An Efficient Algorithm for Computing Parametric Multivariate Polynomial GCD

Deepak Kapur
Department of Computer Science
University of New Mexico
Albuquerque, NM, USA
Kapur@cs.unm.edu

Dong Lu
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
donglu@amss.ac.cn

Michael Monagan
Department of Mathematics
Simon Fraser University
Burnaby, B.C., V5A 1S6, Canada
mmonagan@cecm.sfu.ca

Yao Sun
SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
sunyao@iie.ac.cn

Dingkang Wang
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
dwang@mmrc.iss.ac.cn

ABSTRACT

A new efficient algorithm for computing a parametric greatest common divisor (GCD) of parametric multivariate polynomials over $k[\vec{u}][\vec{x}]$ is presented. The algorithm is based on a well-known simple insight that the GCD of two multivariate polynomials (non-parametric as well as parametric) can be extracted using the generator of the quotient ideal of a polynomial with respect to the second polynomial. And, further, this generator can be obtained by computing a minimal Gröbner basis of the quotient ideal. The main attraction of this idea is that it generalizes to the parametric case for which a comprehensive Gröbner basis is constructed for the parametric quotient ideal. It is proved that in a minimal comprehensive Gröbner system of a parametric quotient ideal, each branch of specializations corresponds to a principal parametric ideal with a single generator. Using this generator, the parametric GCD of that branch is obtained by division. This algorithm does not need to consider whether parametric polynomials are primitive w.r.t. the main variable. This is in sharp contrast to two algorithms recently proposed by Nagasaka (ISSAC, 2017). The resulting algorithm is not only conceptually simple to understand but is considerably efficient. The proposed algorithm and both of Nagasaka's algorithms have been implemented in Singular, and their performance is compared on a number of examples. For more than two polynomials, this process can be repeated by considering pairs of polynomials; the efficiency in that case becomes even more evident.

ISSAC'18, July 16–19, 2018, New York, NY, USA

ACM ISBN 978-1-4503-5550-6/18/07…$15.00
https://doi.org/10.1145/3208976.3208980

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic algorithms**; **Algebraic algorithms**;

KEYWORDS

Parametric multivariate polynomial, Parametric GCD, Minimal comprehensive Gröbner system, Quotient ideal

ACM Reference Format:
Deepak Kapur, Dong Lu, Michael Monagan, Yao Sun, and Dingkang Wang. 2018. An Efficient Algorithm for Computing Parametric Multivariate Polynomial GCD. In *ISSAC'18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3208976.3208980

1 INTRODUCTION

Multivariate polynomial GCD computation is one of the most important operations in computer algebra as it is used in

many algorithms and applications. The problem has been extensively investigated and numerous algorithms have been developed to compute the GCD efficiently beyond Euclid's algorithm using division for univariate polynomials and its extension to multivariate polynomials using pseudo-division. Brown's modular GCD algorithm from [3] was the first GCD algorithm that avoided intermediate expression swell. For sparse polynomials Moses and Yun in [14] developed the EZ GCD algorithm which is based on Hensel lifting. Zippel's sparse modular GCD algorithm [21] uses sparse interpolation. It is currently used in Maple, Magma, and Mathematica. We mention also algorithms of Gianni *et al.* [7] and Sasaki *et al.* [18] which compute a GCD from a Gröbner basis. For sparse multivariate polynomials, Sanuki *et al.* [17] utilized Extended Hensel Construction to compute GCD and found that their algorithm to be comparable in performance to Maple's GCD routine.

Using the concept of parametric polynomials, there have also been many publications studying how to compute the GCD of parametric polynomials. Abramov and Kvashenko [1] used the subresultant chain to compute a parametric univariate polynomial GCD. Ayad [2] presented three algorithms based on parametrization of the Gaussian elimination procedure to compute GCD of a finite set of parametric univariate polynomials. At ISSAC 2017, Nagasaka [16] extended the ideas of Gianni and Trager [7] as well as Sasaki *et al.* [18] to polynomials with parameters for computing the GCD of parametric multivariate polynomials. The main tool used in Nagasaka's algorithms is the comprehensive Gröbner system which is the parametric extension of Gröbner basis, introduced by Weispfenning [20] (and independently by [8] as parametric Gröbner basis) and was improved by Suzuki *et al.* [19], Kapur *et al.* [9, 10] and Nabeshima [15]. In Nagasaka's paper, the algorithms to compute the GCD of parametric multivariate polynomials need to consider whether parametric polynomials are primitive w.r.t. x_1 under different parametric constraints. Moreover, he had to construct an ideal that is maximal for any specialization based on extending Gianni and Trager's results [7]. Both of these steps in his algorithms can be extremely time consuming.

This paper presents a new efficient algorithm for the GCD computation of parametric multivariate polynomials. The main idea of the new algorithm comes from computing a minimal Gröbner basis of a non-parametric colon ideal of two polynomials in the nonparametric case. Let k be a field, $k[\vec{x}]$ be the polynomial ring in the variables $\vec{x} = \{x_1, \ldots, x_n\}$. Assume that f and g are two nonzero polynomials in $k[\vec{x}]$. It is easy to see that the minimal Gröbner basis of the quotient ideal $\langle f \rangle : g$ has only one polynomial h. Then, the GCD of f and g is $\frac{f}{h}$. Most importantly, this construction extends to the case of parametric polynomials in which a Gröbner basis computation of the quotient ideal is replaced by comprehensive Gröbner system construction for parametric polynomials. To compute the GCD of more than two parametric polynomials, the above method is repeated much as in the case of computing the GCD of a family of numbers. Compared with Nagasaka's algorithms, the new algorithm has two advantages: there is no need to check whether parametric polynomials are primitive w.r.t. x_1 in each iteration, and further, it is guaranteed that a parametric polynomial f is divisible by the result in the quotient ideal.These merits make the proposed algorithm more efficient.

This paper is organized as follows. In Section 2, we provide background about the GCD and the comprehensive Gröbner computations for parametric multivariate polynomials. Nagasaka's algorithms are reviewed in Section 3. The proposed algorithm is presented in Section 4. To provide intuition and make the presentation simple, we first briefly discuss how the GCD of non-parametric polynomials can be computed using a minimal Gröbner basis of a quotient ideal. This is followed by extending this method to parametric polynomials. The new algorithm is presented. In Section 5, a non-trivial example is given to illustrate the key steps of the proposed algorithm. This is followed by some remarks about computing the GCD of a system of parametric polynomials in Section 6. Experimental data and a comparison with Nagasaka's algorithms are presented in Section 7. We end with some concluding remarks in Section 8.

2 PRELIMINARIES

Let k be a number field, $\bar{k}$ be the algebraic closure of k, $k[\vec{x}]$ be the polynomial ring in the variables $\vec{x} = \{x_1, \ldots, x_n\}$, $k[\vec{u}]$ be the parametric polynomial ring in the parameters $\vec{u} = \{u_1, \ldots, u_m\}$, and $k[\vec{u}][\vec{x}]$ be the polynomial ring over the parameter ring $k[\vec{u}]$ in $\vec{x}$. It is assumed that $\vec{x} \cap \vec{u} = \emptyset$, i.e., $\vec{x}$ and $\vec{u}$ are disjoint sets. In some cases, we abbreviate $\{x_i, x_{i+1}, \ldots, x_n\}$ to $\vec{x}_i$ $(2 \leq i \leq n)$.

