Communications of JSSAC (2012)
Vol. 1, pp. 39 – 66

Improving Suzuki-Sato's CGS Algorithm by Using Stability of Gröbner Bases and Basic Manipulations for Efficient Implementation

Yosuke Kurata [*]

Department of Mathematics, Kobe University

(Received 12/Jan/2010 Accepted 2/Mar/2010)

Abstract

In this paper, we propose improving Suzuki-Sato's algorithm for computing CGS. This paper consists of two parts. In the first part, using known algebraic manipulations on affine varieties, we describe a detail of basic manipulations to improve Suzuki-Sato's algorithm. In the second part, we present a new algorithm which improves Nabeshima's approach to compute a CGS. Nabeshima's approach uses Gröbner basis computations together with inequations, which involves an additional temporary variable. The approach sometimes generates time-consuming Gröbner basis computations. As a result, Nabeshima's approach is not always faster than Suzuki-Sato's original one. Our new algorithm also uses inequations without the additional variable and works like Suzuki-Sato's algorithm. So that, it is expected that the new algorithm reduces generating time-consuming Gröbner basis computations. We compare the runtime and number of segments measured by the both algorithms and find our algorithm superior in several cases.

1 Introduction

A *comprehensive Gröbner system* (CGS) for computing Gröbner bases of parametric polynomial ideals is used as a systematic tool to classify the roots of parametric polynomial equations. For example, Kanno, *et al.* [6] has explored the potential of algebraic approach for *parametric polynomial spectral factorization* (PPSF). For computing the parametric spectral factor of a parametric polynomial, a classification of the roots of parametric polynomial equations is needed. For this purpose, they devised an algebraic algorithm using CGS computation. Shinohara [18] presented three CGS based algorithms for computing PPSF and showed that the fastest one took 72 hours in computation of PPSF of a quartic parametric polynomial. So that, from a practical point of view, an efficient CGS implementation is necessary in order to solve large problems.

There are algorithms for computing CGS: Weispfenning [22, 23], Montes and Manubens [8, 10], and Suzuki and Sato [20]. Suzuki-Sato's algorithm is the fastest among them in several cases.

[*] kurata@math.kobe-u.ac.jp

Given a finite polynomial set $F \subset K[\bar{A}, \bar{X}]$, where $\bar{A} = (A_1, \ldots, A_m)$ and $\bar{X} = (X_1, \ldots, X_n)$ is parameters and variables, Suzuki-Sato's algorithm computes a CGS $\mathcal{H}$ of F with respect to $<_{\bar{X}}$ by the following algorithm.

1. Compute the reduced Göbner basis G of F with respect to $<_{\bar{A},\bar{X}}$, where $<_{\bar{A},\bar{X}}$ is a elimination term order such that $\bar{X}$ are always bigger than $\bar{A}$.

2. If $G = \{1\}$, then return $\mathcal{H}$.

3. If $G \neq \{1\}$, then collect the head coefficients ($\in K[\bar{A}]$) of polynomials in $G \setminus K[\bar{A}]$, that is $H = \{h_1, \ldots, h_l\} = \{\mathrm{HC}_{<_{\bar{X}}}(g) \in K[\bar{A}] \mid g \in G \setminus K[\bar{A}]\}$.

4. Let $h = \mathrm{LCM}(H)$, and add a segment $(F \cap K[\bar{A}], \{h\}, G)$ to $\mathcal{H}$.

5. Apply this procedure recursively to each $G \cup \{h_i\}, (1 \leq i \leq l)$.

On a computer algebra system equipped with Gröbner bases computations, it is easy to implement Suzuki-Sato's CGS algorithm. However the algorithm given above is impractical. In fact, it requires a large number of steps until the algorithm terminates because of the step 5. Suzuki-Sato's algorithm also often produces many segments $(F \cap K[\bar{A}], \{h\}, G)$ whose parameter spaces $\mathbf{V}(F \cap K[\bar{A}]) \setminus \mathbf{V}(h)$ overlap each other, so that the computational cost will swell without careful treatments of superfluous segments. Thus we need an algorithm whose number of steps becomes small, and an optimal means to obtain irredundant segments.

Suzuki and Sato [20] has referred to them. However they didn't write their details, and no one mentioned their details. In the first part of this paper, after we describe useful basic manipulations to implement CGS algorithm based on Suzuki-Sato's, we give timing data to show efficiency by our original implementation equipped with the manipulation. This part will be useful for the reader who wants to implement the CGS algorithm based on Suzuki-Sato's.

In the second part of this paper, we present an improvement of Nabeshima's method [13]. Nabeshima proposed speed-up techniques for Suzuki-Sato's algorithm in 2007. For a Gröbner basis G computed in $K[\bar{A}, \bar{X}]$, if we regard G as a Gröbner basis in the polynomial ring over the polynomial ring $(K[\bar{A}])[\bar{X}]$, the following property often holds:

$$\text{there exist } g_1, g_2 \in G \text{ such that } \mathrm{HT}_{<_{\bar{X}}}(g_1) \mid \mathrm{HT}_{<_{\bar{X}}}(g_2) \text{ and } g_1 \neq g_2. \qquad (*)$$

In Suzuki-Sato's algorithm, this property generates a lot of segments whose parameter spaces are small and unnecessary. In order to avoid this problem, Nabeshima introduced a Gröbner basis computation in which inequations ($\neq 0$) are treated. For example, in (*), if we suppose $\mathrm{HC}_{<_{\bar{X}}}(g_1) \neq 0$, we can ignore the condition of $\mathrm{HC}_{<_{\bar{X}}}(g_2)$ whether it becomes zero or not. To handle the condition $\mathrm{HC}_{<_{\bar{X}}}(g_1) \neq 0$, Nabeshima's method uses an additional temporary variable "r" as $r = 1/\mathrm{HC}_{<_{\bar{X}}}(g_1)$ and replaces g_1 with $g'_1 = \mathrm{HT}_{<_{\bar{X}}}(g_1) + r \cdot (g_1 - \mathrm{HM}_{<_{\bar{X}}}(g_1))$. Precisely speaking, Nabeshima's method is to keep adding a non-zero condition (polynomial) and computing a Gröbner basis together with the additional variable until the property (*) will not occur. On the other hand, if the property (*) will not occur, the method works the same as Suzuki-Sato's one. The author noticed that this procedure may generate time-consuming Gröbner basis computations. In fact, Suzuki-Sato's algorithm is sometimes faster than Nabeshima's one.

In order to solve this problem, we present another approach which prevents the property (*) from making a lot of unnecessary segments. This approach is developed on a generalization of property (*) and without an additional variable "r". Our approach works basically like Suzuki-Sato's algorithm. It collects the parameter conditions in which the property (*) does not occur for given Gröbner basis G in $(K[\bar{A}])[\bar{X}]$. In addition, we show that G is *stable* under the parameter conditions. Of course, though we do not forget the cost of collecting the parameter conditions, we find that our approach is more efficient than Nabeshima's one in several cases. We will show this by comparing timing data and the number of segments measured by our original implementation equipped with the basic manipulation given in the first part, and Nabeshima's one.

Our paper is organized as follows. In Section 2, we define common mathematical notations and review the original Suzuki-Sato's CGS algorithm and result of stability of an ideal, which is a base of Suzuki-Sato's algorithm. In Section 3, we describe basic manipulations for a CGS implementation based on Suzuki-Sato's. In Section 4, we present our main result. We give a improvement of Nabeshima's method, a new algorithm using the generalization of (*), and we show comparisons of timing data and the number of segments.

2 Notations, Definitions and the Original Algorithm

In this section, we describe some notations and definitions used throughout this paper.

For a polynomial f in a polynomial ring equipped with a term order $<$, $\mathrm{HT}_<(f)$, $\mathrm{HC}_<(f)$, and $\mathrm{HM}_<(f)$ denote the head term of f with respect to $<$, the coefficient of $\mathrm{HT}_<(f)$, and $\mathrm{HC}_<(f) \cdot \mathrm{HT}_<(f)$ respectively. In addition, for a subset I in the polynomial ring, $\mathrm{HT}_<(I)$ and $\mathrm{HM}_<(I)$ denote $\{\mathrm{HT}_<(f) \mid f \in I\}$ and $\{\mathrm{HM}_<(f) \mid f \in I\}$ respectively.

K and L denote fields such that L is an algebraic closure of K. $\bar{X} = \{X_1, \ldots, X_n\}$ and $\bar{A} = \{A_1, \ldots, A_m\}$ denote sets of variables such that $\bar{A} \cap \bar{X} = \emptyset$. $T(\bar{X})$, $T(\bar{A})$ and $T(\bar{A}, \bar{X})$ denote the set of terms of $\bar{X}$, $\bar{A}$ and $\bar{A} \cup \bar{X}$ respectively. $<_{\bar{A},\bar{X}}$ denotes a term order on $T(\bar{A}, \bar{X})$ such that $\bar{A} \ll \bar{X}$, that is any term in $T(\bar{X})$ is greater than any term in $T(\bar{A})$, $<_{\bar{A}}$ and $<_{\bar{X}}$ denote its restriction on $T(\bar{A})$ and $T(\bar{X})$ respectively. $\mathbb{N}$ and $\mathbb{Q}$ are the set of natural numbers and the field of rational numbers respectively.

For any $\bar{a} \in L^m$, we define the canonical specialization homomorphism $\sigma_{\bar{a}} : K[\bar{A}] \longrightarrow L$ induced by $\bar{a}$, and we naturally extend it to $\sigma_{\bar{a}} : (K[\bar{A}])[\bar{X}] \longrightarrow L[\bar{X}]$.

For any subset F of $K[\bar{A}]$, $\mathbf{V}(F)$ denotes the algebraic set defined by F, that is

$$\mathbf{V}(F) = \{\bar{a} \in L^m \mid \forall f \in F,\ f(\bar{a}) = 0\} \subset L^m.$$

Similarly, if $F = \{f_1, \ldots, f_k\}$ is a finite set, $\mathbf{V}(f_1, \ldots, f_k)$ also denotes its algebraic set. Moreover, for any algebraic set $V \subset K^m$, $\mathbf{I}(V)$ denotes the ideal in $K[\bar{A}]$ such that

$$\mathbf{I}(V) = \{f \in K[\bar{A}] \mid \forall \bar{a} \in V, f(\bar{a}) = 0\}.$$

For a polynomial $f \in K[\bar{A}]$, a finite subset $G \subset K[\bar{A}]$, and a term order $<_{\bar{A}}$, $\mathrm{NF}(f, G, <_{\bar{A}})$ denotes one of the normal forms of f modulo G with respect to $<_{\bar{A}}$. In general, $\mathrm{NF}(f, G, <_{\bar{A}})$ is not uniquely determined, however it is uniquely determined if G is a Gröbner basis. For any ideal $I \subset K[\bar{A}]$, $\sqrt{I}$ denotes the radical of I.

Definition 1 (CGS)
Let F be a subset of $K[\bar{A}, \bar{X}]$, and $S_1, \ldots, S_l, T_1, \ldots, T_l$ be finite subsets of $K[\bar{A}]$. A finite set

$$\mathcal{G} = \{(S_1, T_1, G_1), \ldots, (S_l, T_l, G_l)\}$$

of triples is called a comprehensive Gröbner system for F with respect to $<_{\bar{X}}$, if $(\mathbf{V}(S_1) \setminus \mathbf{V}(T_1)) \cup \cdots \cup (\mathbf{V}(S_l) \setminus \mathbf{V}(T_l)) = L^m$ and $\sigma_{\bar{a}}(G_i)$ is a Gröbner basis of the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[\bar{X}]$ with respect to $<_{\bar{X}}$ for all $\bar{a} \in \mathbf{V}(S_i) \setminus \mathbf{V}(T_i)$ for each $i = 1, \ldots, l$. Then, each (S_i, T_i, G_i) or $(\mathbf{V}(S_i) \setminus \mathbf{V}(T_i), G_i)$ is called a segment of $\mathcal{G}$.

For a segment (S, T, G), (S, T) or $\mathbf{V}(S) \setminus \mathbf{V}(T)$ is also called its parameter space or simply called its case.

Suzuki-Sato's CGS algorithm is developed on the results of stability of an ideal in a polynomial ring $R[\bar{X}]$ over Noetherian ring R with identity, which was introduced by Kalkbrener [5].

Definition 2 (Stability of an Ideal)
Let R and R' be Noetherian commutative rings with identity, and let π be a ring homomorphism from R to R', then an ideal I in $R[\bar{X}]$ is called stable under π and $<_{\bar{X}}$ if it satisfies

$$\langle \pi(\langle \mathrm{HM}_{<_{\bar{X}}}(I)\rangle)\rangle = \langle \mathrm{HM}_{<_{\bar{X}}}(\pi(I))\rangle.$$

Theorem 3 (Kalkbrener 1997)
Let π a ring homomorphism from a Noetherian ring R with identity to a field K, I an ideal in $R[\bar{X}]$, and $G = \{g_1, \ldots, g_r\}$ be a Gröbner basis of I with respect to $<_{\bar{X}}$. We assume that the g_is are ordered in such a way that $\pi(\mathrm{HC}(g_i)) \neq 0$ for $1 \leq i \leq r$ and $\pi(\mathrm{HC}(g_j)) = 0$ for $r + 1 \leq j \leq s$. Then the following three conditions are equivalent.

