.

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 ($i \in \mathbf{N}$), and the numbering is not changed if the polynomial is M-reduced; if F_i is M-reduced to 0 then F_i is removed from the memory. Suppose a polynomial F_i is M-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 will be changed during the computation. Let the M-reduction be $F_i := F_i - c_j T_j G_j$, where $c_j \in \mathbf{C}$ and T_j is a power product. F_i is usually M-reduced by G_j many times, say $F_i := (F_i - c_j T_j G_j) - c'_j T'_j G'_j$. The multiplier c_j changes from the M-reduction to M-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 triplet $\langle h_j,\, c_j,\, T_j G_j \rangle$ which we call *clone-triplet*.
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 $(\cdots\ \langle j',\, c'_{j'},\, T'_{j'} G_{j'} \rangle\ \langle j,\, c_j,\, T_j G_j \rangle)$.

5.2 Criteria for clone generation and self-reduction

In our current program, we neglect the paired self-reduction because it occurs very rarely while its implementation is messy. Hence, the following criteria are easy ones and not complete. Suppose the clone-triplet lists for polynomials F_1 and F_2 are

$$\begin{array}{ll} F_1: & (\langle j_1,\, c_1,\, T_1 G_1 \rangle\ \langle j'_1,\, c'_1,\, T'_1 G'_1 \rangle\ \cdots), \\ F_2: & (\langle j_2,\, c_2,\, T_2 G_2 \rangle\ \langle j'_2,\, c'_2,\, T'_2 G'_2 \rangle\ \cdots). \end{array} \tag{5.1}$$

As we have noticed in **2**, we consider only $\mathrm{Spol}(F_1, F_2)$ below. If $T_1G_1 \neq T_2G_2$ then $\mathrm{Spol}(F_1, F_2)$ is not the self-reduction. Does the self-reduction occur always if $T_1G_1 = T_2G_2$? The answer is not always YES; the answer is YES only when the relations in (2.4) hold. Therefore, we judge the clone generation by the following criteria which are described for the case $\tilde{F} := F - cTG$, where T is a power product.

Criterion C1 If $|\mathrm{lc}(G)| < \|\mathrm{rt}(G)\|/5$ then G is a polynomial with small leading term.
If $|\mathrm{lc}(G)| > 5\,\|\mathrm{rt}(G)\|$ then G is a polynomial with large leading term.

Criterion C2 For the case of small leading term:
If $\mathrm{lt}(\mathrm{rt}(TG)) \succ \mathrm{lt}(\mathrm{rt}(F))$ and $5\,|\mathrm{lc}(F)/\mathrm{lc}(G)| < \|\mathrm{rt}(F)\|/\|\mathrm{rt}(G)\|$ then $\tilde{F} = \mathrm{clone}(G)$.
If $\mathrm{lt}(\mathrm{rt}(TG)) \propto \mathrm{lt}(\mathrm{rt}(F))$ and $5\,|\mathrm{lc}(F)/\mathrm{lc}(G)| < \|\mathrm{rt}(F)\|/\|\mathrm{rt}(G)\|$ and
$5\,|\mathrm{lc}(\mathrm{rt}(F))| < |\mathrm{lc}(\mathrm{rt}(TG))|$ then $\tilde{F} = \mathrm{clone}(G)$.

Criterion C3 For the case of large leading term:
If $\mathrm{lt}(\mathrm{rt}(TG)) \succ \mathrm{lt}(\mathrm{rt}(F))$ and $|\mathrm{lc}(F)/\mathrm{lc}(G)| < 5\,\|\mathrm{rt}(F)\|/\|\mathrm{rt}(G)\|$ then $\tilde{F} = \mathrm{clone}(F)$.
If $\mathrm{lt}(\mathrm{rt}(TG)) \propto \mathrm{lt}(\mathrm{rt}(F))$ and $|\mathrm{lc}(F)/\mathrm{lc}(G)| < 5\,\|\mathrm{rt}(F)\|/\|\mathrm{rt}(G)\|$ and
$|\mathrm{lc}(\mathrm{rt}(F))| < 5\,|\mathrm{lc}(\mathrm{rt}(TG))|$ then $\tilde{F} = \mathrm{clone}(F)$.

With the above criteria, we can judge the self-reduction by the following criterion.

Criterion SR If $j_1 = j_2$ and $T_1G_1 = T_2G_2$ then $\mathrm{Spol}(F_1, F_2)$ causes the self-reduction.