We introduce some notation and definitions for non parametric multivariate polynomials. Two polynomials $f(\vec{x})$, $g(\vec{x}) \in k[\vec{x}]$ are associates if $\exists c \in \bar{k}$ such that $f(\vec{x}) = c\, g(\vec{x})$; we denote this equivalence relation by $f(\vec{x}) \sim g(\vec{x})$. For a polynomial $f \in k[\vec{x}]$, the leading term, leading coefficient, leading monomial and the total degree of f w.r.t. a monomial order $\prec$ are denoted by $\mathrm{lt}(f)$, $\mathrm{lc}(f)$, $\mathrm{lm}(f)$ and $\mathrm{tdeg}(f)$ respectively. We have $\mathrm{lt}(f) = \mathrm{lc}(f) \cdot \mathrm{lm}(f)$. The ideal in $k[\vec{x}]$, generated by $f_1, \ldots, f_s$, is denoted by $\langle f_1, \ldots, f_s \rangle$.

Definition 2.1. Let $f_1, \ldots, f_s \in k[\vec{x}]$. Then $h \in k[\vec{x}]$ is called a **greatest common divisor** (GCD) of $f_1, \ldots, f_s$, denoted $h = \gcd(f_1, \ldots, f_s)$, if

(1) $\forall i$ $(1 \leq i \leq s)$, h divides f_i and
(2) if g is any polynomial which divides $f_1, \ldots, f_s$, then g divides h.

Particularly, we define $\gcd(f_1, \ldots, f_s) = \gcd(f_2, \ldots, f_s)$ if $f_1 = 0$, and $\gcd(0, 0) = 0$, for convenience.

A GCD of polynomials is defined modulo associates. For any given polynomials $f_1, \ldots, f_s \in k[\vec{x}]$, there exist $\bar{f}_1, \ldots, \bar{f}_s \in k[\vec{x}]$ such that

$$f_i = \gcd(f_1, \ldots, f_s) \cdot \bar{f}_i, \quad (1 \leq i \leq s)$$

then $\bar{f}_1, \ldots, \bar{f}_s$ are called the **cofactors** of $f_1, \ldots, f_s$.

Definition 2.2. Let $f \in k[\vec{x}]$. f is said to be primitive w.r.t. x_1 if it is primitive as a polynomial in $k[\vec{x}_2][x_1]$, that is, its coefficients in $k[\vec{x}_2]$ are co-prime.

Definition 2.3. Let g be a nonzero multivariate polynomial and I is an ideal in $k[\vec{x}]$. The set

$$I : g = \{f \in k[\vec{x}] : fg \in I\}$$

is called the **quotient ideal** (or **colon ideal**) of I divided by g.

For example, in $k[x_1, x_2, x_3]$ we have $\langle x_1x_3, x_2x_3\rangle : x_3 = \{f \in k[x_1, x_2, x_3] : x_3f \in \langle x_1x_3, x_2x_3\rangle\} = \{f \in k[x_1, x_2, x_3] : x_3f = Ax_1x_3 + Bx_2x_3\} = \{f \in k[x_1, x_2, x_3] : f = Ax_1 + Bx_2\} = \langle x_1, x_2\rangle$, where $A, B \in k[x_1, x_2, x_3]$.

Definition 2.4. A **minimal Gröbner basis** for a polynomial ideal $I \subseteq k[\vec{x}]$ is a Gröbner basis G for I such that:

(1) $\mathrm{lc}(p) = 1$ for all $p \in G$;
(2) $\mathrm{lm}(p) \notin \langle \mathrm{lm}(G - \{p\})\rangle$ for all $p \in G$.

Next we introduce some notation and definitions for parametric multivariate polynomials. For a polynomial $g \in k[\vec{u}][\vec{x}]$, the leading term, leading coefficient, leading monomial and total degree of g w.r.t. the monomial order $\prec_{\vec{x}}$ are denoted by $\mathrm{lt}_{\vec{x}}(g)$, $\mathrm{lc}_{\vec{x}}(g)$, $\mathrm{lm}_{\vec{x}}(g)$ and $\mathrm{tdeg}_{\vec{x}}(g)$ respectively. If $g \in k[\vec{x}]$ or $g \in k[\vec{u}][\vec{x}]$, we use $\mathrm{lc}_{x_i}(g)$ to denote the leading coefficient of g w.r.t. x_i.

A **specialization** of $k[\vec{u}]$ is a homomorphism $\sigma : k[\vec{u}] \to \bar{k}$. In this paper, we only consider the specializations induced by the elements in $\bar{k}^m$. That is, for $\vec{a} \in \bar{k}^m$, the induced specialization $\sigma_{\vec{a}}$ is defined as

$$\sigma_{\vec{a}} : \varphi \to \varphi(\vec{a}),$$

where $\varphi \in k[\vec{u}]$. Every specialization σ: $k[\vec{u}] \to \bar{k}$ extends canonically to a specialization σ: $k[\vec{u}][\vec{x}] \to \bar{k}[\vec{x}]$ by applying σ coefficient-wise.

For an ideal $E \subset k[\vec{u}]$, the variety defined by E in $\bar{k}^m$ is denoted by $\mathbf{V}(E) = \{\vec{a} \in \bar{k}^m \mid f(\vec{a}) = 0 \ \textit{for all} \ f \in E\}$. In this paper, an **algebraically constructible set** A always has the form: $A = \mathbf{V}(E) \setminus \mathbf{V}(N)$, where E, N are ideals in $k[\vec{u}]$. It is easy to see that the algebraically constructible set A is not empty by ensuring that at least one $f \in N$ is not in the radical of E. Let $V \subset \bar{k}^m$ be a variety. Let $\mathbf{I}(V) = \{f \in k[\vec{u}] \mid f(\vec{a}) = 0 \ \textit{for all} \ \vec{a} \in V\}$. According to Corollary 3 ([4], page 176), $\mathbf{I}(V)$ is a radical ideal.

For a parametric polynomial system, the comprehensive Gröbner system and minimal comprehensive Gröbner system are given below.

Definition 2.5. Let F be a set in $k[\vec{u}][\vec{x}]$, $A_1, \ldots, A_l$ be algebraically constructible subsets of $\bar{k}^m$, $G_1, \ldots, G_l$ be subsets of $k[\vec{u}][\vec{x}]$, and S be a subset of $\bar{k}^m$ such that $S \subset A_1 \cup \cdots \cup A_l$. A finite set $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ is called a **comprehensive Gröbner system** (CGS) on S for F if $\sigma_{\vec{a}}(G_i)$ is a Gröbner basis for for the ideal $\langle \sigma_{\vec{a}}(F)\rangle \subset \bar{k}[\vec{x}]$ for $\vec{a} \in A_i$ and $i = 1, \ldots, l$. Each (A_i, G_i) is called a branch of $\mathcal{G}$. In particular, if $S = \bar{k}^m$, then $\mathcal{G}$ is called a comprehensive Gröbner system for F.