1. *I is stable under π and $<_{\bar{X}}$.*
2. *$\{\pi(g_1), \ldots, \pi(g_r)\}$ is a Gröbner basis of $\langle \pi(I)\rangle \subset K[\bar{X}]$ with respect to $<_{\bar{X}}$.*
3. *For every $j \in \{r + 1, \ldots, s\}$, the polynomial $\pi(g_j)$ is reducible to 0 modulo $\{\pi(g_1), \ldots, \pi(g_r)\}$.*

In Suzuki-Sato [20], they showed next lemma, which is an easy consequence of Theorem 3, and proposed an algorithm for computing CGS.

Lemma 4
For an ideal I in $(K[\bar{A}])[\bar{X}]$, let $G = \{g_1, \ldots, g_s\}$ be a Gröbner basis of I with respect to $<_{\bar{X}}$ such that $g_i \notin K[\bar{A}]$ for $1 \leq i \leq r$ and $g_j \in K[\bar{A}]$ for $r + 1 \leq j \leq s$. Then $\sigma_{\bar{a}}(G)$ is a Gröbner basis of $\langle \sigma_{\bar{a}}(I)\rangle$ with respect to $<_{\bar{X}}$ for any $\bar{a} \in \mathbf{V}(g_{r+1}, \ldots, g_s) \setminus \left(\mathbf{V}(\mathrm{HC}_{<_{\bar{X}}}(g_1)) \cup \cdots \cup \mathbf{V}(\mathrm{HC}_{<_{\bar{X}}}(g_r))\right)$ in L^m.

For a Gröbner basis $G \subset (K[\bar{A}])[\bar{X}]$, this lemma shows that we can determine a segment (S, T, G), where $S = \{g_{r+1}, \ldots, g_s\}$, $T = \{\mathrm{LCM}(\mathrm{HC}_{<_{\bar{X}}}(g_1), \ldots, \mathrm{HC}_{<_{\bar{X}}}(g_r))\}$, and LCM denotes the least common multiple. However we cannot know whether $\sigma_{\bar{a}}(G)$ becomes a Gröbner basis or not for $\bar{a} \in \mathbf{V}(T)$. In order to deal with $\bar{a} \in \mathbf{V}(T)$, we compute a Gröbner basis of $G \cup T \subset (K[\bar{A}])[\bar{X}]$ again. Thus we can compute a CGS of F by the following algorithm. The termination of the algorithm and more details can be found in [20].

Algorithm 5
CGS($(a_1, \ldots, a_d), F, <_{\bar{A},\bar{X}}$)

```
INPUT:  A $d$-tuple $(a_1, \ldots, a_d)$ of natural numbers, a finite set $F \subset K[\bar{A}, \bar{X}]$, and
        a term order $<_{\bar{A},\bar{X}}$.   $\cdots$   (1)
OUTPUT: A set $\mathcal{H}$ of segments $(No, S, T, G)$, where $No \in \mathbb{N}^k$, $S$, $T \subset K[\bar{A}]$, and
        $G$ is the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$.

BEGIN
    $G \leftarrow$ ReducedGB($F, <_{\bar{A},\bar{X}}$);
    $\mathcal{H} \leftarrow \left\{\left((a_1, \ldots, a_d), F \cap K[\bar{A}], G \cap K[\bar{A}], \{1\}\right)\right\}$;   $\cdots$   (2)
    IF $1 \in G$ THEN
        return $\mathcal{H}$;
        END IF
        $h \longleftarrow$ LCM$(h_1, \dots, h_l)$;        $\left(\{h_1, \dots, h_l\} = \{\mathrm{HC}_{<_{\bar{X}}}(g) \in K[\bar{A}] \mid g \in G\}\right)$
        $\mathcal{H} \leftarrow \mathcal{H} \cup \left\{\left((a_1, \dots, a_d), G \cap K[\bar{A}], \{h\}, G \setminus K[\bar{A}]\right)\right\}$;
        FOR $i = 1, \dots, l$ DO
            $\mathcal{H} \leftarrow \mathcal{H} \cup$ CGS$((a_1, \dots, a_d, i), G \cup \{h_i\}, <_{\bar{A},\bar{X}})$;
        END FOR
        return $\mathcal{H}$;
END
```

Remark 6
We have the following remarks in the algorithm CGS.

(1) *A d-tuple $(a_1, \dots, a_d)$ is a serial number of the segment which is determined at the call of* CGS*. Using this number, we can represent the "depth" or "width" of the segment (see Fig. 1). In this case, its depth is d and the segment is generated a_dth at depth d.*

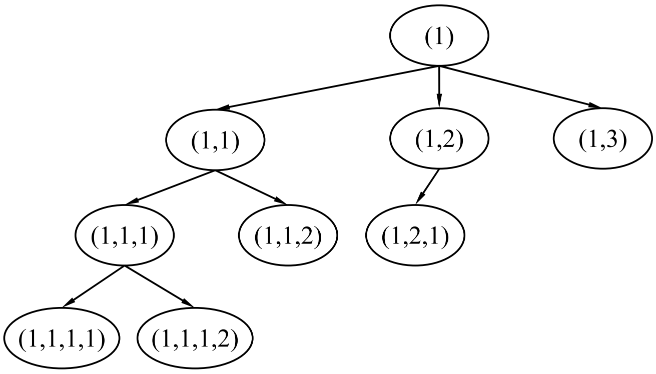

Fig. 1:

$D = \max\left\{d \mid \left((a_1, \dots, a_d), S, T, G\right) \in \mathcal{H}\right\}$ *denotes the maximum depth of CGS $\mathcal{H}$.*

(2) *In general, an input F need not be a Gröbner basis with respect to $<_{\bar{A},\bar{X}}$, so that for a Gröbner basis G of F, $\mathbf{V}(F \cap K[\bar{A}])$ may be different from $\mathbf{V}(G \cap K[\bar{A}])$. If both parameter spaces are different, then the reduced Gröbner basis of $\sigma_{\bar{a}}(F)$ is $\{1\}$ for any $\bar{a} \in \mathbf{V}(F \cap K[\bar{A}]) \setminus \mathbf{V}(G \cap K[\bar{A}])$. In Suzuki-Sato's algorithm, a Gröbner basis becomes $\{1\}$ when and only when $\mathbf{V}(F \cap K[\bar{A}]) \neq \mathbf{V}(G \cap K[\bar{A}])$.*

3 Basic Manipulations for CGS Implementation

An algorithm based on CGS often generates many superfluous segments, so that it gets into computational difficulties if we do not apply optimizations. In order to avoid these difficulties, we must reduce the useless segments, which appear during a computation, by treating their parameter spaces carefully.

In this section, we give details of basic manipulations to implement a high-speed CGS computation based on CGS.

3.1 Preliminary

In this subsection, we describe manipulations for parameter spaces. First, for a segment (S, T, G), we show how to check whether a parameter space is empty or not. To see this, it is enough to check that $\mathbf{V}(S) \subset \mathbf{V}(T)$ in L^m, namely it is enough to check that $\mathbf{I}(\mathbf{V}(S)) \supset \mathbf{I}(\mathbf{V}(T)) \iff \sqrt{\langle S \rangle} \supset T$. we can check it by using the radical membership test.

Algorithm 7
`CaseIsZero`$((S, T), <_{\bar{A}})$

INPUT: A parameter space (S, T), and a term order $<_{\bar{A}}$ on $T(\bar{A})$.
OUTPUT: "true" if $\mathbf{V}(S) \setminus \mathbf{V}(T)$ is empty, otherwise "false".

```
BEGIN
    $Y \leftarrow$ A temporary variable except $\bar{A} \cup \bar{X}$;
    FOR EACH $f \in T$ DO
        $B \leftarrow$ ReducedGB$(\{1 - Yf\} \cup S, <_{\bar{A}})$;
        IF $B \neq \{1\}$ THEN
            return false;
        END IF
    END FOR
    return true;
END
```

Next, for two segments (S, T, G) and (S', T', G'), we show how to check whether or not the intersection of both parameter spaces is empty, namely we check whether $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \cap (\mathbf{V}(S') \setminus \mathbf{V}(T'))$ is empty or not.

From $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \cap (\mathbf{V}(S') \setminus \mathbf{V}(T')) = (\mathbf{V}(S) \cap \mathbf{V}(S')) \setminus (\mathbf{V}(T) \cup \mathbf{V}(T')) = \mathbf{V}(S \cup S') \setminus \mathbf{V}(\langle T \rangle \cap \langle T' \rangle)$, it is enough to check that $\mathbf{V}(S \cup S') \setminus \mathbf{V}(\langle T \rangle \cap \langle T' \rangle) = \emptyset$ eventually. Here a Gröbner basis of the ideal intersection $\langle T \rangle \cap \langle T' \rangle$ can be computed. More details can be found in [1, 2]. `IntersectionGB` bellow is a function for computing a Gröbner basis of an ideal intersection.

Algorithm 8
`CaseIsDisjoint`$((S, T), (S', T'), <_{\bar{A}})$

INPUT: Two parameter spaces (S, T) and (S', T'), and a term order $<_{\bar{A}}$ on $T(\bar{A})$.
OUTPUT: "true" if $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \cap (\mathbf{V}(S') \setminus \mathbf{V}(T'))$ is empty, otherwise "false".

```
BEGIN
    $U \leftarrow$ IntersectionGB$(T, T', <_{\bar{A}})$;
    return CaseIsZero$((S \cup S', U), <_{\bar{A}})$;
END
```

Furthermore, for subsets $S, T \subset K[\bar{A}]$, we define a function `VarietyIsDisjoint`$(S, T, <_{\bar{A}})$ to check whether $\mathbf{V}(S) \cap \mathbf{V}(T)$ is empty or not. This function outputs "true" if the reduced Gröbner basis of $S \cup T$ is $\{1\}$, otherwise it outputs "false".

For two segments (S, T, G) and (S', T', G'), How do we compute the subtraction of both parameter spaces? Namely, how do we compute generators of algebraic sets V and W such that $V \setminus W = (\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T'))$? In order to see the answer, we first show next lemma concerning to decompose into disjoint three parameter spaces from the union of both parameter spaces.

Lemma 9
Let (S, T) and (S', T') be parameter spaces. If $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \cap (\mathbf{V}(S') \setminus \mathbf{V}(T')) \neq \emptyset$, then $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \cup (\mathbf{V}(S') \setminus \mathbf{V}(T')) = \big(\mathbf{V}(S') \setminus \mathbf{V}(T')\big) \cup \big((\mathbf{V}(S) \cap \mathbf{V}(S') \cap \mathbf{V}(T')) \setminus \mathbf{V}(T)\big) \cup \big(\mathbf{V}(S) \setminus (\mathbf{V}(S') \cup \mathbf{V}(T))\big)$. Moreover the three sets of right-hand side are parameter spaces respectively and they are disjoint each other.

Proof It is easy to see that $(\mathbf{V}(S) \cap \mathbf{V}(S') \cap \mathbf{V}(T')) \setminus \mathbf{V}(T) = \mathbf{V}(S \cup S' \cup T') \setminus \mathbf{V}(T)$ and $\mathbf{V}(S) \setminus (\mathbf{V}(S') \cup \mathbf{V}(T)) = \mathbf{V}(S) \setminus \mathbf{V}(\langle S' \rangle \cap \langle T \rangle)$, thus the three sets of right-hand side are parameter spaces respectively.

Next, we show disjointness. We set $C_1 = \mathbf{V}(S') \setminus \mathbf{V}(T')$, $C_2 = (\mathbf{V}(S) \cap \mathbf{V}(S') \cap \mathbf{V}(T')) \setminus \mathbf{V}(T)$ and $C_3 = \mathbf{V}(S) \setminus (\mathbf{V}(S') \cup \mathbf{V}(T))$. We first consider C_1 and C_2. $C_2 \subset \mathbf{V}(T')$ implies $C_1 \cap C_2 = \emptyset$. Next we consider C_1 and C_3. $C_1 \subset \mathbf{V}(S')$ implies $C_1 \cap C_3 = \emptyset$. Finally we consider C_2 and C_3. $C_2 \subset \mathbf{V}(S')$ implies $C_2 \cap C_3 = \emptyset$.

Finally, we show the equality. We have already known the disjointness, so that it is enough to show the equation $\big((\mathbf{V}(S) \setminus \mathbf{V}(T)) \cup (\mathbf{V}(S') \setminus \mathbf{V}(T'))\big) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T')) = (\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T')) = C_2 \cup C_3$. We will show this by calculation.

$$\begin{aligned}
& (\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T')) \\
= & \big((\mathbf{V}(S) \cap (\mathbf{V}(S') \cap \mathbf{V}(T'))) \setminus \mathbf{V}(T)\big) \cup \big((\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus \mathbf{V}(S')\big) \\
= & \big((\mathbf{V}(S) \cap \mathbf{V}(S') \cap \mathbf{V}(T')) \setminus \mathbf{V}(T)\big) \cup \big(\mathbf{V}(S) \setminus (\mathbf{V}(S') \cup \mathbf{V}(T))\big)
\end{aligned}$$

∎

Proposition 10
Let (S, T) and (S', T') be parameter spaces. Then

$$(\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T')) = \big(\mathbf{V}(S \cup S' \cup T') \setminus \mathbf{V}(T)\big) \cup \big(\mathbf{V}(S) \setminus (\mathbf{V}(\langle S' \rangle \cap \langle T \rangle))\big).$$

Moreover two parameter spaces of right-hand side are disjoint.