Note that, if $(j_1' = j_2'$ and $T_1'G_1' = T_2'G_2')$ in addition to $(j_1 = j_2$ and $T_1G_1 = T_2G_2)$ then $\mathrm{Spol}(F_1, F_2)$ causes the self-reduction by double clones, and so on.

5.3 Reducer subtraction

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 M-reduction (and S-polynomial generation): $\tilde{F}_i := F_i - c_jT_jG_j \Rightarrow \tilde{F}_i := \tilde{F}_i/\mathrm{lc}(\tilde{F}_i)$. By this normalization, the multiplier c_j must be modified accordingly: we change all the multipliers in the clone-triplet list for F_i as $\langle j,\, c_j,\, T_jG_j\rangle \Rightarrow \langle j,\, c_j/\mathrm{lc}(\tilde{F}_i),\, T_jG_j\rangle$.

If conditions in (2.5) hold then the reducer subtraction is easy: with the notations in the previous subsection, F_1 and F_2 are M-reduced as $F_1 := F_1 - c_1T_1G_1$ and $F_2 := F_2 - c_2T_1G_1$, and we have $c_1 = c_2$, hence we subtract $c_1T_1G_1$ from both F_1 and F_2. If conditions in (2.5) do not hold then we have $c_1 \neq c_2$ although $c_1 \approx c_2$. In this case, we compute c as

$$c = \begin{cases} c_1 & \text{if } |c_1| \leq |c_2|, \\ c_2 & \text{if } |c_1| > |c_2|, \end{cases} \tag{5.2}$$

and subtract G_1 from F_1 and F_2 as $F_1 := F_1 - cT_1G_1$ and $F_2 := F_2 - cT_1G_1$.

5.4 Estimating the intrinsic cancellation

The actual term cancellations are the sum of cancellations due to the self-reductions and the intrinsic cancellations. Therefore, if we remove all the cancellations due to the self-reductions, then the rest cancellations must be intrinsic cancellations. Below, we show how the intrinsic cancellation is estimated 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 the self-reduction by double clones occurs and we encounter term cancellation of $O(10^{10})$. We will see that the reducers are subtracted successfully and the intrinsic cancellation is estimated adequately.

Example 4 Intrinsic cancellation in the fifth line 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').

$$\begin{aligned}
Q_1 &= + \#\mathrm{BE}[\underline{1.1152418136}78930955845340517 1\mathrm{e}{-1},\ 8.6\mathrm{e}{-28}]\, y^3 \\
&\quad + \#\mathrm{BE}[\underline{3.3457253}711642806415804801040\mathrm{e}{-1},\ 3.7\mathrm{e}{-27}]\, xy \\
&\quad - \#\mathrm{BE}[\underline{4.16135}062896641687828404499 50\mathrm{e}{-5},\ 1.1\mathrm{e}{-28}]\, y^2, \\
Q_2 &= - \#\mathrm{BE}[\underline{1.1152418132}002535431698179200\mathrm{e}{-1},\ 8.6\mathrm{e}{-28}]\, y^3 \\
&\quad - \#\mathrm{BE}[\underline{3.3457254}396007606295094537600\mathrm{e}{-1},\ 3.7\mathrm{e}{-27}]\, xy \\
&\quad + \#\mathrm{BE}[\underline{4.1612}957039749613089747630908\mathrm{e}{-5},\ 1.1\mathrm{e}{-28}]\, y^2.
\end{aligned}$$

Subtracting a multiple of P_4 from Q_1 and Q_2, we obtain $Q_1 \to Q_1'$ and $Q_2 \to Q_2'$:

$$\begin{aligned}
Q_1' &= + \#\mathrm{BE}[\underline{2.3}290837408651847149108379154\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\
&\quad - \#\mathrm{BE}[\underline{1.1194031}410170599640717774130\mathrm{e}{-2},\ 1.1\mathrm{e}{-28}]\, y^2, \\
Q_2' &= - \#\mathrm{BE}[\underline{2.2}812159995976324552013022754\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\
&\quad - \#\mathrm{BE}[6.843647998792897365603906826 4\mathrm{e}{-9},\ 3.7\mathrm{e}{-27}]\, xy \\
&\quad + \#\mathrm{BE}[\underline{1.1194030}860920685085024681310\mathrm{e}{-2},\ 1.1\mathrm{e}{-28}]\, y^2.
\end{aligned}$$