Definition 2.6. A comprehensive Gröbner system $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ on S for F is said to be **minimal**, if for every $i = 1, \ldots, l$,

(1) $A_i \neq \emptyset$, and furthermore, for each $j = 1, \ldots, l$, $A_i \cap A_j = \emptyset$ whenever $i \neq j$;
(2) $\sigma_{\vec{a}}(G_i)$ is a minimal Gröbner basis for $\langle \sigma_{\vec{a}}(F)\rangle \subset \bar{k}[\vec{x}]$ for $\vec{a} \in A_i$;
(3) for each $g \in G_i$, $\sigma_{\vec{a}}(\mathrm{lc}_{\vec{x}}(g)) \neq 0$ for any $\vec{a} \in A_i$.

Abramov and Kvashenko [1] studied the parametric GCD of univariate polynomials with one parameter. The definition of parametric GCD (one parameter) can be easily extended to the case m ($m \geq 1$).

Definition 2.7. For $F = \{f_1, \ldots, f_s\} \subset k[\vec{u}][\vec{x}]$ and $S \subset \bar{k}^m$, we call $\{(A_1, g_1), \ldots, (A_r, g_r)\}$ a **parametric GCD** of F on S, if for every $i = 1, \ldots, r$, $\sigma_{\vec{a}}(g_i)$ is a GCD of $\sigma_{\vec{a}}(F)$ for any specialization $\vec{a} \in A_i$, where $A_1, \ldots, A_r \subset \bar{k}^m$ are algebraically constructible sets and $S = \cup_{i=1}^{r} A_i$, $g_1, \ldots, g_r \in k[\vec{u}][\vec{x}]$. If $S = \bar{k}^m$, we simply call it a parametric GCD of F.

3 NAGASAKA'S ALGORITHMS

As stated in the introduction, the GCD of polynomials have been extensively studied in the literature because of the enormous importance of this operation in many symbolic computation algorithms and applications; see [3, 14, 21] for instance. The main issue in the GCD computation is that of intermediate expression swell as analyzed in Knuth vol. 2.

Gianni *et al.* [7] and Sasaki *et al.* [18] studied the GCD of polynomials by computing a Gröbner basis instead of using the Euclidean algorithm. Nagasaka [16] extended their results to polynomials with parameters and proposed two algorithms to compute the parametric GCD of parametric polynomials. In the following, we provide an overview of Nagasaka's algorithms and illustrate their shortcomings; more details about the algorithms can be found in [16].

3.1 Extending Gianni and Trager's Algorithm

Nagasaka extended Proposition 2 in [7] to state:

LEMMA 3.1. *Let $f_1, \ldots, f_s, g \in k[\vec{x}]$ be primitive w.r.t. x_1, J be a maximal ideal in $k[\vec{x}_2]$ such that $\langle f_1, \ldots, f_s, J\rangle \ni 1$ and $\langle \mathrm{lc}_{x_1}(gf_i), J\rangle \ni 1$ for some i. Let G be a Gröbner basis for $\langle gf_1, \ldots, gf_s, J^r\rangle$ w.r.t. any total degree order. Then, the polynomial $\bar{g}$ in G of least total degree is an associate of g if the least total degree of the elements in J^r is larger than* $\mathrm{tdeg}(g)^2$.

Nagasaka further extended Lemma 3.1 to the case of parametric polynomials for which additional conditions on the ideal $J \subset k[\vec{u}][\vec{x}_2]$ for each specialization $\vec{\omega} \in \bar{k}^m$ must be satisfied:

(1) $\sigma_{\vec{\omega}}(f_1), \ldots, \sigma_{\vec{\omega}}(f_s)$ are primitive w.r.t. x_1;
(2) $\sigma_{\vec{\omega}}(J)$ is a maximal ideal in $k[\vec{x}_2]$;
(3) $\langle \mathrm{lc}_{x_1}(\sigma_{\vec{\omega}}(f_i)), \sigma_{\vec{\omega}}(J)\rangle \ni 1$ for some i;
(4) $\langle \bar{f}_1, \ldots, \bar{f}_s, \sigma_{\vec{\omega}}(J)\rangle \ni 1$, where each $\bar{f}_i \in k[\vec{x}]$ is the cofactor of $\sigma_{\vec{\omega}}(f_i)$.

To satisfy these conditions, the parametric space $\bar{k}^m$ needs to be decomposed into branches such that F and each J have the following properties.

Definition 3.2. For the given $F = \{f_1, \ldots, f_s\} \subset k[\vec{u}][\vec{x}]$ with $S \subset \bar{k}^m$ and $J \subset k[\vec{u}][\vec{x}_2]$, we introduce the following.

(1) F is said to be S**-primitive** if for any specialization $\vec{\omega} \in S$, $\sigma_{\vec{\omega}}(f_1), \ldots, \sigma_{\vec{\omega}}(f_s)$ are primitive w.r.t. x_1;
(2) J is said to be S**-maximal** if for any specialization $\vec{\omega} \in S$, $\sigma_{\vec{\omega}}(J)$ is a maximal ideal in $k[\vec{x}_2]$;
(3) F is said to be S**-nonvanishlc** if for any specialization $\vec{\omega} \in S$, $\mathrm{lc}_{x_1}(\sigma_{\vec{\omega}}(f_i)) = \sigma_{\vec{\omega}}(\mathrm{lc}_{x_1}(f_i))$ for each i;
(4) F is said to be S**-nondegenerate** if for any specialization $\vec{\omega} \in S$, $\langle \mathrm{lc}_{x_1}(\sigma_{\vec{\omega}}(f_i)), \sigma_{\vec{\omega}}(J)\rangle \ni 1$ for some i;
(5) J is said to be S**-luckyprime** if for any specialization $\vec{\omega} \in S$, $\langle \bar{f}_1, \ldots, \bar{f}_s, \sigma_{\vec{\omega}}(J)\rangle \ni 1$, where each $\bar{f}_i \in k[\vec{x}]$ is the cofactor of $\sigma_{\vec{\omega}}(f_i)$.

Under these conditions, Nagasaka proposed an algorithm to compute the parametric GCD by combining Lemma 3.1 and Definition 3.2, which we call henceforth, the Nagasaka-GT algorithm.

Step 1: compute the S-primitive part of F w.r.t. x_1;
Step 2: decompose S such that F is S-nonvanishlc;
Step 3: construct a maximal ideal $J \subset k[\vec{x}_2]$ such that F is S-nondegenerate;
Step 4: compute a minimal CGS for $\langle F \cup J^r\rangle$ on S, where r satisfies the degree condition of Lemma 3.1;
Step 5: check whether J is a S-luckyprime, if not, return to the Step 3;
Step 6: obtain the parametric GCD of F.

As the reader will notice, the above conditions are complicated and not easy to appreciate. Further, while implementing the Nagasaka-GT algorithm in Singular, we discovered the following shortcomings. Without any loss of generality, we assume in the following examples that $\vec{u} = \{a, b\}$, $\vec{x} = \{x_1, x_2, x_3\}$ and $S = \mathbb{C}$ and consider the lexicographic order with $x_1 \succ x_2 \succ x_3$.