Algorithm 11
CaseSubtraction$((S, T, G), (S', T', G')), <_{\bar{A}})$

INPUT: *Two segments (S, T, G) and (S', T', G'), and a term order $<_{\bar{A}}$ on $T(\bar{A})$.*
OUTPUT: *A finite set $\mathcal{S}$ of segments. For parameter spaces of segments in $\mathcal{S}$ are disjoint each other, and its union coincides with $(\mathbf{V}(S) \setminus \mathbf{V}(T)) \setminus (\mathbf{V}(S') \setminus \mathbf{V}(T'))$.*

```
BEGIN
    $\mathcal{S} \leftarrow \emptyset$;
    IF CaseIsZero$((S \cup S' \cup T', T), <_{\bar{A}})$ = false THEN
        $\mathcal{S} \leftarrow \{(S \cup S' \cup T', T, G)\} \cup \mathcal{S}$;
    END IF
    IF CaseIsZero$((S, S' \cdot T), <_{\bar{A}})$ = false THEN
        $U \leftarrow$ IntersectionGB$(S', T, <_{\bar{A}})$;
        $\mathcal{S} \leftarrow \{(S, U, G)\} \cup \mathcal{S}$;
    END IF
    return $\mathcal{S}$;
END
```

3.2 Using Square Free Computation and Factorization

For a polynomial $f \in K[\bar{A}]$, The factorization of f given by $f = p_1^{n_1} \cdots p_l^{n_l}$ implies that $\mathbf{V}(f) = \mathbf{V}(p_1) \cup \cdots \cup \mathbf{V}(p_l)$. As a result, we always have $\deg(f) \geq \deg(p_i)$ $(1 \leq i \leq l)$, and coefficients of polynomials p_i become small. Furthermore in CGS, we empirically know that the cost of a factorization and a square-free computation tend to be smaller than that of a Gröbner basis computation. Also it is expected that these computations can reduce the maximum depth of recurrent algorithm. Using this idea and manipulations of parameter spaces given previous subsection, we can reformulate CGS.

Algorithm 12
$\mathtt{CGS}_1((a_1, \ldots, a_d), F, <_{\bar{A},\bar{X}})$

INPUT: *A d-tuple $(a_1, \ldots, a_d)$ of natural numbers, a finite set $F \subset K[\bar{A}, \bar{X}]$, and a term order $<_{\bar{A},\bar{X}}$.*
OUTPUT: *A set $\mathcal{H}$ of segments (No, S, T, G), where $No \in \mathbb{N}^k$, S, $T \subset K[\bar{A}]$, and G is the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$.*

```
BEGIN
    $G \leftarrow$ ReducedGB$(F, <_{\bar{A},\bar{X}})$;
    IF CaseIsZero$((F \cap K[\bar{A}], G \cap K[\bar{A}]), <_{\bar{A}}) = false$ THEN
        $\mathcal{H} \leftarrow \{((a_1, \ldots, a_d), F \cap K[\bar{A}], G \cap K[\bar{A}], \{1\})\}$;
    END IF
    IF $1 \in G$ THEN
        return $\mathcal{H}$;
    END IF
    $\{p_1, \ldots, p_s\} \leftarrow \bigcup_{g \in G \setminus K[\bar{A}]}$ Factors$(\mathrm{HC}_{<_{\bar{X}}}(g))$;    $\cdots$ (1)
    $p \leftarrow p_1 \cdot \cdots \cdot p_s$;
    IF VarietyIsDisjoint$(G \cap K[\bar{A}], \{p\}, <_{\bar{A}}) = true$ THEN    $\cdots$ (2)
        $\mathcal{H} \leftarrow \mathcal{H} \cup \{((a_1, \ldots, a_d), G \cap K[\bar{A}], \{1\}, G \setminus K[\bar{A}])\}$;
        return $\mathcal{H}$;
    ELSE
        IF CaseIsZero$((G \cap K[\bar{A}], \{p\}), <_{\bar{A}}) = false$ THEN
            $\mathcal{H} \leftarrow \mathcal{H} \cup \{((a_1, \ldots, a_d), G \cap K[\bar{A}], \{p\}, G \setminus K[\bar{A}])\}$;
        END IF
        FOR $i = 1, \ldots, s$ DO
            IF ReducedGB$((G \cap K[\bar{A}]) \cup \{p_i\}, <_{\bar{A}}) \neq \{1\}$ THEN    $\cdots$ (3)
                $\mathcal{H} \leftarrow \mathcal{H} \cup \mathtt{CGS}_1((a_1, \ldots, a_d, i), G \cup \{p_i\}, <_{\bar{A},\bar{X}})$;
            END IF
        END FOR
        return $\mathcal{H}$;
    END IF
END
```

Remark 13
We have the following remarks in the algorithm $\mathtt{CGS}_1$.

(1) *Let $\{h_1, \ldots, h_l\}$ be the set of the head coefficient ($\in K[\bar{A}]$) of every $g \in G \setminus K[\bar{A}]$ with respect to $<_{\bar{X}}$. It is easy to see $\mathbf{V}(h_1 \cdots h_l) = \mathbf{V}(p) = \mathbf{V}(p_1) \cup \cdots \cup \mathbf{V}(p_s)$. A function Factors$(f)$ computes the set of all prime factors of $f \in K[\bar{A}]$ over K.*

(2) *In general, for a segment (S, T, G), we have $\mathbf{V}(S) \setminus \mathbf{V}(T) = \mathbf{V}(S)$ if*

$$\mathbf{V}(S) \cap \mathbf{V}(T) = \emptyset \qquad (*1).$$

Then the $(S, \{1\}, G)$ is same as the (S, T, G) as a segment. In this algorithm, it is not necessary to call $\mathtt{CGS}_1$ *again when (*1) holds.*

(3) *If the reduced Gröbner basis G_i of $(G \cap K[\bar{A}]) \cup \{p_i\}$ is $\{1\}$, the reduced Gröbner basis of $F = G \cup \{p_i\}$ must be $\{1\}$. Furthermore the Gröbner basis computation of G_i with respect to $<_{\bar{A}}$ is still faster than that of F with respect to the elimination order $<_{\bar{A},\bar{X}}$. Therefore we should use step (3) in order to improve the performance.*

Example 14
*In Table 1, we compare two algorithms: "*CGS *with the manipulations of parameter spaces (with MPS)" and "*$\mathtt{CGS}_1$*". In the table, "Segments" means number of segments, "Max Depth" means maximum depth of a tree structure, and "Total Time (sec.)" shown in second. The details of problems and the computational environment can be found in Appendix A.*

Problem	Algorithm	Segments	Max Depth	Total Time (sec.)
S_2	**CGS with MPS**	25	11	14.94
	CGS$_1$	24	7	1.61
S_3	**CGS with MPS**	16	7	20.80
	CGS$_1$	19	5	2.56
S_4	**CGS with MPS**	39	7	144.59
	CGS$_1$	37	6	8.64

Table 1:

3.3 Using Minimal Gröbner Basis on $(K[\bar{A}])[\bar{X}]$

In Suzuki-Sato's algorithm, a Gröbner basis $F \subset (K[\bar{A}])[\bar{X}]$ with respect to $<_{\bar{X}}$ is obtained by Gröbner basis computation in $K[\bar{A}, \bar{X}]$ with respect to the elimination order $<_{\bar{A},\bar{X}}$. A Gröbner basis in $(K[\bar{A}])[\bar{X}]$ sometimes contains superfluous polynomials even though we compute the reduced Gröbner basis G in $K[\bar{A}, \bar{X}]$. This is a disadvantage in efficiency.

Example 15
For a polynomial set $F = \{aX + 1,\ bY + Y,\ aZ + bZ + Z\} \subset (\mathbb{Q}[a, b])[X, Y, Z]$, let $<_{\{a,b\}}$ the graded reverse lexicographic order such that $b <_{\{a,b\}} a$, $<_{\{X,Y,Z\}}$ the lexicographic order such that $Z <_{\{X,Y,Z\}} Y <_{\{X,Y,Z\}} X$, and $<_{\{a,b\},\{X,Y,Z\}}$ be the elimination order induced by $<_{\{a,b\}}$ and $<_{\{X,Y,Z\}}$ such that $\{a, b\} \ll \{X, Y, Z\}$. Then the reduced Gröbner basis of F with respect to $<_{\{a,b\},\{X,Y,Z\}}$ is

$$G = \{g_1, g_2, g_3, g_4, g_5\} = \{(a + b + 1)Z,\ (b + 1)Y,\ YZ,\ aX + 1,\ (b + 1)XZ - Z\}.$$

In this situation, we have

$$\mathrm{HM}_{<_{\{X,Y,Z\}}}(g_5) \in \langle \mathrm{HM}_{<_{X,Y,Z}}(g_1), \mathrm{HM}_{<_{X,Y,Z}}(g_2), \mathrm{HM}_{<_{X,Y,Z}}(g_3), \mathrm{HM}_{<_{X,Y,Z}}(g_4)\rangle$$

Indeed, $g_5 = Xg_1 - Zg_4$. This implies that g_5 is redundant as the Gröbner basis G in $(K[\bar{A}])[\bar{X}]$.

Therefore, we introduce the concept of *minimal Gröbner bases* in polynomial rings over a polynomial ring.

Definition 16 (Minimal Gröbner Basis)
Let $G \subset (K[\bar{A}])[\bar{X}]$ be a Gröbner basis with respect to $<_{\bar{X}}$. Then G is called a minimal Gröbner basis if it satisfies that

for every g in G, $\mathrm{HM}_{<_{\bar{X}}}(g) \notin \langle \mathrm{HM}_{<_{\bar{X}}}(G \setminus \{g\}) \rangle$.

Let G be a Gröbner basis of an ideal $I \subset (K[\bar{A}])[\bar{X}]$ with respect to $<_{\bar{X}}$. From the definition of Gröbner bases, if $\mathrm{HM}_{<_{\bar{X}}}(g)$ lies in $\langle \mathrm{HM}_{<_{\bar{X}}}(G \setminus \{g\}) \rangle$ for some $g \in G$, $G \setminus \{g\}$ is also a Gröbner basis of I. Thus, we can compute a minimal Gröbner basis by repeating the above process. Moreover, we can check whether or not $\mathrm{HM}_{<_{\bar{X}}}(g)$ lies in $\langle \mathrm{HM}_{<_{\bar{X}}}(G \setminus \{g\}) \rangle$ by the following proposition.

Proposition 17
Let G be a Gröbner basis in $(K[\bar{A}])[\bar{X}]$. For a $g \in G$, let $H = \{\mathrm{HC}_{<_{\bar{X}}}(g') \in K[\bar{A}] \mid g' \in G \setminus \{g\},\ \mathrm{HT}_{<_{\bar{X}}}(g') \mid \mathrm{HT}_{<_{\bar{X}}}(g)\}$, and H_G be a Gröbner basis of H with respect to $<_{\bar{A}}$. Then $\mathrm{HM}_{<_{\bar{X}}}(g) \in \langle \mathrm{HM}_{<_{\bar{X}}}(G \setminus \{g\}) \rangle$ if and only if $H_G \neq \emptyset$ and $\mathrm{NF}(\mathrm{HC}_{<_{\bar{A}}}(g), H_G, <_{\bar{A}}) = 0$.

Algorithm 18
`MinimalSet`$(G, <_{\bar{A}}, <_{\bar{X}})$

INPUT: A Gröbner basis $G \subset (K[\bar{A}])[\bar{X}]$, and term orders $<_{\bar{A}}$ and $<_{\bar{X}}$.
OUTPUT: A minimal Gröbner basis $G_{min} \subset (K[\bar{A}])[\bar{X}]$.

```
BEGIN
    $G_{min} \leftarrow \emptyset$;
    FOR EACH $g \in G$ DO
        $H \leftarrow \{\mathrm{HC}_{<_{\bar{X}}}(g') \mid g' \in G \setminus \{g\},\ \mathrm{HT}_{<_{\bar{X}}}(g') \mid \mathrm{HT}_{<_{\bar{X}}}(g)\}$;
        IF $H \neq \emptyset$ THEN
            $H_G \leftarrow$ ReducedGB$(H, <_{\bar{A}})$;
            IF NF$(\mathrm{HC}_{<_{\bar{X}}}(g), H_G, <_{\bar{A}}) \neq 0$ THEN
                $G_{min} \leftarrow G_{min} \cup \{g\}$;
            END IF
        ELSE
            $G_{min} \leftarrow G_{min} \cup \{g\}$;
        END IF
    END FOR
    return $G_{min}$;
END
```

Nabeshima [12] dealt with the concept of reduced Gröbner basis in $(K[\bar{A}])[\bar{X}]$, its uniqueness, and its computation. He showed that it takes high cost to compute the uniquely determined strong reduced Gröbner basis. Consequently, from a practical point of view computing strong reduced Gröbner basis has no advantage unless we need reducedness or uniqueness.