Subtracting a multiple of P_1 from Q_1' and Q_2', we obtain $Q_1' \to Q_1''$ and $Q_2' \to Q_2''$:

$$\begin{aligned}
S_1'' &= + \#\mathrm{BE}[\underline{2.3}290837408651847149108379154\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3, \\
S_2'' &= - \#\mathrm{BE}[\underline{2.2}812159995976324552013022754\mathrm{e}{-9},\ 8.6\mathrm{e}{-28}]\, y^3 \\
&\quad - \#\mathrm{BE}[6.843647998792897365603906826 4\mathrm{e}{-9},\ 3.7\mathrm{e}{-27}]\, xy \\
&\quad - \#\mathrm{BE}[5.492499145556930928190424214 8\mathrm{e}{-10},\ 1.1\mathrm{e}{-28}]\, y^2.
\end{aligned}$$

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

6 Concluding remarks

We showed that, restricting the M-reductions to leading-term reductions, we are able to describe local steps of Gröbner base computation in terms of matrices and analyze the self-reduction and intrinsic cancellation in terms of determinants (Theorem 1). Furthermore, we showed that the main-term cancellation due to the self-reduction causes no problem if we utilize big-efloats, so long as the self-reductions investigated in **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 the 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] M. Bodrato and A. Zanoni. Intervals, syzygies, numerical Gröbner bases: a mixed study. *Proceedings of CASC2006 (Computer Algebra in Scientific Computing)*; Springer-Verlag LNCS 4194, 64-76, 2006.

[2] J.E. Collins. Subresultant and reduced polynomial remainder sequence. *J. ACM* **14** (1967), 128-142.

[3] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag New York, 1997.

[4] E. Fortuna, P. Gianni and B. Trager. Degree reduction under specialization. *J. Pure Appl. Algebra* **164** (2001), 153-164.

[5] L. Gonzalez-Vega, C. Traverso and A. Zanoni. Hilbert stratification and parametric Gröbner bases. *Proceedings of CASC2005 (Computer Algebra in Scientific Computing)*; Springer-Verlag LNCS 3718, 220-235, 2005.

[6] F. Kako and T. Sasaki. Proposal of "effective" floating-point number. Preprint of Univ. Tsukuba, May 1997 (unpublished).

[7] A. Kondratyev, H.J. Stetter and S. Winkler. Numerical computation of Gröbner bases. *Proceedings of CASC2004 (Computer Algebra in Scientific Computing)*, 295-306, St. Petersburg, Russia, 2004.

[8] B. Mourrain. Pythagore's dilemma, symbolic-numeric computation, and the border basis method. *Symbolic-Numeric Computations (Trends in Mathematics)*, 223-243, Birkhäuser Verlag, 2007.

[9] K. Shirayanagi. An algorithm to compute floating-point Gröbner bases. *Mathematical Computation with Maple V. Ideas and Applications*, Birkhäuser, 95-106, 1993.

[10] K. Shirayanagi. Floating point Gröbner bases. *Mathematics and Computers in Simulation* **42** (1996), 509-528.

[11] T. Sasaki and F. Kako. Computing floating-point Gröbner base stably. *Proceedings of SNC2007 (Symbolic Numeric Computation)*, 180-189, London, Canada, 2007.

[12] K. Shirayanagi and M. Sweedler. Remarks on automatic algorithm stabilization. *J. Symb. Comput.*, **26** (1998), 761-765.

[13] H.J. Stetter. Stabilization of polynomial systems solving with Gröbner bases. *Proceedings of ISSAC'97 (Intern'l Symposium on Symbolic and Algebraic Computation)*, 117-124, ACM Press, 1997.

[14] H.J. Stetter. *Numerical Polynomial Algebra*. SIAM Publ., Philadelphia, 2004.

[15] H.J. Stetter. Approximate Gröbner bases – an impossible concept? *Proceedings of SNC2005 (Symbolic-Numeric Computation)*, 235-236, Xi'an, China, 2005.

[16] C. Traverso. Syzygies, and the stabilization of numerical Buchberger algorithm. *Proceedings of LMCS2002 (Logic, Mathematics and Computer Science)*, 244-255, RISC-Linz, Austria, 2002.

[17] C. Traverso and A. Zanoni. Numerical stability and stabilization of Gröbner basis computation. *Proceedings of ISSAC2002 (Intern'l Symposium on Symbolic and Algebraic Computation)*, 262-269, ACM Press, 2002.

[18] V. Weispfenning. Gröbner bases for inexact input data. *Proceedings of CASC2003 (Computer Algebra in Scientific Computing)*, 403-411, Passau, Germany, 2003.