(1) In Step 1, Nagasaka needs to call this algorithm repeatedly to compute the primitive part of each parametric polynomial. For example, we want to compute the primitive part of $f = (1-a)x_1^3x_2^2 + a(b-1)x_1^3x_2x_3 + (a^2 - a)x_1x_2^2 + (a-b)x_1x_3 + (a-1)x_2^2 + a(b-1)x_2x_3^3 + ax_3$ w.r.t. x_1 on $\mathbb{C}$. We must know the parametric GCD of coefficients of f w.r.t. x_1 on $\mathbb{C}$, i.e., we have to call this algorithm to compute the parametric GCD of f_{11}, f_{12}, f_{13}, where $f_{11} = (1-a)x_2^2 + a(b-1)x_2x_3$, $f_{12} = (a^2-a)x_2^2 + (a-b)x_3$ and $f_{13} = (a-1)x_2^2 + a(b-1)x_2x_3^3 + ax_3$. As the number of variables increases, this becomes more and more tedious, resulting in computational inefficiency.
(2) Step 2 is not necessary. Step 1 has ensured that the leading coefficient of f w.r.t. x_1 is not zero on each branch S_j, i.e, $\mathrm{lc}_{x_1}(\sigma_{\vec{\omega}}(f)) = \sigma_{\vec{\omega}}(\mathrm{lc}_{x_1}(f))$ for any specialization $\vec{\omega} \in S_j$. Therefore, Step 2 can be removed.
(3) If the parameter space S is divided into many small areas, more and more maximal ideals need to be constructed in Step 3. Although Nagasaka proved that a maximal ideal $J \subset k[x_2, x_3]$ which is S-nondegenerate and S-luckyprime can be constructed in a finite number of steps, we do not know how much time it takes to construct so many maximal ideals.
(4) We need to estimate the value of r in Step 4. Since $J = \langle x_2 - c_2, x_3 - c_3\rangle$ and we do not know the polynomial g in Lemma 3.1, we often let $r := \min\{\mathrm{tdeg}_{\vec{x}}(f_i)^2 + 1 \mid f_i \in F\}$. For instance, let $F = \{f_1, f_2\}$, where $f_1 = ax_1^3x_2^2x_3 + (1-b)(x_2^2 + x_3)$, $f_2 = (1-a)x_1^3x_2^2x_3 + b(x_2^2 + x_3)$. Then, $r = 37$. There are two problems: First, it will take more time to compute the minimal CGS of $\langle F \cup J^{37}\rangle$ which sometimes does not terminate. Second, since $c_2, c_3 \in \mathbb{C}$ are chosen randomly, sometimes c_i^{37} is a large integer.

3.2 Extending Sasaki and Suzuki's Algorithm

Sasaki and Suzuki [18] also used a Gröbner basis construction to compute the GCD of polynomials, by improving upon Gianni and Trager's results. They obtained a similar theorem, but did not need to use a maximal ideal J.

THEOREM 3.3. (Theorem 1 in [18]) *Let $f_1, f_2 \in k[\vec{x}]$ be primitive w.r.t. x_1, and G be the Gröbner basis w.r.t. any block order such that $x_1 \succ \vec{x}_2$ for $\langle f_1, f_2\rangle$. Then, there exists a polynomial $h \in k[\vec{x}_2]$ such that $\bar{g} = h \cdot \gcd(f_1, f_2)$, where $\bar{g}$ is the polynomial in G of least degree in x_1.*

Using the insight in Theorem 3.3, Nagasaka proposed a second algorithm (henceforth called, Nagasaka-SS algorithm).

Step 1: compute an S-primitive decomposition;
Step 2: compute a minimal CGS;
Step 3: compute a parametric GCD of coefficients of the candidate factor;
Step 4: compute the primitive part in each branch.

There are similarities between the Nagasaka-SS algorithm and Nagasaka-GT algorithm which are also sources of inefficiency: both need to compute S-primitive decompositions and make recursive calls to compute the parametric GCD of coefficients of polynomials. The Nagasaka-SS algorithm has been observed to be more efficient than the Nagasaka-GT algorithm, since the Nagasaka-SS algorithm does not need to construct many maximal ideals and only needs to compute the minimal CGS of $\langle F\rangle$ rather than $\langle F \cup J^r\rangle$.

4 THE PROPOSED ALGORITHM

We propose a new algorithm for computing the GCD of two parametric multivariate polynomials. To present the key ideas, we first give the algorithm for the non-parametric case and then we extend it to the parametric case. The key idea is well-known: compute the cofactor by computing the quotient ideal of one polynomial with respect to the other polynomial.

This quotient ideal is known to be principal and has a single generator which can be computed by a single minimal Gröbner basis computation. This generator, which is the cofactor of the first polynomial, is used to obtain the GCD by dividing the polynomial by its cofactor. For the parametric case, a minimal comprehensive Gröbner system of a module is computed, leading to multiple branches for different specializations; for each branch, the generator is used to obtain the GCD for the associated parametric specializations.

To experimentally compare the proposed algorithm with both of Nagasaka's algorithms, we have implemented them all in Singular on a single platform so that their comparative performance can be fairly analyzed (Section 7).

4.1 GCD for non-parametric polynomials

As stated above, there are many well-known algorithms for computing the GCD of multivariate polynomials starting from Euclid's algorithm improved by Collins using reduced polynomial remainder sequences (PRS), Brown and Traub and Brown's subresultant PRS with EZGCD algorithm in MACSYMA for multivariate polynomials in general and Zippel's algorithm based on sparse interpolation which is more efficient for sparse polynomials. There are also algorithms based on Gröbner basis computations. We are, however, interested in algorithms which generalize to parametric polynomial systems. To our knowledge, algorithms based on the Euclidean division algorithm (really pseudo-division in case of multivariate polynomials) and hence, Gröbner bases are most suited to generalize to parametric polynomial systems.

THEOREM 4.1. *Consider two polynomials $f_1, f_2 \in k[\vec{x}]\setminus\{0\}$ such that $f_1 = d \cdot \bar{f}_1$ and $f_2 = d \cdot \bar{f}_2$, where $d = \gcd(f_1, f_2)$ and $\gcd(\bar{f}_1, \bar{f}_2) = 1$. Then, $\langle \bar{f}_1\rangle = \langle f_1\rangle : f_2$, $\langle \bar{f}_2\rangle = \langle f_2\rangle : f_1$.*

Theorem 4.1 implies that $\langle f_1\rangle : f_2$ is a principal ideal. A minimal Gröbner basis G of $\langle f_1\rangle : f_2$ w.r.t. a monomial order $\prec$ is $\{g\}$ such that $\gcd(f_1, f_2) = f_1/g$. Depending upon the structure of f_1, f_2 and the degree of their GCD relative to the degrees of f_1, f_2, computing $\langle f_1\rangle : f_2$ or $\langle f_2\rangle : f_1$ can have varied performance.

A quotient ideal can be constructed using ideal intersection [4] (pp.183–197) which involves introducing a new variable z to construct a new ideal $J = \langle zf_1, (1-z)f_2\rangle \subset k[z, \vec{x}]$. Given that the complexity of Gröbner basis computations is heavily influenced by the number of variables and the degrees of the polynomials, we believe that computations over modules are likely to be more efficient (Chapter 5, [5]).