Example 19
In Table 2, we compare two algorithms: "$\mathtt{CGS}_1$ only" and "$\mathtt{CGS}_1$ with minimal Gröbner basis computation (with MinGB)". In the table, "Min Time" is a time in seconds for `MinimalSet`. *The details of problems and the computational environment can be found in Appendix A.*

Problem	Algorithm	Segments	Min Time	Total Time
S_1	**CGS$_1$ only**	36	***	468.6
	CGS$_1$ with MinGB	27	8.4	296.1
N_1	**CGS$_1$ only**	52	***	19.97
	CGS$_1$ with MinGB	16	0.17	1.06
N_2	**CGS$_1$ only**	57	***	58.09
	CGS$_1$ with MinGB	33	1.78	9.16

Table 2:

We can see that "$\mathtt{CGS}_1$ with MinGB" is faster than "$\mathtt{CGS}_1$ only". However we remark that `MinimalSet` may be a costly computation because it needs a Gröbner basis computation and a normal form computation over $\mathbb{Q}$, and these computation over $\mathbb{Q}$ may be costly. We have equipped our implementation with switch for using `MinimalSet`. We can choose both of algorithms easily when we compute CGS by our implementation.

3.4 Eliminating Duplicated Parameter Spaces

Suzuki-Sato's algorithm does not require that parameter spaces of segments are pairwise disjoint, so that keeping unnecessary duplicated parameter spaces involves a disadvantage for computational efficiency. In order to avoid this disadvantage, we give a mechanism to remove duplicated parameter spaces whenever they appear.

First, we define a partial order on serial numbers contained in segments.

Definition 20
For two segments (No_1, S_1, T_1, G_1) and (No_2, S_2, T_2, G_2) with $No_1 = (a_1, \ldots, a_i)$ and $No_2 = (b_1, \ldots, b_j)$, We say that $No_1 \geq No_2$ if

$$i \leq j \quad \text{and} \quad a_1 = b_1,\ a_2 = b_2, \ldots, a_i = b_i.$$

In this case, the segment (No_1, S_1, T_1, G_1) is called an ancestor of (No_2, S_2, T_2, G_2), and (No_2, S_2, T_2, G_2) is also called a descendant of (No_1, S_1, T_1, G_1). Moreover they are called a parent and a child respectively if $j = i + 1$.

In $\mathtt{CGS}_1$, for two segments (No_1, S_1, T_1, G_1) and (No_2, S_2, T_2, G_2) with $No_1 \geq No_2$, we always have $\mathbf{V}(S_1) \supset \mathbf{V}(S_2)$, however we must not remove (No_2, S_2, T_2, G_2) because $\mathbf{V}(S_1) \setminus \mathbf{V}(T_1) \not\supset \mathbf{V}(S_2)$. In contrast,

Proposition 21
For two segments (No_1, S_1, T_1, G_1) and (No_2, S_2, T_2, G_2), if

$$No_1 \not\geq No_2 \quad \text{and} \quad \mathbf{V}(S_1) \supset \mathbf{V}(S_2),$$

then (No_2, S_2, T_2, G_2) is removable.

Using this proposition, we can completely remove useless segments, however a large number of radical membership tests (Gröbner bases computation) to check $\mathbf{V}(S_1) \supset \mathbf{V}(S_2)$ is necessary. As a result, it probably gives damage to computational efficiency. In order to avoid this problem, we use the ideal membership test instead of the radical membership test.

Corollary 22
For two segments (No_1, S_1, T_1, G_1) and (No_2, S_2, T_2, G_2). If

$$No_1 \not\geq No_2 \quad \text{and} \quad \langle S_1 \rangle \subset \langle S_2 \rangle$$

then (No_2, S_2, T_2, G_2) is removable.

In this case, we remark that even if $\mathbf{V}(S_1) \supset \mathbf{V}(S_2)$, we may have $S_1 \not\subset \langle S_2 \rangle$. We can check $S_1 \subset \langle S_2 \rangle$ completely by using normal form computation only when S_2 is a Gröbner basis.

Actually, it is more efficient to use the radical membership test until the algorithm arrives at given depth and the ideal membership test at stages that are deeper than it.

Algorithm 23

```
ElimSP($(No_0, S_0), \mathcal{S}, N, <_{\bar{A}}$)

INPUT:  A pair $(No_0, S_0)$ of a serial number $No_0$ and a Gröbner basis $S_0 \subset K[\bar{A}]$,
        a finite set $\mathcal{S}$ of pairs, a natural number $N$, and a term order $<_{\bar{A}}$.    ···
        (1)
OUTPUT: "true" if the segment having a serial number $No_0$ is removable, other-
        wise "false".

BEGIN
    FOR EACH $(No, S) \in \mathcal{S}$ DO
        IF $No \not\geq No_0$ THEN
            IF Length($No$) $\leq N$ THEN   ···   (2)
                IF CaseIsZero($(S_0, S), <_{\bar{A}}$) $=$ true THEN   ···   (3)
                    return true;
                END IF
            ELSE
                $Flg \leftarrow 1$;
                FOR EACH $f \in S$ and $Flg = 1$ DO   ···   (4)
                    IF NF($f, S_0, <_{\bar{A}}$) $\neq 0$ THEN
                        $Flg \leftarrow 0$;
                    END IF
                END FOR
                IF $Flg = 1$ THEN
                    return true;
                END IF
            END IF
        END IF
    END FOR
    return false;
END
```

Remark 24

We have the following remarks in the algorithm ElimSP*.*

(1) *A pair (No_0, S_0) is made from the segment (No_0, S_0, T_0, G_0) and S_0 is a Gröbner basis with respect to $<_{\bar{A}}$. The set $\mathcal{S}$ consists of pairs (No, S) made from segments (No, S, T, G) which have already been computed and determined as ones in the final CGS. A natural number N denotes the fixed depth. The algorithm uses the radical membership test if a depth of the serial number is equal or smaller than N, otherwise it uses the ideal membership test in order to remove useless segments.*

(2) *A function* Length *returns the depth of a given serial number. For example,* Length$(No) = d$ *for a $No = (a_1, \ldots, a_d)$.*

(3) *We can check $\mathbf{V}(S_0) \subset \mathbf{V}(S)$ by* CaseIsZero$((S_0, S), <_{\bar{A}})$*. We remark that the function* CaseIsZero *uses the radical membership test.*

(4) *We check whether $S \subset \langle S_0 \rangle$ or not by using the ideal membership test at this* FOR *loop.*

Example 25

*In Table 3, we compare two algorithms: "*CGS$_1$ *with the eliminating duplicated parameter spaces (with Elim)" and "*CGS$_1$ *without it". In the table, "Elim Time" is the time for* ElimSP *in seconds. The details of problems and the computational environment can be found in Appendix A.*

In our implementation, the fixed number N in ElimSP is set at 3 tentatively. The number N can be changed easily in our implementation.

Problem	**Algorithm**	**Segments**	**Elim Time**	**Total Time**
S_2	**$\mathtt{CGS}_1$ without Elim**	1156	***	5.79
	$\mathtt{CGS}_1$ with Elim	24	0.06	1.67
S_4	**$\mathtt{CGS}_1$ without Elim**	93	***	9.47
	$\mathtt{CGS}_1$ with Elim	37	0.06	8.62
M_1	**$\mathtt{CGS}_1$ without Elim**	2311	***	11.15
	$\mathtt{CGS}_1$ with Elim	113	0.22	2.58

Table 3:

3.5 Using Prime Ideal Decomposition

The algorithm $\mathtt{CGS}_1$ works basically as follows. For an input $((a_1,\dots,a_d),F,<_{\bar{A},\bar{X}})$, it computes a Gröbner basis G of F first. Next it computes the union of all the prime factors of the every head coefficients of polynomials in $G \setminus K[\bar{A}]$. Finally it calls $\mathtt{CGS}_1$ itself again after it adds each prime factor into G. We call this flow *factorization method.* More detail of factorization method is as follows.

- **Factorization method**

  1. For an input $F \subset K[\bar{A},\bar{X}]$, compute a Gröbner basis G of F with respect to $<_{\bar{A},\bar{X}}$.
  2. Collect every head coefficient of $G \setminus K[\bar{A}]$, that is $H = \{\mathrm{HC}_{<_{\bar{X}}}(g) \in K[\bar{A}] \mid g \in G \setminus K[\bar{A}]\}$.
  3. Let $\{p_1,\cdots,p_s\}$ be the union of prime factors of all polynomials in H.
  4. Add p_i into G ($G \cup \{p_i\}$) for each $1 \le i \le s$, and apply this procedure recursively for $G \cup \{p_i\}$.

It is possible to improve the efficiency by using prime ideal decomposition in $K[\bar{A}]$ instead of adding polynomials. More detail is as follows.

- **Prime ideal decomposition method**

  1. For an input $F \subset K[\bar{A},\bar{X}]$, computes a Gröbner basis G of F with respect to $<_{\bar{A},\bar{X}}$.
  2. Collect every head coefficient of $G \setminus K[\bar{A}]$, that is $H = \{\mathrm{HC}_{<_{\bar{X}}}(g) \in K[\bar{A}] \mid g \in G \setminus K[\bar{A}]\}$.
  3. Let $\{p_1,\cdots,p_t\}$ be the union of prime factors of all polynomials in H.
  4. Compute the irredundant prime ideal decomposition of each radical of the ideal $\langle (G \cap K[\bar{A}]) \cup \{p_i\}\rangle$. Let $\{P_1,P_2,\dots,P_s\}$ be the union of isolated prime components of the every irredundant prime ideal decomposition. That is, for every irredundant prime ideal decomposition $\sqrt{\langle (G \cap K[\bar{A}]) \cup \{p_i\}\rangle} = P_{i,1} \cap P_{i,2} \cap \cdots \cap P_{i,u_i}$, let $\{P_1,P_2,\dots,P_s\}$ be the set of isolated prime components of
  $$\bigcup_{1 \le i \le t} \{P_{i,1},P_{i,2},\dots,P_{i,u_i}\}.$$
  5. Join P_i with G ($G \cup P_i$) for each $1 \le i \le s$, and apply this procedure recursively for $G \cup P_i$.

Proposition 26
Even though we use the prime ideal decomposition method instead of the factorization method, the algorithm $\mathtt{CGS}_1$ *terminates.*

Proof For the factorization method, we remark that the algorithm always terminate because $\langle G \cap K[\bar{A}]\rangle \subsetneq \langle (G \cap K[\bar{A}]) \cup \{p_i\}\rangle$ holds at every depth. Since $\{P_1, P_2, \ldots, P_s\}$ is irredundant prime component, for any $1 \leq i \leq s$ there exists $1 \leq j \leq t$ such that $\sqrt{\langle (G \cap K[\bar{A}]) \cup \{p_j\}\rangle} \subsetneq P_i$. This implies $\langle G \cap K[\bar{A}]\rangle \subsetneq P_i$, hence the algorithm using the prime ideal decomposition method always terminates by the same reason of the proof of the factorization method (original Suzuki-Sato's algorithm). ∎

Example 27
In Table 4, we compare two algorithms: "CGS_1 with the factorization method (with Fact)" and "CGS_1 with the prime ideal decomposition method (with Prim)". The details of problems and the computational environment can be found in Appendix A.

Problem	Algorithm	Segments	Total Time (sec.)
S_3	**CGS_1 with Fact**	17	2.50
	CGS_1 with Prim	15	0.87
M_2	**CGS_1 with Fact**	33	8.50
	CGS_1 with Prim	30	6.34

Table 4:

Of course, in general, the prime decomposition of an ideal is a costly computation, so that this method is not always faster than the other one. We should keep in mind that it is one of methods which are worth trying.

3.6 Using the Stability Condition of Zero-dimensional Radical Ideals

In [5], Kalkbrener showed the following theorem in addition to Theorem 3.

Theorem 28
Let π be a ring homomorphism from a Noetherian ring R with identity to a field K, and let $J \subset R$ be an ideal such that $J \subset \ker(\pi)$. Then the following conditions are equivalent.

1. *$\ker(\pi)$ is an isolated prime ideal of J which coincide with the corresponding primary components.*
2. *For any ideal $I \subset R[\bar{X}]$ such that $I \cap R = J$ and any term order $<_{\bar{X}}$, I is stable under π and $<_{\bar{X}}$.*

The following corollary is an easy consequence of Theorem 28

Corollary 29
For an ideal $I \subset (K[\bar{A}])[\bar{X}]$, if $I \cap K[\bar{A}]$ is a zero-dimensional radical ideal in $K[\bar{A}]$, then I is stable under $\sigma_{\bar{a}}$ and $<_{\bar{X}}$ for any $\bar{a} \in \mathbf{V}(I \cap K[\bar{A}])$.

The following two lemmas are important.

Lemma 30
For an ideal $I \subset K[\bar{A}, \bar{X}]$, let J be the ideal in $K[\bar{A}, \bar{X}]$ generated by $I \cap K[\bar{A}] \subset K[\bar{A}]$, and J' be the ideal in $K[\bar{A}, \bar{X}]$ generated by $\sqrt{I \cap K[\bar{A}]} \subset K[\bar{A}]$ respectively. Then

$$J' = \sqrt{J}.$$

Proof We first show $J' \subset \sqrt{J}$. For any generator $f \in \sqrt{I \cap K[\bar{A}]}$ in J', there exists an $M \in \mathbb{N}$ such that $f^M \in I \cap K[\bar{A}]$, thus $f^M \in J$. This implies $f \in \sqrt{J}$.

Next we show $J' \supset \sqrt{J}$. From the definition of J and J', we obviously have the $J \subset J'$. Taking the radical of the both, we have the $\sqrt{J} \subset \sqrt{J'}$. Thus it is enough to show the $J' = \sqrt{J'}$.

$J' \subset \sqrt{J'}$ is obvious, thus we show $J' \supset \sqrt{J'}$. For any $f \in \sqrt{J'}$, we suppose that f is ordered with respect to $<_{\bar{X}}$, where $<_{\bar{X}}$ is any term order on $T(\bar{X})$, that is $f = c_1\bar{X}^{\alpha_1} + c_2\bar{X}^{\alpha_2} + \cdots + c_j\bar{X}^{\alpha_j}$, $(c_i \in K[\bar{A}],\ \bar{X}^{\alpha_1} >_{\bar{X}} \bar{X}^{\alpha_2} >_{\bar{X}} \cdots >_{\bar{X}} \bar{X}^{\alpha_j})$. Then there exists an $M \in \mathbb{N}$ such that $f^M \in J'$. For this f^M, it can be written as

$$f^M = (c_1\bar{X}^{\alpha_1})^M + (\textit{monomials which have the term less than } \bar{X}^{\alpha_1 M} \textit{ w.r.t. } >_{\bar{X}}).$$

Then we have $c_1^M \in \sqrt{I \cap K[\bar{A}]}$ because f^M also can be written as $f^M = \sum_i d_i g_i$, $(d_i \in K[\bar{A}, \bar{X}],\ g_i \in \sqrt{I \cap K[\bar{A}]})$, this shows that c_1^M can be written as $c_1^M = \sum_i h_{1,i} g_i$, $(h_{1,i} \in K[\bar{A}])$. This implies $c_1\bar{X}^{\alpha_1} \in J'$.

Next we set $f_1 = f - c_1\bar{X}^{\alpha_1}$. Then $f_1 \in \sqrt{J'}$, thus applying above argument again, we obtain $c_2\bar{X}^{\alpha_2} \in J'$. Repeating this way, we obtain $c_1\bar{X}^{\alpha_1}, \ldots, c_j\bar{X}^{\alpha_j} \in J'$. This implies $f \in J'$. ∎

Lemma 31
For an ideal $I \subset K[\bar{A}, \bar{X}]$, let J be an ideal in $K[\bar{A}, \bar{X}]$ generated by $I \cap K[\bar{A}] \subset K[\bar{A}]$ and J' be an ideal in $K[\bar{A}, \bar{X}]$ generated by $\sqrt{I \cap K[\bar{A}]} \subset K[\bar{A}]$ respectively. Then

$$(I + J') \cap K[\bar{A}] = \sqrt{I \cap K[\bar{A}]}$$

Proof $(I + J') \cap K[\bar{A}] \supset \sqrt{I \cap K[\bar{A}]}$ is obvious.

We show $(I + J') \cap K[\bar{A}] \subset \sqrt{I \cap K[\bar{A}]}$. For any $f \in (I + J') \cap K[\bar{A}]$, there exists a $p \in I$ and a $q \in J'$ such that $f = p + q$. Since $J' = \sqrt{J}$ obtained by Lemma 30, there exists an $M \in \mathbb{N}$ such that $q^M \in J \subset I$. Thus $f^M = (p + q)^M \in I$. We consequently have $f^M \in I \cap K[\bar{A}]$ since $f^M \in K[\bar{A}]$. This implies $f \in \sqrt{I \cap K[\bar{A}]}$. ∎

Using the above result, if $\langle G \cap K[\bar{A}] \rangle$ becomes zero-dimensional during computing a CGS, we do not need to compute segments at deeper stages than the stage, so that we determine a terminal segment at the stage. Specifically, we insert the following algorithm into $\texttt{CGS}_1$

Algorithm 32
$\texttt{ZeroDimRoutine}((a_1, \ldots, a_d), F, <_{\bar{A},\bar{X}})$

```
INPUT:  A $d$-tuple $(a_1, \ldots, a_d)$ of natural numbers, a finite set $F \subset K[\bar{A}, \bar{X}]$ such
        that $\langle F \cap K[\bar{A}] \rangle$ is zero-dimensional, and a term order $<_{\bar{A},\bar{X}}$.
OUTPUT: A set $\mathcal{H}$ of segments $(No, S, T, G)$, where $No \in \mathbb{N}^k$, $S,\ T \subset K[\bar{A}]$, and
        $G$ is the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$.

BEGIN
    $G_z \leftarrow \texttt{ZeroRadical}(F \cap K[\bar{A}], <_{\bar{A}})$;
    $F_z \leftarrow (F \setminus K[\bar{A}]) \cup G_z$;
    $G \leftarrow \texttt{ReducedGB}(F_z, <_{\bar{A},\bar{X}})$;
    return $\left\{\left((a_1, \ldots, a_d), G \cap K[\bar{A}], \{1\}, G \setminus K[\bar{A}]\right)\right\}$;
END
```

In the algorithm, F_z is a set of generator of $\langle F \rangle + \langle G_z \rangle$ because $\langle F \rangle + \langle G_z \rangle = \langle F \setminus K[\bar{A}] \rangle + \langle G_z \rangle = \langle (F \setminus K[\bar{A}]) \cup G_z \rangle$, where $\langle F \rangle \cap K[\bar{A}] \subset \langle G_z \rangle$.

Example 33

In Table 5, we compare two algorithms: "$\mathtt{CGS}_1$ *with the zero-dimensional radical manipulation (with Zrad)" and "*$\mathtt{CGS}_1$ *without it". In the table, "NZC Time" is the time in seconds for computing segments which have a non zero-dimensional parameter space. The details of problems and the computational environment can be found in Appendix A.*

Problem	Algorithm	Segments	NZC Time	Total Time
S_5	$\mathtt{CGS}_1$ without Zrad	51	4893	6540
	$\mathtt{CGS}_1$ with Zrad	45	3960	4422
S_6	$\mathtt{CGS}_1$ without Zrad	***	***	> 12 hours
	$\mathtt{CGS}_1$ with Zrad	35	403	21563

Table 5:

If a zero-dimensional parameter space does not appear during computing a CGS, we cannot use this method. In [7], *discrete comprehensive Gröbner bases* (DCGB) also gives a parametric Gröbner basis for a zero-dimensional parameter space. Thus we can also use a DCGB computation for this situation.

4 An Efficient Method for Suzuki-Sato's CGS Algorithm Using Stability of Gröbner Bases

In this section, we give the detail of our main results. First, we show another difficulty of $\mathtt{CGS}_1$ and its existing solution (Nabeshima's method).

4.1 Nabeshima's Approach

Nabeshima [13] proposed a smart method for speeding-up $\mathtt{CGS}_1$. It uses the following property, which often holds in $(K[\bar{A}])[\bar{X}]$

For a Gröbner basis G in $(K[\bar{A}])[\bar{X}]$,

$$\text{there exist distinct } g \text{ and } g' \in G \text{ such that } \mathrm{HT}_{<_{\bar{X}}}(g) \mid \mathrm{HT}_{<_{\bar{X}}}(g') \qquad (*2)$$

If the coefficient ring is a field K, the reduced Gröbner basis never has property (*2). However it often holds when the coefficient ring is a Noetherian ring which is not a field. In particular, even though we compute the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$ with respect to the elimination order such that $\bar{A} \ll \bar{X}$, the basis often has property (*2) as a Gröbner basis in $(K[\bar{A}])[\bar{X}]$. If the property (*2) holds, $\mathtt{CGS}_1$ works toward generating many superfluous segments, so that the total efficiency becomes worse. In order to avoid this, Nabeshima introduced Gröbner basis computation in which inequations ($\neq 0$) are treated. In (*2), if we assume $\mathrm{HC}_{<_{\bar{X}}}(g) \neq 0$, we must not consider the condition of $\mathrm{HC}_{<_{\bar{X}}}(g')$.

Example 34

This example is taken from Nabeshima [13].

For $G = \{f_1, f_2, f_3\} = \{aX^3,\ bX^2,\ cX\} \subset (\mathbb{Q}[a, b, c])[X]$, *let* a, b, c *be parameters, and let* X *be a variable.*

$G \subset K[a, b, c, X]$ *is already the reduced Gröbner basis with respect to the elimination order* $<_{\{a,b,c\},\{X\}}$ *such that* $\{a, b, c\} \ll \{X\}$*. This implies that* G *is also a Gröbner basis with respect to* $<_{\{X\}}$

in $(\mathbb{Q}[a,b,c])[X]$. Then we have $\mathrm{HT}_{<_{\{X\}}}(f_2) \mid \mathrm{HT}_{<_{\{X\}}}(f_1)$, $\mathrm{HT}_{<_{\{X\}}}(f_3) \mid \mathrm{HT}_{<_{\{X\}}}(f_2)$, and $\mathrm{HT}_{<_{\{X\}}}(f_3) \mid \mathrm{HT}_{<_{\{X\}}}(f_1)$, so that we can see that (*2) holds.

Suzuki-Sato's algorithm basically works as Fig. 2 when we compute a CGS of G by the algorithm, and finally obtain 16 segments.

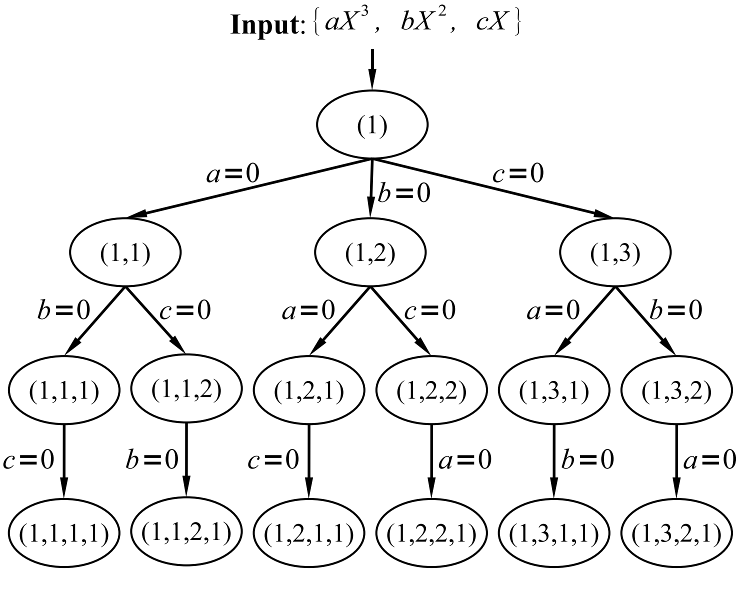

Fig. 2:

Even though we apply all manipulations given in Section 3, we finally obtain 8 segments.

$$\begin{cases} \{cX\} & \text{if } a = 0,\ bc \neq 0 \\ \{cX\} & \text{if } a = b = 0,\ c \neq 0 \\ \{cX\} & \text{if } b = 0,\ ac \neq 0 \\ \{bX^2\} & \text{if } a = c = 0,\ b \neq 0 \end{cases} \qquad \begin{cases} \emptyset & \text{if } a = b = c = 0 \\ \{aX^3\} & \text{if } b = c = 0,\ a \neq 0 \\ \{bX^2\} & \text{if } c = 0,\ ab \neq 0 \\ \{cX\} & \text{if } abc \neq 0 \end{cases}$$

Next, we will consider an application of inequations ($\neq 0$). Once we have obtained $\mathrm{HT}_{<_{\{X\}}}(f_3) \mid \mathrm{HT}_{<_{\{X\}}}(f_2)$, $\mathrm{HT}_{<_{\{X\}}}(f_3) \mid \mathrm{HT}_{<_{\{X\}}}(f_1)$, and $\mathrm{HT}_{<_{\{X\}}}(f_2) \mid \mathrm{HT}_{<_{\{X\}}}(f_1)$, we can assume that $b \neq 0$ or $c \neq 0$. In this situation, if we assume $c \neq 0$, we can ignore the condition of a and b. Thus a Gröbner basis of $\langle G \rangle$ is $G_1 = \{cX\}$ if $c \neq 0$.

We next consider the case $c = 0$ $(\mathbf{V}(c))$. For $G_{\{c=0\}} = \{f_1, f_2\} = \{aX^3,\ bX^2\}$, we have obtained $\mathrm{HT}_{<_{\{X\}}}(f_2) \mid \mathrm{HT}_{<_{\{X\}}}(f_1)$, so that we assume that $b \neq 0$ $(\mathbf{V}(c)\backslash\mathbf{V}(b))$. Then we can ignore the condition of a, and a Gröbner basis of $\langle G \rangle$ is $G_2 = \{bX^2\}$.

Finally we consider the case $c = 0$, $b = 0$ $(\mathbf{V}(b,c))$. $G_{\{b=0,c=0\}} = \{f_1\} = \{aX^3\}$ is already a Gröbner basis of $\langle G \rangle$. In particular, a Gröbner basis is $G_3 = \{aX^3\}$ if $a \neq 0$ $(\mathbf{V}(b,c) \setminus \mathbf{V}(a))$, and $G_4 = \emptyset$ if $a = 0$ $(\mathbf{V}(a,b,c))$. This argument is illustrated as Fig. 3.