Let $\mathbf{e_1} = (1, 0)$ and $\mathbf{e_2} = (0, 1)$. Then $\{\mathbf{e_1}, \mathbf{e_2}\}$ is a free basis of $(k[\vec{x}])^2$. For any element $\vec{v}$ in $(k[\vec{x}])^2$, it can be expressed as $\vec{v} = h_1 \cdot \mathbf{e_1} + h_2 \cdot \mathbf{e_2}$ where $h_1, h_2 \in k[\vec{x}]$. For any submodule W of $(k[\vec{x}])^2$, we can also compute the Gröbner basis of W. The module case follows the ideal case almost exactly. However, we need to extend the notion of monomial orders to the free module $(k[\vec{x}])^2$. Let $\prec$ be a monomial order on $k[\vec{x}]$, then extend $\prec$ to the $(k[\vec{x}])^2$ in a position over term fashion with $\mathbf{e_2} \prec \mathbf{e_1}$.

THEOREM 4.2. *Let f_1, f_2 be two polynomials in $k[\vec{x}] \setminus \{0\}$ and $\prec$ be a monomial order on $k[\vec{x}]$. Suppose $W \subset (k[\vec{x}])^2$ is a $k[\vec{x}]$-module generated by $\{f_1 \cdot \mathbf{e_1}, f_2 \cdot \mathbf{e_1} - \mathbf{e_2}\}$ and G is a* **minimal** *Gröbner basis of W w.r.t. an order extended from $\prec$ in a position over term fashion with $\mathbf{e_2} \prec \mathbf{e_1}$. Then there exists a unique polynomial $g \in k[\vec{x}] \setminus \{0\}$ such that $g \cdot \mathbf{e_2} \in G$ and $\langle g\rangle = \langle f_1\rangle : f_2$.*

PROOF. Let $H = \{h \in k[\vec{x}] \mid h \cdot \mathbf{e_2} \in G\}$. We prove $\langle H\rangle = \langle f_1\rangle : f_2$ below.

We first show $\langle f_1\rangle : f_2 \subset \langle H\rangle$. For any given polynomial p in $\langle f_1\rangle : f_2$, there exists a polynomial $q \in k[\vec{x}]$ such that $pf_2 = qf_1$. Then, $p \cdot \mathbf{e_2} = q(f_1 \cdot \mathbf{e_1}) - p(f_2 \cdot \mathbf{e_1} - \mathbf{e_2})$ implies $p \cdot \mathbf{e_2} \in W$. Since G is a minimal Gröbner basis of W, it follows that $p \in \langle H\rangle$.

For the converse, suppose $h \in \langle H\rangle$. Then there exist polynomials $g_1, \ldots, g_s, p_1, \ldots, p_s \in k[\vec{x}]$ such that $h = \sum_{i=1}^{s} (p_i g_i)$ and $g_i \cdot \mathbf{e_2} \in G$ for $1 \le i \le s$. Thus, we have $h \cdot \mathbf{e_2} \in \langle G\rangle$, which implies $h \cdot \mathbf{e_2} = A(f_1 \cdot \mathbf{e_1}) + B(f_2 \cdot \mathbf{e_1} - \mathbf{e_2})$ for some polynomials $A, B \in k[\vec{x}]$. From this equation we can obtain the following equations.

$$\begin{cases} 0 = Af_1 + Bf_2, \\ h = -B. \end{cases}$$

Therefore, we have $h \in \langle f_1\rangle : f_2$.

In sum, we have $\langle H\rangle = \langle f_1\rangle : f_2$. By Theorem 4.1, we obtain $\langle \bar{f}_1\rangle = \langle H\rangle$, where $f_1 = d \cdot \bar{f}_1$, $f_2 = d \cdot \bar{f}_2$, and $d = \gcd(f_1, f_2)$. As f_1 and f_2 are both nonzero by assumption, $\langle H\rangle$ is not empty and is a principal ideal. Besides, G is a Gröbner basis of W, there must exist a polynomial $g \in k[\vec{x}] \setminus \{0\}$ such that $g \cdot \mathbf{e_2} \in G$ and $\mathrm{lm}(g) = \mathrm{lm}(\bar{f}_1)$. Moreover, we have $g = \bar{f}_1$ as G is minimal, because otherwise there should exist another polynomial in G that divides $(g - \bar{f}_1) \cdot \mathbf{e_2}$ and has a smaller leading monomial than $\mathrm{lm}(g)$. □

Theorem 4.1 only discusses the case when f_1 and f_2 are both nonzero polynomials. We can extend the result to more general cases.

COROLLARY 4.3. *Let f_1, f_2 be two polynomials in $k[\vec{x}]$ and $\prec$ be a monomial order on $k[\vec{x}]$. Suppose $W \subset (k[\vec{x}])^2$ is a $k[\vec{x}]$-module generated by $\{f_1 \cdot \mathbf{e_1}, f_2 \cdot \mathbf{e_1} - \mathbf{e_2}\}$ and G is a* **minimal** *Gröbner basis for W w.r.t. an order extended from $\prec$ in a position over term fashion with $\mathbf{e_2} \prec \mathbf{e_1}$. Let $H = \{h \in k[\vec{x}] \mid h \cdot \mathbf{e_2} \in G\}$. Then*

1. *If H is empty, then $f_1 = 0$ and $f_2 \neq 0$. In this case, $\gcd(f_1, f_2) = f_2$.*
2. *If H is not empty, then $H = \{g\}$ and $\gcd(f_1, f_2) = f_1/g$.*

PROOF. If $f_1 = 0, f_2 \neq 0$, then H can be checked to be empty. If $f_1 = f_2 = 0$, then $H = \{1\}$. If $f_1 \neq 0$ and $f_2 = 0$, then $H = \{1\}$ and $\gcd(f_1, f_2) = f_1$. In the case of f_1 and f_2 being nonzero, the result follows Theorem 4.2. □

By Corollary 4.3, the GCD of f_1 and f_2 can be obtained from the Gröbner basis G directly without any knowledge of f_1 or f_2 being zero or not.

4.2 GCD for parametric polynomials

The nice thing about using quotient ideals for computing the GCD is that Corollary 4.3 generalizes easily to the parametric case.

THEOREM 4.4. *Given $f_1, f_2 \in k[\vec{u}][\vec{x}]$ and an algebraically constructible set $A = \mathbf{V}(E) \setminus \mathbf{V}(N) \subset \bar{k}^m$, let $\mathcal{G} = \{(A_i, G_i)\}_{i=1}^{l}$ be a* **minimal** *comprehensive Gröbner system of the module $W = \langle f_1 \cdot \mathbf{e_1}, f_2 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A w.r.t. an order extended from $\prec_{\vec{x}}$ in a position over term fashion with $\mathbf{e_2} < \mathbf{e_1}$. For each branch (A_i, G_i) let $H_i = \{h \in k[\vec{u}][\vec{x}] \mid h \cdot \mathbf{e_2} \in G_i\}$. Then we have the following results.*

(1) *If H_i is empty, then $\gcd(\sigma_{\vec{\omega}}(f_1), \sigma_{\vec{\omega}}(f_2)) = \sigma_{\vec{\omega}}(f_2)$ for any $\vec{\omega} \in A_i$.*
(2) *If H_i is not empty, then $H_i = \{g_i\}$ and $\gcd(\sigma_{\vec{\omega}}(f_1), \sigma_{\vec{\omega}}(f_2)) = \frac{\sigma_{\vec{\omega}}(f_1)}{\sigma_{\vec{\omega}}(g_i)}$ for any $\vec{\omega} \in A_i$.*