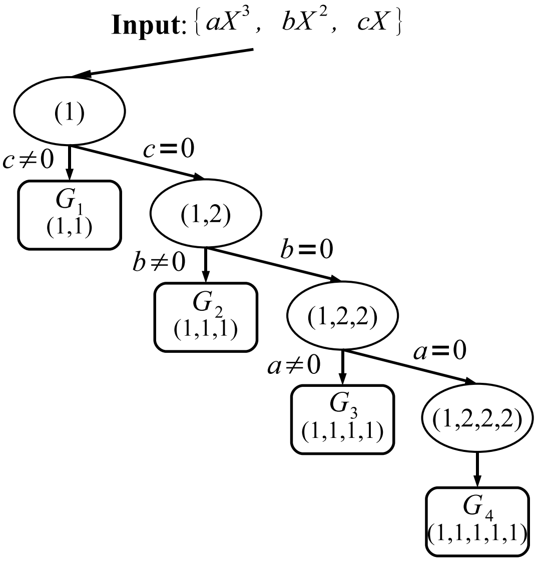

Fig. 3:

We eventually see that the following 4 segments suffice to form a CGS of $\langle G\rangle$*.*

$$\begin{cases} G_1 = \{cX\} & \text{if}\, c \neq 0 \\ G_2 = \{bX^2\} & \text{if}\, b \neq 0,\ c = 0 \\ G_3 = \{aX^3\} & \text{if}\, a \neq 0,\ b = c = 0 \\ G_4 = \emptyset & \text{if}\, a = b = c = 0 \end{cases}$$

In order to apply this idea, Nabeshima's method computes a Gröbner basis together with a new temporary variable r.

Theorem 35 (Nabeshima 2007)
Let $R = K[\bar{A}]$*,* I *an ideal in* $R[\bar{X}]$*, and* $H = \{g, g_1, \ldots, g_l\}$ *be a Gröbner basis of* I *with respect to* $<_{\bar{X}}$*. Then select* g *from* H*, and we regard the variable* r *as* $r = 1/\mathrm{HC}_{<_{\bar{X}}}(g)$ *where* r *is a new temporary variable except* $\bar{A}$ *and* $\bar{X}$*, and let* $g' = \mathrm{HT}_{<_{\bar{X}}}(g) + r\cdot(g - \mathrm{HM}_{<_{\bar{X}}}(g))$*. Furthermore, for* $H' = (H\setminus\{g\})\cup\{g'\} = \{g', g_1, \ldots, g_l\} \subset (K[r,\bar{A}])[\bar{X}]$*, let* G' *be a Gröbner basis of* $\langle H'\rangle$ *with respect to* $<_{\bar{X}}$*. Then for*

$$G = \{f \in R[\bar{X}] \mid f \neq 0,\ f = \mathrm{HC}_{<_{\bar{X}}}(g)^k \cdot \sigma_{\{r=\frac{1}{\mathrm{HC}_{<_{\bar{X}}}(g)}\}}(q),\ \deg_r(q) = k,\ q \in G'\}$$

and $\{h_1, \ldots, h_e\} = \{\mathrm{HC}_{<_{\bar{X}}}(f) \in R \mid f \in G\}$*,* $\sigma_{\bar{a}}(G)$ *is a Gröbner basis of* $\langle\sigma_{\bar{a}}(I)\rangle$ *with respect to* $<_{\bar{X}}$ *for any* $\bar{a} \in L^m \setminus (\mathbf{V}(\mathrm{HC}_{<_{\bar{X}}}(g)) \cup \mathbf{V}(h_1) \cup \cdots \cup \mathbf{V}(h_e)) = L^m \setminus \mathbf{V}\big(\mathrm{LCM}(\mathrm{HC}_{<_{\bar{X}}}(g), h_1, \ldots, h_e)\big)$*.* *(*$\sigma_{\{r=\frac{1}{\mathrm{HC}_{<_{\bar{X}}}(g)}\}}(q)$ *means substituting* $1/\mathrm{HC}_{<_{\bar{X}}}(g)$ *for the variable* r *in* q*)*

An outline of Nabeshima's method is as follows. For (F, N_z, p), $F \subset (K[r,\bar{A}])[\bar{X}]$ is a finite set, $N_z \subset K[\bar{A}]$ is a finite set of non-zero condition polynomials, and $p \subset K[\bar{A}]$ is a current non-zero condition polynomial.

Algorithm 36
$\mathtt{NabCGS}(F, N_z, p)$

```
1. Compute a Gröbner basis $G' \subset (K[r,\bar{A}])[\bar{X}]$ of $F$ with respect to $<_{\bar{X}}$. This computation is produced by Gröbner basis computation on $K[r,\bar{A},\bar{X}]$ with the elimination term order such that $(\{r\} \cup \bar{A}) \ll \bar{X}$.
2. If $p \neq 0$, then let $G = \{f \in (K[\bar{A}])[\bar{X}] \mid f \neq 0,\ f = p^k \cdot \sigma_{\{r=\frac{1}{p}\}}(q),\ \deg_r(q) = k,\ q \in G'\}$. Otherwise let $G = G'$.
3. Let $E = \{f \in G \setminus K[\bar{A}] \mid \exists f' \in G \setminus \{f\}$ such that $\mathrm{HT}_{<_{\bar{X}}}(f) \mid \mathrm{HT}_{<_{\bar{X}}}(f')\}$. If $E \neq \emptyset$, then go to step 4. Otherwise go to step 7.
4. Select the $g$ from $E$ such that $\mathrm{HT}_{<_{\bar{X}}}(g)$ is the smallest in $\mathrm{HT}_{<_{\bar{X}}}(E)$ with respect to $<_{\bar{X}}$.
5. Let $g' = \mathrm{HT}_{<_{\bar{X}}}(g) + r \cdot (g - \mathrm{HM}_{<_{\bar{X}}}(g))$ and $F' = (F \setminus \{g\}) \cup \{g'\}$.
6. Apply this procedure recursively to $(F', N_z \cup \{\mathrm{HC}_{<_{\bar{X}}}(g)\}, \mathrm{HC}_{<_{\bar{X}}}(g))$ and
   $(G \cup \{\mathrm{HC}_{<_{\bar{X}}}(g)\}, N_z, 0)$.
7. If $\langle G \rangle = \langle 1 \rangle$, then return $\mathcal{H}$.
8. Let $\{h_1, \ldots, h_l\} = \bigcup_{f \in G \setminus K[\bar{A}]} \mathtt{Factors}(\mathrm{HC}_{<_{\bar{X}}}(f))$.
9. Let $h = h_1 \cdot \cdots \cdot h_l$, and add a segment $(F \cap K[\bar{A}], \{h\}, G)$ to $\mathcal{H}$.
10. Apply this procedure recursively to each $(G \cup \{h_i\}, N_z, 0)$ where $1 \leq i \leq l$.
```

More detail of NabCGS can be found in [13]. In NabCGS, step 1 and 7–10 are essentially the same as $\mathtt{CGS}_1$. That is, if the variable E is empty, which means that $G \setminus K[\bar{A}]$ does not satisfy (*2), then NabCGS works the same as $\mathtt{CGS}_1$. This computation can reduce the number of segments against an output of $\mathtt{CGS}_1$ in many cases, however it may generate time-consuming Gröbner basis computation (in the step 1–6) by increasing variables. We will see the detail in Section 4.4.

Now, what is a method which prevents (*2) from making a lot of unnecessary segments without increasing variables? In the next subsection, we give a solution for this question.

4.2 Another Stability Criterion

In Example 34, for the output CGS of $G = \{f_1, f_2, f_3\} = \{aX^3,\ bX^2,\ cX\}$ by $\mathtt{CGS}_1$, the parameter condition of segment whose serial number is (1) is $a \neq 0,\ b \neq 0,\ c \neq 0$. On the other hand, we have already known that it suffices to consider $c \neq 0$. Selecting f_3 first is caused by that $\mathrm{HT}_{<\{X\}}(f_3) = X$ is "smallest" and it can divide the head term of another polynomials. So that we can reduce the number of conditions which should be considered. In order to illustrate this argument more generally, we recall the following well known notion.

Definition 37
For α_1 and $\alpha_2 \in T(\bar{X})$, the divisibility relation $\preceq_{div}$ on $T(\bar{X})$ is defined as follows.

$$\alpha_1 \preceq_{div} \alpha_2 \quad \Longleftrightarrow \quad \text{there exist } \beta \in T(\bar{X}) \text{ such that } \alpha_2 = \alpha_1\beta.$$

Then $\preceq_{div}$ becomes a partial order on $T(\bar{X})$.

In other words, selecting f_3 from G is caused by that $\{X\}$ is the minimal basis of $\mathrm{HT}_{<_{\bar{X}}}(G) = \{X^3,\ X^2,\ X\}$ with respect to divisibility relation. Furthermore, $\{X\}$ becomes minimal basis of $\mathrm{HT}_{<_{\bar{X}}}(\sigma_{\bar{a}}(G))$ for $\sigma_{\bar{a}}$ if $\bar{a} \in L^3 \setminus \mathbf{V}(c)$. (It is not necessary to consider a condition of a and b)

In general, for a given Gröbner basis $G \subset R[\bar{X}]$, $\sigma(G)$ also become a Gröbner basis in $K[\bar{X}]$ for a specialization σ which unchanges the minimal basis of $\mathrm{HT}_{<_{\bar{X}}}(G)$.

Theorem 38

Let $R = K[\bar{A}]$, let I be an ideal in $R[\bar{X}]$, and let $G = \{g_1, \dots, g_s\}$ be a Gröbner basis of I with respect to $<_{\bar{X}}$. Moreover, $\mathrm{MHT}_{<_{\bar{X}}}(G)$ denotes the minimal basis of $\mathrm{HT}_{<_{\bar{X}}}(G)$ in $T(\bar{X})$ with respect to the divisibility relation, that is $\mathrm{MHT}_{<_{\bar{X}}}(G) = \{\mathrm{HT}_{<_{\bar{X}}}(g) \in T(\bar{X}) \mid \mathrm{HT}_{<_{\bar{X}}}(g') \nmid \mathrm{HT}_{<_{\bar{X}}}(g),\ \mathrm{HT}_{<_{\bar{X}}}(g') \neq \mathrm{HT}_{<_{\bar{X}}}(g),\ g, g' \in G\}$. If a specialization σ satisfies

$$\sigma(\mathrm{HC}_{<_{\bar{X}}}(g)) \neq 0 \textit{ for any } g \in G \textit{ such that } \mathrm{HT}_{<_{\bar{X}}}(g) \in \mathrm{MHT}_{<_{\bar{X}}}(G),$$

then $\sigma(G)$ is a Gröbner basis of $\langle \sigma(I) \rangle$ with respect to $<_{\bar{X}}$.

Proof By Theorem 3, it is enough to show that I is stable under the σ and $<_{\bar{X}}$, that is we show that

$$\langle \sigma(\langle \mathrm{HM}_{<_{\bar{X}}}(I) \rangle) \rangle = \langle \mathrm{HM}_{<_{\bar{X}}}(\sigma(I)) \rangle.$$

$\langle \sigma(\langle \mathrm{HM}_{<_{\bar{X}}}(I) \rangle) \rangle \subset \langle \mathrm{HM}_{<_{\bar{X}}}(\sigma(I)) \rangle$ is obvious. In order to show the reverse inclusion, it is enough to show that $\langle \sigma(\mathrm{HM}_{<_{\bar{X}}}(g_1)), \dots, \sigma(\mathrm{HM}_{<_{\bar{X}}}(g_s)) \rangle \supset \langle \mathrm{HM}_{<_{\bar{X}}}(\sigma(I)) \rangle$. Namely, it is enough to show that for $f \in I$ with $\sigma(f) \neq 0$,

$$\begin{gathered} \text{there exists } \mathrm{HT}_{<_{\bar{X}}}(g_i) \text{ such that } \mathrm{HT}_{<_{\bar{X}}}(g_i) \text{ divides } \mathrm{HT}_{<_{\bar{X}}}(\sigma(f)) \text{ and } \sigma(\mathrm{HC}_{<_{\bar{X}}}(g_i)) \neq 0, \\ \text{where } 1 \leq i \leq s. \end{gathered} \tag{*3}$$

We do the proof by induction on $<_{\bar{X}}$.

(Induction basis)

If $\mathrm{HT}_{<_{\bar{X}}}(f) = 1$, $\sigma(f) \neq 0$ implies that $\mathrm{HC}_{<_{\bar{X}}}(\sigma(f)) = \sigma(\mathrm{HC}_{<_{\bar{X}}}(f))$.
Thus $\mathrm{HT}_{<_{\bar{X}}}(\sigma(f)) = \mathrm{HT}_{<_{\bar{X}}}(f)$ is divisible by some terms in $\{\mathrm{HT}_{<_{\bar{X}}}(g_1), \dots, \mathrm{HT}_{<_{\bar{X}}}(g_s)\}$.

(Induction step)

We assume that (*3) holds for polynomials whose head terms are smaller than $\mathrm{HT}_{<_{\bar{X}}}(f)$ with respect to $<_{\bar{X}}$.

If $\sigma(\mathrm{HC}_{<_{\bar{X}}}(f)) \neq 0$, $\mathrm{HC}_{<_{\bar{X}}}(\sigma(f)) = \sigma(\mathrm{HC}_{<_{\bar{X}}}(f))$ is obvious. Thus $\mathrm{HT}_{<_{\bar{X}}}(\sigma(f)) = \mathrm{HT}_{<_{\bar{X}}}(f)$ is divisible by some terms in $\{\mathrm{HT}_{<_{\bar{X}}}(g_1), \dots, \mathrm{HT}_{<_{\bar{X}}}(g_s)\}$.

Finally, we consider the case $\sigma(\mathrm{HC}_{<_{\bar{X}}}(f)) = 0$. From the definition of σ and $f \in I$, we have $\mathrm{HT}_{<_{\bar{X}}}(g) \mid \mathrm{HT}_{<_{\bar{X}}}(f)$ and $\sigma(\mathrm{HC}_{<_{\bar{X}}}(g)) \neq 0$ for some $g \in G$ such that $\mathrm{HT}_{<_{\bar{X}}}(g) \in \mathrm{MHT}_{<_{\bar{X}}}(G)$. Then for the g, defining

$$f' = \mathrm{HC}_{<_{\bar{X}}}(g) f - \mathrm{HC}_{<_{\bar{X}}}(f) \frac{\mathrm{HT}_{<_{\bar{X}}}(f)}{\mathrm{HT}_{<_{\bar{X}}}(g)} g,$$

we obtain $\mathrm{HT}_{<_{\bar{X}}}(\sigma(f')) = \mathrm{HT}_{<_{\bar{X}}}(\sigma(f))$ and $\mathrm{HT}_{<_{\bar{X}}}(f') <_{\bar{X}} \mathrm{HT}_{<_{\bar{X}}}(f)$.
Hence, by the induction hypothesis, $\mathrm{HT}_{<_{\bar{X}}}(\sigma(f'))$ is divisible by some terms in $\{\mathrm{HT}_{<_{\bar{X}}}(g_1), \dots, \mathrm{HT}_{<_{\bar{X}}}(g_s)\}$. This implies that $\mathrm{HT}_{<_{\bar{X}}}(\sigma(f))$ is also divisible by these. ∎

This theorem can be extended to the following corollary.

Corollary 39

Let $R = K[\bar{A}]$, I an ideal in $R[\bar{X}]$, and $G = \{g_1, \dots, g_s\}$ be a Gröbner basis of I with respect to $<_{\bar{X}}$. We assume that the g_is are ordered in such a way that $g_1, \dots, g_r \notin R$ for $1 \leq r \leq s$ and $g_{r+1}, \dots, g_s \in R$, and let $G' = \{g_1, \dots, g_r\}$. If a specialization σ satisfies that

$$\begin{array}{c}\sigma(g_{r+1}) = \cdots = \sigma(g_s) = 0 \textit{ and} \\ \sigma(\mathrm{HC}_{<_{\bar{X}}}(g)) \neq 0 \textit{ for any } g \in G' \textit{ such that } \mathrm{HT}_{<_{\bar{X}}}(g) \in \mathrm{MHT}_{<_{\bar{X}}}(G'),\end{array} \qquad \textit{(*4)}$$

*then $\sigma(G')$ is a Gröbner basis of $\langle\sigma(I)\rangle$ with respect to $<_{\bar{X}}$. Moreover, the specialization $\sigma_{\bar{a}}$ induced by $\bar{a} \in L^m$ satisfies (*4) if and only if*

$$\bar{a} \in \mathbf{V}(g_{r+1}, \ldots, g_s) \setminus \Big(\mathbf{V}(h_1) \cup \cdots \cup \mathbf{V}(h_l)\Big),$$

where $\{h_1, \ldots, h_l\} = \{g \in G' \mid \mathrm{HT}_{<_{\bar{X}}}(g) \in \mathrm{MHT}_{<_{\bar{X}}}(G')\}$.

Example 40
Let $G = \{f_1, f_2, f_3, f_4, f_5\} = \{a_{1,1}XY^2 + a_{1,2}XY + a_{1,3}X,\ a_{2,1}XY,\ a_{3,1}X^3 + a_{3,2}XY,\ a_{4,1}X^2Y + a_{4,2}XY + a_{4,3}Y,\ a_{5,1}X^2 + a_{5,2}XY\} \subset (K[\bar{A}])[X, Y]$ be a Gröbner basis with respect to the total degree order $<_{\bar{X}}$. Thus, the minimal basis of $\mathrm{HT}_{<_{\bar{X}}}(G)$ is $\{\mathrm{HT}_{<_{\bar{X}}}(f_2),\ \mathrm{HT}_{<_{\bar{X}}}(f_5)\} = \{XY,\ X^2\}$. Then, using Corollary 39, for any

$$\bar{a} \in L^m \setminus \mathbf{V}(a_{2,1} \cdot a_{5,1}),$$

$\sigma_{\bar{a}}(G)$ is a Gröbner basis of $\langle\sigma_{\bar{a}}(G)\rangle$ with respect to $<_{\bar{X}}$. Note that we need not to consider conditions of $a_{i,j}$ appeared in f_1, f_3, and f_4.

In order to improve the efficiency, we rewrite Corollary 39 by using factorization.

Corollary 41
*Let $R = K[\bar{A}]$, I an ideal in $R[\bar{X}]$, and $G = \{g_1, \ldots, g_s\}$ be a Gröbner basis of I with respect to $<_{\bar{X}}$. We assume that g_is are ordered in such a way that $g_1, \ldots, g_r \notin R$ for $1 \le r \le s$ and $g_{r+1}, \ldots, g_s \in R$, and let $G' = \{g_1, \ldots, g_r\}$. Then the specialization $\sigma_{\bar{a}}$ induced by $\bar{a} \in L^m$ satisfies (*4) if and only if*

$$\bar{a} \in \mathbf{V}(g_{r+1}, \ldots, g_s) \setminus \mathbf{V}(p_1 \cdots p_t), \qquad (*5)$$

where $\{p_1, \ldots, p_t\}$ is the union of prime factors of $\{g \in G' \mid \mathrm{HT}_{<_{\bar{X}}}(g) \in \mathrm{MHT}_{<_{\bar{X}}}(G')\}$.

4.3 The Algorithm Using Another Stability Criterion

In this subsection, we show a new algorithm for computing a CGS into which we incorporate the results of Section 4.2.

First, using Corollary 41, we give an algorithm which outputs a determined segment as ones in the final CGS and parameter spaces which should be computed next.

Algorithm 42
`NewBranches`$(G, <_{\bar{A}}, <_{\bar{X}})$

```
INPUT:  A Gröbner basis $G$ in $K[\bar{A}, \bar{X}]$, and term orders $<_{\bar{A}}$ and $<_{\bar{X}}$.
OUTPUT: A pair $(\mathcal{S}, \mathcal{N})$. $\mathcal{S}$ forms a set of determined segments, and $\mathcal{N}$ forms a
        set of parameter spaces which should be computed next.

BEGIN
    $G_{\bar{A}} \leftarrow G \cap K[\bar{A}]$;
    $G_{\bar{X}} = \{g_1, \ldots, g_r\} \leftarrow G \setminus K[\bar{A}]$;
    $T_{min} \leftarrow$ MHT$(G_{\bar{X}}, <_{\bar{X}})$;   $\cdots$ (1)
    $C_{min} \leftarrow \{\mathrm{HC}_{<_{\bar{X}}}(g) \in K[\bar{A}] \mid \mathrm{HT}_{<_{\bar{X}}}(g) \in T_{min},\ g \in G_{\bar{X}}\}$;
    $\{p_1, \ldots, p_t\} \leftarrow \bigcup_{f \in C_{min}}$ Factors$(f)$;
    $\mathcal{N} \leftarrow \Big\{\{p_1\}, \ldots, \{p_t\}\Big\}$;
    $B \leftarrow \{p_1 \cdot \cdots \cdot p_t\}$;
    IF CaseIsZero($(G_{\bar{A}}, B), <_{\bar{A}}$) = $false$ THEN
        $\mathcal{S} \leftarrow \{(G_{\bar{A}}, B, G_{\bar{X}})\}$;
        IF VarietyIsDisjoint($G_{\bar{A}}, B, <_{\bar{A}}$) = $true$ THEN     $\cdots$     (2)
            return ($\mathcal{S}, \emptyset$);
        END IF
    ELSE
        $\mathcal{S} \leftarrow \emptyset$;
    END IF
    return ($\mathcal{S}, \mathcal{N}$);
END
```

Remark 43
We have the following remarks in the algorithm `NewBranches`.

(1) `MHT`$(G, <_{\bar{X}})$ *computes the minimal divisibility basis of* $\mathrm{HT}_{<_{\bar{X}}}(G)$.

(2) *This* `IF` *sentence is the same treatment as (2) of Remark 6 (cf.* $\mathtt{CGS}_1$*).*

We show a new algorithm for computing a CGS using the algorithm `NewBranches`.

Algorithm 44
`NewCGS_Main`$((a_1, \ldots, a_d), F, <_{\bar{A},\bar{X}})$

```
INPUT:  A $d$-tuple $(a_1, \ldots, a_d)$ of natural numbers, a finite set $F \subset K[\bar{A}, \bar{X}]$, a
        term order $<_{\bar{A},\bar{X}}$.
OUTPUT: A set $\mathcal{H}$ of segments $(No, S, T, G)$, where $No \in \mathbb{N}^k$, $S$, $T \subset K[\bar{A}]$, and
        $G$ is the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$.

BEGIN
    $\mathcal{H} \leftarrow \emptyset$;
    $G \leftarrow$ ReducedGB($F, <_{\bar{A},\bar{X}}$);
    IF CaseIsZero($(F \cap K[\bar{A}], G \cap K[\bar{A}]), <_{\bar{A}}$) = $false$ THEN
        $\mathcal{H} \leftarrow \mathcal{H} \cup \Big\{\Big((a_1, \ldots, a_d), F \cap K[\bar{A}], G \cap K[\bar{A}], \{1\}\Big)\Big\}$;
    END IF
    IF $1 \in G$ THEN
        return $\mathcal{H}$;
    END IF
    $(\mathcal{S}, \mathcal{N}) \leftarrow$ NewBranches($G, <_{\bar{A}}, <_{\bar{X}}$);
    IF $\mathcal{S} \neq \emptyset$ THEN
        $\mathcal{H} \leftarrow \mathcal{H} \cup \Big\{\Big((a_1, \ldots, a_d), S, T, G\Big)\Big\}$;        $\Big((S, T, G) \in \mathcal{S}\Big)$
    END IF
    $i = 1$;
    FOR EACH $\{p_1, \ldots, p_t\} \in \mathcal{N}$ DO
        $\mathcal{H} \leftarrow \mathcal{H} \cup$ NewCGS_Main($(a_1, \ldots, a_d, i), G \cup \{p_1, \ldots, p_t\}, <_{\bar{A},\bar{X}}$);
        $i = i + 1$;
    END FOR
    return $\mathcal{H}$;
END
```

Algorithm 45
`NewCGS`$(F, <_{\bar{X}}, <_{\bar{A}})$

```
INPUT:  A finite set $F \subset K[\bar{A}, \bar{X}]$, and term orders $<_{\bar{X}}$ and $<_{\bar{A}}$.
OUTPUT: A set $\mathcal{H}$ of segments $(No, S, T, G)$, where $No \in \mathbb{N}^k$, $S$, $T \subset K[\bar{A}]$, and
        $G$ is the reduced Gröbner basis in $K[\bar{A}, \bar{X}]$.
```

```
BEGIN
    $<_{\bar{A},\bar{X}} \leftarrow$ the elimination order induced by $<_{\bar{A}}$ and $<_{\bar{X}}$ such that $\bar{A} \ll \bar{X}$;
    IF $F \cap K[\bar{A}] \neq \emptyset$ THEN
        $\mathcal{H} \leftarrow \left\{\left((1), \emptyset, F \cap K[\bar{A}], \{1\}\right)\right\}$;
    ELSE
        $\mathcal{H} \leftarrow \emptyset$;
    END IF
    $\mathcal{H} \leftarrow \mathcal{H} \cup$ NewCGS_Main$((1), F, <_{\bar{A},\bar{X}})$;
    return $\mathcal{H}$;
END
```

Theorem 46
For any given finite subset $F \subset (K[\bar{A}])[\bar{X}]$, and term orders $<_{\bar{X}}$ and $<_{\bar{A}}$, the algorithm NewCGS$(F, <_{\bar{X}}, <_{\bar{A}})$ *always terminates, and outputs a CGS of F with respect to $<_{\bar{X}}$.*

Proof We prove the following three claims.

1. The algorithm always terminates.
2. For every segment (No, S, T, G), $\sigma_{\bar{a}}(G)$ is a Gröbner basis of $\langle\sigma_{\bar{a}}(F)\rangle$ with respect to $<_{\bar{X}}$ for any $\bar{a} \in \mathbf{V}(S) \setminus \mathbf{V}(T)$.
3. The union of every parameter space of a segment in the output $\mathcal{H}$ coincides with L^m, that is

$$\bigcup_{(No,S,T,G)\in\mathcal{H}} \mathbf{V}(S) \setminus \mathbf{V}(T) = L^m$$

The claim 1 and 3 are proved in the same way as proof of Theorem 2.3 of [20]. The claim 2 is obvious from Corollary 39. ∎

4.4 Comparisons

In this subsection, we compare our new algorithm NewCGS with Nabeshima's implementation. We have implemented NewCGS in Risa/Asir [14]. The procedure is written in the user language of Risa/Asir. Nabeshima also has implemented NabCGS in the user language of Risa/Asir. In order to make the performance of Gröbner basis computation equal, we revise Nabeshima's implementation so that it uses the new built-in function for Gröbner basis computation (nd_gr_trace) instead of the old one (dp_gr_main).