PROOF. Since $\mathcal{G}$ is a **minimal** comprehensive Gröbner system, in each branch (A_i, G_i), the set $\sigma_{\vec{\omega}}(G_i)$ is a minimal Gröbner basis for any $\vec{\omega} \in A_i$. Besides, there is no polynomial G_i specializes to 0 because the leading coefficients of all polynomials in G_i are nonzero under specialization. □

Note that in Theorem 4.4 (2), the expression $\frac{\sigma_{\vec{\omega}}(f_1)}{\sigma_{\vec{\omega}}(g_i)}$ is a polynomial in $k[\vec{x}]$ for any $\vec{\omega} \in A_i$, but the expression f_1/g_i is not necessarily a polynomial in $k[\vec{u}][\vec{x}]$. However, using pseudo-division of f_1 by g_i since $\mathrm{lc}_{\vec{x}}(g_i)$ is a nonzero polynomial in $k[\vec{u}]$ that does not vanish for any σ in the branch, an associate of $\frac{\sigma_{\vec{\omega}}(f_1)}{\sigma_{\vec{\omega}}(g_i)}$ is computed.

To compute $q \in k[\vec{u}][\vec{x}]$ such that $\sigma_{\vec{\omega}}(q) \sim \sigma_{\vec{\omega}}(f_1/g_i) = \sigma_{\vec{\omega}}(f_1)/\sigma_{\vec{\omega}}(g_i)$, f_1 is multiplied by $\mathrm{lc}_{\vec{x}}(g_i)$ repeatedly during pseudo-division so that

$$(\mathrm{lc}_{\vec{x}}(g_i))^k f_1 = q \cdot g_i + r,$$

and no term in r is divisible by the leading term $\mathrm{lm}_{\vec{x}}(g_i)$.

We use a simple example to illustrate this. Let $f = x^2 - by + b$, $g = ax$ with $q = \mathrm{Quo}(f, g)$ and an algebraically constructible set $A = \mathbf{V}(\langle ab \rangle) \setminus \mathbf{V}(\langle a \rangle)$. Using a lexicographic order on $\vec{x}$, where $\vec{x} = \{x, y\}$ and $x > y$, g pseudo-divides f in $k[\vec{u}][\vec{x}]$, giving $\mathrm{lc}_{\vec{x}}(g) \cdot f = x \cdot g + r$, where $r = -aby + ab$. It is obvious that r is zero on A. Thus $q = x$. Moreover, for any $\vec{\omega} \in A$, $\frac{\sigma_{\vec{\omega}}(f)}{\sigma_{\vec{\omega}}(g)} = \frac{1}{a}x$. Therefore, $\sigma_{\vec{\omega}}(q) \sim \sigma_{\vec{\omega}}(f)/\sigma_{\vec{\omega}}(g)$. This operation is similar to the pseudo-division algorithm in [12, 13].

4.3 Algorithm

Now, we propose the main algorithm in this paper to compute the parametric GCD of parametric multivariate polynomials. This algorithm is called the **parametric GCD algorithm**.

PROPOSITION 4.5. *The parametric GCD algorithm works correctly.*

PROOF. The proof follows directly from Theorem 4.4. □

In the parametric GCD algorithm, if f_1 (or f_2) vanishes on the constructible set A, we only need to compute a minimal comprehensive Gröbner system $\{(A_j, h_j)\}_{j=1}^{s}$ of f_2 (or f_1), and then the GCD of f_1 and f_2 on each branch A_j is h_j.

We can compute the parametric GCD recursively if the number of polynomials is bigger than two.

Algorithm 1: parametric GCD algorithm

Input : $f_1, f_2 \in k[\vec{u}][\vec{x}]$, a constructible set $A \subset \bar{k}^m$, and two monomial orders $\prec_{\vec{x}}, \prec_{\vec{u}}$.
Output : a comprehensive GCD: $\{(A_j, h_j)\}_{j=1}^{s}$, where $h_i = \gcd(f_1, f_2)$ under any specialization from A_j and $\cup_{j=1}^{s} A_j = A$.

```
begin
    compute a comprehensive Gröbner system {(A_i, G_i)}_{i=1}^s for the module ⟨f_1 · e_1, f_2 · e_1 − e_2⟩.
    let i = 1.
    while i ≤ s do
        let H_i = {h ∈ k[u][x] | h · e_2 ∈ G_i}.
        if H_i is empty then
            h_i = f_2 on A_i;
        else
            H_i has exactly one polynomial, say g_i; and
            h_i = Quo(f_1, g_i) on A_i.
        end if
        let i = i + 1.
    end while
    return return {(A_j, h_j)}_{j=1}^s.
end
```

5 AN ILLUSTRATIVE EXAMPLE

We illustrate the algorithm with a simple example. Let $f_1, f_2, f_3 \in \mathbb{C}[\vec{u}][\vec{x}]$ be as follows:

$f_1 = ax^2 + bxy + a^2xz + abx + abyz + b^2y$,
$f_2 = ax^2 + bxy + (ab - a)xz - a^2x + (b^2 - b)yz - aby$,
$f_3 = ax^2 + bxy + a^2xz + (a^2 - ab)x + abyz + (ab - b^2)y$,

where $\vec{u} = \{a, b\}$, $\vec{x} = \{x, y, z\}$, $\prec_{\vec{x}}$ and $\prec_{\vec{u}}$ are all lexicographic orders with $z < y < x$ and $b < a$, respectively.

We first compute a minimal CGS $\mathcal{G}_0$ of $\langle f_1 \cdot \mathbf{e_1}, f_2 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$. There are six branches in $\mathcal{G}_0$. The first branch of $\mathcal{G}_0$ is $(A_1, G_0) = (\mathbf{V}(\langle 0 \rangle) \setminus \mathbf{V}(\langle a^3 - a^2b + a^2 \rangle), \{(x + az + b) \cdot \mathbf{e_2}, ((a^2 - ab + a)xz + (a^2 + ab)x + (ab - b^2 + b)yz + (ab + b^2)y) \cdot \mathbf{e_1} + \mathbf{e_2}, f_1 \cdot \mathbf{e_1}\})$. Then, $H_0 = \{x + az + b \in \mathbb{C}[\vec{u}][\vec{x}] \mid (x + az + b) \cdot \mathbf{e_2} \in G_1\}$ and the GCD of f_1 and f_2 on A_1 is $h_1 = f_1/(x + az + b) = ax + by$. Similarly, the GCDs on other five branches are: $(A_2, h_2) = (\mathbf{V}(\langle a - b + 1 \rangle) \setminus \mathbf{V}(\langle 2b^2 - 3b + 1 \rangle), (b - 1)x + by)$, $(A_3, h_3) = (\mathbf{V}(\langle a, b - 1 \rangle), y)$, $(A_4, h_4) = (\mathbf{V}(\langle 2a + 1, 2b - 1 \rangle) \setminus \mathbf{V}(\langle b - 1 \rangle), -\frac{1}{2}x^2 + \frac{1}{2}xy + \frac{1}{4}xz - \frac{1}{4}x - \frac{1}{4}yz + \frac{1}{4}y)$, $(A_5, h_5) = (\mathbf{V}(\langle a, b \rangle), 0)$, and $(A_6, h_6) = (\mathbf{V}(\langle a \rangle) \setminus \mathbf{V}(\langle ab^3 - ab^2 - b^4 + 2b^3 - b^2 \rangle), by)$.