The details of problems used in this section and the computational environment can be found in Appendix A. The results is shown in Table 6 and 7. In the tables, both our implementation and Nabeshima's one equipped Suzuki-Sato's original algorithm (cf. CGS$_1$). Comparing both the timings of CGS$_1$, we can see a difference of the performance between both CGS$_1$. NabCGS[s],($s \in \mathbb{N}$) means the selection strategy presented in [13]. In the algorithm NabCGS, we remember that the step 4 selects the smallest polynomial in E. In NabCGS[s], the step 4 selects the smallest polynomial in $E_s = \{f \in E \mid \#(M(f)) \leq s\}$ instead of E, where $\#(M(f))$ is the number of monomials in f.

In S_2, the inefficiency of NabCGS, NabCGS[2] or NabCGS[3] is caused by an existence of time-consuming Gröbner basis computation. This computation is made in the step 1–6 of NabCGS. In this computation, a polynomial appeared in the intermediate computations have a large coefficients, that is it consists of 62555 monomials and the sum of bit length of its coefficients is 28230054

Problem	Implementation	Option	Time (sec.)	Segment
S_1	**New**	$\mathtt{CGS}_1$	296.08	27
		`NewCGS`	10.85	5
	Nabeshima	$\mathtt{CGS}_1$	150.97	36
		`NabCGS`	>25min.	***
		`NabCGS[1]`	465.49	36
		`NabCGS[2]`	452.38	36
		`NabCGS[3]`	521.65	36
S_2	**New**	$\mathtt{CGS}_1$	1.912	22
		`NewCGS`	0.956	17
	Nabeshima	$\mathtt{CGS}_1$	1.596	32
		`NabCGS`	>25min.	***
		`NabCGS[1]`	9.469	35
		`NabCGS[2]`	>25min.	***
		`NabCGS[3]`	>25min.	***
S_3	**New**	$\mathtt{CGS}_1$	2.952	17
		`NewCGS`	2.952	17
	Nabeshima	$\mathtt{CGS}_1$	5.812	25
		`NabCGS`	>25min.	***
		`NabCGS[1]`	6.768	27
		`NabCGS[2]`	7.989	42
		`NabCGS[3]`	>25min.	***
S_4	**New**	$\mathtt{CGS}_1$	9.22	33
		`NewCGS`	1.25	19
	Nabeshima	$\mathtt{CGS}_1$	47.20	55
		`NabCGS`	>25min.	***
		`NabCGS[1]`	>25min.	***
		`NabCGS[2]`	>25min.	***
		`NabCGS[3]`	>25min.	***
M_1	**New**	$\mathtt{CGS}_1$	1.316	73
		`NewCGS`	0.704	64
	Nabeshima	$\mathtt{CGS}_1$	1.764	249
		`NabCGS`	>25min.	***
		`NabCGS[1]`	6.028	196
		`NabCGS[2]`	>25min.	***
		`NabCGS[3]`	>25min.	***
M_2	**New**	$\mathtt{CGS}_1$	0.020	9
		`NewCGS`	0.020	9
	Nabeshima	$\mathtt{CGS}_1$	0.008	16
		`NabCGS`	>25min.	***
		`NabCGS[1]`	0.016	12
		`NabCGS[2]`	0.020	13
		`NabCGS[3]`	>25min.	***

Table 6:

Problem	Implementation	Option	Time (sec.)	Segment
N_1	New	$\mathtt{CGS}_1$	0.83	16
		NewCGS	0.22	8
	Nabeshima	$\mathtt{CGS}_1$	93.78	875
		NabCGS	0.11	17
		NabCGS[1]	77.44	621
		NabCGS[2]	0.30	53
		NabCGS[3]	0.13	17
N_2	New	$\mathtt{CGS}_1$	10.41	33
		NewCGS	1.37	20
	Nabeshima	$\mathtt{CGS}_1$	>25min.	***
		NabCGS	>25min.	***
		NabCGS[1]	48.20	458
		NabCGS[2]	>25min.	***
		NabCGS[3]	>25min.	***

Table 7:

(roughly 8500000 digits). On the other hand, In NewCGS, the maximum number of monomials in all polynomials appeared in the intermediate computations is 47 and the maximum bit length of coefficients in all polynomials is 507 (roughly 150 digits). In NabCGS[1], the step 1–6 is called 425 times and the step 1 and 7–10 is called 604 times during the computation, so that roughly speaking, 59% of the computation works as $\mathtt{CGS}_1$.

Our new algorithm is not always faster than the others. One of the reasons of this observation is overhead for manipulations for parameter spaces. For instance, in M_2, this is a reason why each $\mathtt{CGS}_1$ in "New" and "Nabeshima" is the fastest respectively.

5 Conclusion

This paper presents basic manipulations for parameter space appeared in a CGS algorithm based on Suzuki-Sato's and introduces a method for optimizing the property (*2) without an additional variable. Nabeshima's method always uses an additional variable for Gröbner basis computation. This computation often suffers from intermediate coefficient swell. Our new approach does not use an additional variable, and frequency of intermediate coefficient swell is relatively small.

Finally, our new CGS implementation will be found in OpenXM Risa/Asir-contrib.

References

[1] Becker, T. and Weispfenning, V. *Gröbner Bases*. GTM 141, Springer, 1993.

[2] Cox, D., Little, J. and O'Shea, D. *Ideals, Varieties, and Algorithms, Third Edition*. UTM, Springer, 2007.

[3] Gebauer, R. and Möller, H.M. On an installation of Buchberger's algorithm. *J. Symbolic Computation*. Vol. 6/2-3, pp. 275–286. 1988.

[4] Giovini, A., Mora, T., Niesi, G., Robbiano, L. and Traverso, C. “One sugar cube, please” OR Selection strategies in the Buchberger algorithm. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC ’91).* ACM Press, New York, pp. 49–54. 1991.

[5] Kalkbrener, M. On the Stability of Gröbner Bases Under Specializations. *J. Symbolic Computation*. Vol. 24/1, pp. 51–58. 1997.

[6] Kanno, M., Anai, H., Yokoyama, K., and Hara, S. Prametric Optimization in Control Using the Sum of Roots for Parametric Polynomial Spectral Factorization. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC ’07).* ACM Press, New York, pp. 211–218. 2007.

[7] Kurata, Y. and Noro, M. Computation of Discrete Comprehensive Gröbner Bases Using Modular Dynamic Evaluation. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC ’07).* ACM Press, New York, pp. 243–250. 2007.

[8] Manubens, M. and Montes, A. Improving the DISPGB algorithm using the discriminant ideal. *J. Symbolic Computation*. Vol. 41/11, pp. 1245–1263. 2006.

[9] Möller, H.M. On the Construction of Gröbner Bases Using Syzygies. *J. Symbolic Computation*. Vol. 6/2-3, pp. 345–359. 1988.

[10] Montes, A. A new algorithm for discussing Gröbner bases with parameters. *J. Symbolic Computation*. Vol. 33/2, pp. 183–208. 2002.

[11] Nabeshima, K. A Direct Products of Fields Approach to Comprehensive Gröbner Bases over Finite Fields. *Proc. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2005)*, IEEE Computer Society Press, pp. 39–47. 2005.

[12] Nabeshima, K. Reduced Gröbner bases in polynomial rings over a polynomial ring. *International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2006)* (Wang, D. and Zheng, Z., editors), pp. 15–32. 2006.

[13] Nabeshima, K. A Speed-Up of the Algorithm for Computing Comprehensive Gröbner Systems. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC ’07).* ACM Press, New York, pp. 299–306. 2007.

[14] Noro, M. et al. A Computer Algebra System Risa/Asir.
`http://www.math.kobe-u.ac.jp/Asir/asir.html`. 2009.

[15] Sato, Y., Suzuki, A and Nabeshima, K. Discrete Comprehensive Gröbner Bases II. *Computer Mathematics, Proc. 6th Asian Symposium (ASCM 2003)*, Lecture Notes Series on Computing Vol. 10, World Scientific, pp. 240–247. 2003.

[16] Sato, Y., Suzuki, A and Nabeshima, K. ACGB on Varieties. *Proc. 6th International Workshop on Computer Algebra in Scientific Computing (CASC 2003)*, pp. 313–318. 2003.

[17] Sato, Y. Stability of Gröbner bases and ACGB. *Proc. Algorithmic Algebra and Logic 2005 (Conference in Honor of the 60th Birthday of Volker Weispfenning)*, Books on Demand GmbH, pp. 223–228. 2005.

[18] Shinohara, N. Parametric Polynomial Spectral Factorization. *Bulletin of the Japan Society for Symbolic and Algebraic Computation (in Japanese).* Vol.16/1, pp. 3–19. 2009.

[19] Suzuki, A. and Sato, Y. An alternative approach to Comprehensive Gröbner Bases. *J. Symbolic Computation*. Vol. 36/3-4, pp. 649–667. 2003.

[20] Suzuki, A. and Sato, Y. A Simple Algorithm to Compute Comprehensive Gröbner Bases Using Gröbner Bases. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC '06)*, ACM Press, New York, pp. 326–331. 2006.

[21] Traverso, C. Gröbner trace algorithms. *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC '88)*, Springer-Verlag, London, pp. 125–138. 1988.

[22] Weispfenning, V. Comprehensive Gröbner bases. *J. Symbolic Computation*. Vol. 14/1, pp. 1–29. 1992.

[23] Weispfenning, V. Canonical comprehensive Gröbner bases. *J. Symbolic Computation*. Vol. 36/3-4, pp. 669–683. 2003.

A Problems for Comparisons

In this paper, we use following problems for comparisons. All measures are taken on a Linux PC with Intel Xeon X5470 at 3.33GHz and 32GB of memory.

- $\boldsymbol{S_1}$

  $$\{X^5 - a, Y^6 - b, X + Y - Z\}$$

  In the case, X, Y, Z and a, b are variables and parameters respectively, and the term oder is the lexicographic such that $X > Y > Z$.

- $\boldsymbol{S_2}$

  $$\{P,\ Q,\ (X_1 - X_2)^2 + (Y_1 - Y_2)^2 - S,\ \frac{\partial P}{\partial X_1}\frac{\partial Q}{\partial Y_2} - \frac{\partial P}{\partial Y_1}\frac{\partial Q}{\partial X_2},\ \frac{\partial P}{\partial X_1}(Y_1 - Y_2) - \frac{\partial P}{\partial Y_1}(X_1 - X_2)\}$$

  with $P = aX_1^2 + bY_1$ and $Q = cY_2^2 + dX_2$. In the case, X_1, X_2, Y_1, Y_2, S are variables and the term order is the lexicographic such that $X_1 > X_2 > Y_1 > Y_2 > S$.

- $\boldsymbol{S_3}$
  The same polynomial set as S_2 with $P = X_1^2 + Y_1^2 + a$ and $Q = Y_2 - bX_2^2 + c$. The others conditions are similar to S_2.

- $\boldsymbol{S_4}$

  $$\{P - Z,\ X^2 + Y^2 + Z^2 - S,\ X + \frac{\partial P}{\partial X}Z,\ Y + \frac{\partial P}{\partial Y}Z\}$$

  with $P = (X - a)^2 + bY^2 + b$. In the case, X, Y, Z, S and a, b are variables and parameters respectively, and the term order is the lexicographic such that $X > Y > Z > S$.

- $\boldsymbol{S_5}$
  The same polynomial set as S_4 with $P = (X - a)^2 + bY^2 + c$. In the case, a, b, c are parameters, and the others conditions are similar to S_4.

- $\boldsymbol{S_6}$
  The same polynomial set as S_4 with $P = (X - a)^2 + bY^2 + a^2 - b$. The others conditions are similar to S_4.

- $\boldsymbol{M_1}$

  $$\{a + dS_1,\ b - dC_1,\ l_2C_2 + l_3C_3 - d,\ l_2S_2 + l_3S_3 - c,\ S_1^2 + C_1^2 - 1,\ S_2^2 + C_2^2 - 1,\ S_3^2 + C_3^2 - 1\}$$

  In the case, $S_1, C_1, S_2, C_2, S_3, C_3$ and a, b, c, d are variables and parameters respectively, and the term order is the lexicographic such that $S_1 > C_1 > S_2 > C_2 > S_3 > C_3$.

- $\boldsymbol{M_2}$

  $$\{aX^2Y + a + 3b^2,\ a(b - c)XY + abX + 5c\}$$

  In the case, X, Y and a, b, c are variables and parameters respectively, and the term order is the lexicographic such that $X > Y$.

- $\boldsymbol{N_1}$

  $$\{X^4 + aX^3 + bX^2 + cX + d,\ 4X^3 + 3aX^2 + 2bX + c\}$$

  In the case, X is a variable and a, b, c, d are parameters.

- $\boldsymbol{N_2}$

  $$\{aX^2 + bY,\ cW^2 + Z,\ (X - Z)^2 + (Y - W)^2,\ 2dXW - 2bY\}$$

  In the case, X, Y, Z, W and a, b, c, d are variables and parameters respectively, and the term order is the lexicographic such that $X > Y > Z > W$.

Each set lies in a polynomial ring over $\mathbb{Q}$. These problems are taken from literature: The polynomial sets S_1, S_2, S_3, S_4, S_5 and S_6 are taken from Section 4 of Suzuki and Sato [20], M_1 and M_2 are taken from Section 5 of Manubens and Montes [8], N_1 and N_2 are taken from Section 5 in Nabeshima [13].