For A_1, we now compute the GCD of h_1 and f_3. A minimal CGS $\mathcal{G}_1$ of $\langle h_1 \cdot \mathbf{e_1}, f_3 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A_1 has one branch: $(A_1, G_1) = (\mathbf{V}(\langle 0 \rangle) \setminus \mathbf{V}(\langle a^3 - a^2b + a^2 \rangle), \{\mathbf{e_2}, h_1 \cdot \mathbf{e_1}\})$. Then $H_1 = \{1\}$ and the GCD of h_1 and f_3 on A_1 is $h_{11} = h_1/1 = ax + by$.

Using GCDs for other branches, compute the GCD of h_2 and f_3 on A_2. A minimal CGS $\mathcal{G}_2$ of $\langle h_2 \cdot \mathbf{e_1}, f_3 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A_2 has one branch: $(A_2, G_2) = (\mathbf{V}(\langle a - b + 1 \rangle) \setminus \mathbf{V}(\langle 2b^2 - 3b + 1 \rangle), \{\mathbf{e_2}, h_2 \cdot \mathbf{e_1}\})$. Then $H_2 = \{1\}$ and the GCD of h_2 and f_3 on A_2 is $h_{22} = h_2/1 = (b-1)x + by$.

For the GCD of h_3 and f_3 on A_3: A minimal CGS $\mathcal{G}_3$ of $\langle h_3 \cdot \mathbf{e_1}, f_3 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A_3, and obtain one branch: $(A_3, G_3) = (\mathbf{V}(\langle a, b - 1 \rangle), \{\mathbf{e_2}, h_3 \cdot \mathbf{e_1}\})$. $H_3 = \{1\}$ so the GCD of h_3 and f_3 on A_3 is $h_{33} = h_3/1 = y$.

For the GCD of h_4 and f_3 on A_4: A minimal CGS $\mathcal{G}_4$ of $\langle h_4 \cdot \mathbf{e_1}, f_3 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A_4, has one branch: $(A_4, G_4) = (\mathbf{V}(\langle 2a + 1, 2b - 1 \rangle) \setminus \mathbf{V}(\langle b - 1 \rangle), \{(2x - z + 1)\mathbf{e_2}, (3x - 3y) \cdot \mathbf{e_1} - 4\mathbf{e_2}\})$. $H_4 = \{2x - z + 1\}$ so the GCD of h_4 and f_3 on A_4 is $h_{44} = h_4/(2x - z + 1) = -x + y$.

For branch A_5, $h_5 = 0$ and f_3 vanishes giving the GCD $h_{55} = 0$.

For the GCD of h_6 and f_3 on A_6: A minimal CGS $\mathcal{G}_6$ of $\langle h_6 \cdot \mathbf{e_1}, f_3 \cdot \mathbf{e_1} - \mathbf{e_2} \rangle$ on A_6 also has a single branch: $(A_6, G_6) = (\mathbf{V}(\langle a \rangle) \setminus \mathbf{V}(\langle ab^3 - ab^2 - b^4 + 2b^3 - b^2 \rangle), \{\mathbf{e_2}, h_6 \cdot \mathbf{e_1}\})$. Then $H_6 = \{1\}$ so the GCD of h_6 and f_3 on A_6 is $h_{66} = h_6/1 = by$.

The parametric GCDs of $\{f_1, f_2, f_3\}$ are

$$\begin{cases} (\mathbf{V}(\langle 0 \rangle) \setminus \mathbf{V}(\langle a^3 - a^2 b + a^2 \rangle), ax + by), \\ (\mathbf{V}(\langle a - b + 1 \rangle) \setminus \mathbf{V}(\langle 2b^2 - 3b + 1 \rangle), (b-1)x + by), \\ (\mathbf{V}(\langle a, b - 1 \rangle), y), \\ (\mathbf{V}(\langle 2a + 1, 2b - 1 \rangle) \setminus \mathbf{V}(\langle b - 1 \rangle), -x + y), \\ (\mathbf{V}(\langle a, b \rangle), 0), \\ (\mathbf{V}(\langle a \rangle) \setminus \mathbf{V}(\langle ab^3 - ab^2 - b^4 + 2b^3 - b^2 \rangle), by). \end{cases}$$

6 GCD OF A SYSTEM OF POLYNOMIALS

Given a system of parametric polynomials (with more than 2 polynomials), their GCD can also be computed by successively computing the GCD two polynomials at a time. Which two polynomials we choose can make a big difference. We recognize that many heuristics are possible based on the degrees of the polynomials as well as by first considering specializations on which all but one polynomials vanish.

Currently, we compute the GCD of a pair of parametric polynomials whose output is a finite set of constructible sets with the corresponding GCD. For each such branch, the GCD is used to compute its GCD with the next polynomial leading to more branches. The performance of our naive implementation is reasonable because as computations proceed, the degree of intermediate GCDs goes down substantially. Our ultimate goal is to use a single comprehensive Gröbner system computation for this but we have not been able to develop such an algorithm yet.

7 IMPLEMENTATION AND COMPARATIVE PERFORMANCE

The proposed algorithm has been implemented in the computer algebra system *Singular (4-0-3)* [6]. The implementation has been tried on a number of examples including the examples in [16] and it has been compared with implementations of the two algorithms proposed by Nagasaka. The following table compares our implementation with Nagasaka's two algorithms for computing GCD of parametric multivariate polynomials (Nagasaka-GT and Nagasaka-SS) implemented by us in Singular. The parametric polynomials for the examples are given below:

Ex.1: $F_1 = \{ax^3 + (a^3 - a + 1)x^2y + (a^2 + 2)xy^2 + (3a^2 - 3)y^3, ax^3 + (a+1)x^2y + 4xy^2 + 3y^3\}$, $\vec{x} = \{x, y\}$, $\vec{u} = \{a\}$.

Ex.2: $F_2 = \{(x + ay + bz)^3 + c(x + ay + bz) + d, 3(x + ay + bz)^2 + c, 3a(x + ay + bz)^2 + ac, 3b(x + ay + bz)^2 + bc\}$, $\vec{x} = \{x, y, z\}$, $\vec{u} = \{a, b, c, d\}$.

Ex.3: $F_3 = \{axz + (a-1)yz, (a-1)x^2 + axy\}$, $\vec{x} = \{x, y, z\}$, $\vec{u} = \{a\}$.

Ex.4: $F_4 = \{ax^3y^2z + (1-b)(y^2+z), (1-a)x^3y^2z + b(y^2+z)\}$, $\vec{x} = \{x, y, z\}$, $\vec{u} = \{a, b\}$.

Ex.5: $F_5 = \{(1-a)y^2 - bx^2 - cxy, (1-b)x^2 - ay^2 - cxy\}$, $\vec{x} = \{x, y\}$, $\vec{u} = \{a, b, c\}$.

Ex.6: $F_6 = \{ax^2 + bxy + a^2xz + abx + abyz + b^2y, ax^2 + bxy + (ab - a)xz - a^2x + (b^2 - b)yz - aby, ax^2 + bxy + a^2xz + (a^2 - ab)x + abyz + (ab - b^2)y\}$, $\vec{x} = \{x, y, z\}$, $\vec{u} = \{a, b\}$.

Ex.7: $F_7 = \{ax^2y + bx + y^3, ax^2y + bxy + cx, y^2 + bx^2y + cxy\}$, $\vec{x} = \{x, y\}$, $\vec{u} = \{a, b, c\}$.

Ex.8: $F_8 = \{ax^3y + cxz^2, x^2y + 3dy + z, cx^2 + bxy, x^2y^2 + ax^2\}$, $\vec{x} = \{x, y, z\}$, $\vec{u} = \{a, b, c, d\}$.

Ex.9: $F_9 = \{(ax + by)(x + a)(y - b), (aby^2 + b - 1)(bx + ay)(x + b)(y - a), (axy + a^2x - 3a)(ax + by)(x + b), (bx + ay)(ax + by)(ax + b)(by + a)\}$, $\vec{x} = \{x, y\}$, $\vec{u} = \{a, b\}$.

Ex.10: $F_{10} = \{(1-a)x^2y + bx^2 + y^2, ax^2y + (1-b)xy + cx, y^2 + bx^2y + (1-c)xy\}$, $\vec{x} = \{x, y\}$, $\vec{u} = \{a, b, c\}$.

For all these examples, the term orders used on $\vec{u}$ and $\vec{x}$ are lexicographic orders, respectively.

In Table 1, entries labeled **New** are for the proposed algorithm. Timings were obtained on a Core i7-4790 3.60GHz with 4GB Memory running Windows 7. As is evident from Table 1, the proposed algorithm performs better than the Nagasaka's algorithms. The code for the three algorithms and the examples are available on the web at:

http://www.mmrc.iss.ac.cn/~dwang/software.html.

8 CONCLUDING REMARKS

A new algorithm for computing the parametric GCD has been proposed. Using module comprehensive Göbner system, the parametric GCD of multivariate polynomials can be computed. The experimental data in Table 1 suggests that the proposed algorithm is superior in practice in comparison with both the algorithms proposed by Nagasaka. We think this is because our method does not compute the primitive part of polynomials in different parameter spaces, and our theorem guarantees that a parametric polynomial is divisible by another parametric polynomial on various algebraically constructible sets. Since the computational efficiency of our algorithm depends on the number of branches in a module comprehensive Gröbner system, we believe that the proposed algorithm can be further improved by removing inessential polynomials from comprehensive Gröbner system computations as discussed in [11]. This will be further studied in the future along with heuristics to minimize the number of

Table 1: Timings

Example	Algorithm	Time(sec.)
Ex.1	New	0.640
	Nagasaka-GT	2.062
	Nagasaka-SS	0.809
Ex.2	New	1.023
	Nagasaka-GT	47.210
	Nagasaka-SS	19.680
Ex.3	New	0.836
	Nagasaka-GT	6.730
	Nagasaka-SS	4.125
Ex.4	New	0.597
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	12.736
Ex.5	New	2.475
	Nagasaka-GT	10.760
	Nagasaka-SS	4.108
Ex.6	New	2.426
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	21.558
Ex.7	New	6.419
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	$>$ 1h
Ex.8	New	5.286
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	37.172
Ex.9	New	15.351
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	98.744
Ex.10	New	10.011
	Nagasaka-GT	$>$ 1h
	Nagasaka-SS	$>$ 1h

branches to be considered for computing the GCD of a system of polynomials.

ACKNOWLEDGMENTS

This research was supported in part by the National Natural Science Foundation of China under Grant No. 11371356, the CAS Project QYZDJ-SSW-SYS022, the National Science Foundation DMS-1217054 and the CAS/SAFEA International Partnership Program for Creative Research Teams. The authors would like to thank the anonymous referees for their detailed suggestions on the paper which have made it more readable.

REFERENCES

[1] S.A. Abramov and K.Y. Kvashenko. 1993. On the greatest common divisor of polynomials which depend on a parameter. In *Proceedings of the 1993 ACM International Symposium on Symbolic and Algebraic Computation*. 152–156.

[2] A. Ayad. 2010. Complexity of algorithms for computing greatest common divisors of parametric univariate polynomials. *International Journal of Algebra* 4, 4 (2010), 173–188.

[3] W.S. Brown. 1971. On Euclid's algorithm and the computation of polynomial greatest common divisors. *J. ACM* 18 (1971), 478–504.

[4] D. Cox, J. Little, and D. O'shea. 1992. *Ideals, varieties, and algorithms*. Springer, third edition.

[5] D. Cox, J. Little, and D. O'shea. 2005. *Using algebraic geometry*. Springer, New York, second edition.

[6] W. Decker, G.-M. Greuel, G. Pfister, and H. Schoenemann. 2016. SINGULAR 4.0.3. a computer algebra system for polynomial computations, FB Mathematik der Universitaet, D-67653 Kaiserslautern. https://www.singular.uni-kl.de/.

[7] P. Gianni and B. Trager. 1985. GCDs and factoring multivariate polynomials using Gröbner bases. In *Proceedings of EUROCAL '85, European Conference on Computer Algebra. Lecture Notes in Computer Science, vol 204. Springer, Berlin, Heidelberg*. 409–410.

[8] D. Kapur. 1995. An approach for solving systems of parametric polynomial equations. In *Principles and Practices of Constraint Programming*, Saraswat and Van Hentenryck (Eds.). MIT Press, 217–244.

[9] D. Kapur, Y. Sun, and D.K. Wang. 2010. A new algorithm for computing comprehensive Gröbner systems. In *Proceedings of the 2010 ACM International Symposium on Symbolic and Algebraic Computation*. 29–36.

[10] D. Kapur, Y. Sun, and D.K. Wang. 2013. An efficient algorithm for computing a comprehensive Gröbner system of a parametric polynomial system. *Journal of Symbolic Computation* 49 (2013), 27–44.

[11] D. Kapur and Y. Yang. 2014. An algorithm for computing a minimal comprehensive Gröbner basis of a parametric polynomial system. In *Proceedings of Conference Encuentros de Algebra Comptacionaly Aplicaciones (EACA)*.

[12] A. Montes. 2002. A new algorithm for discussing Gröbner bases with parameters. *Journal of Symbolic Computation* 33, 2 (2002), 183–208.

[13] A. Montes and H. Schoenemann. 2016. grobcov.lib. http://www.singular.uni-kl.de/Manual/latest/sing_900.htm.

[14] J. Moses and D.Y.Y. Yun. 1973. The EZ GCD algorithm. In *Proceedings of ACM'73*. ACM Press, New York, 159–166.

[15] K. Nabeshima. 2012. Stability conditions of monomial bases and comprehensive Gröbner systems. In *Proceedings of the International Conference on Computer Algebra in Scientific Computing*, Vol. 7442. Springer-Verlag, 248–259.

[16] K. Nagasaka. 2017. Parametric greatest common divisors using comprehensive Gröbner systems. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*. 341–348.

[17] M. Sanuki, D. Inaba, and T. Sasaki. 2016. Computation of GCD of sparse multivariate polynomials by extended hensel construction. In *Proceedings of the 2016 IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. 34–41.

[18] T. Sasaki and M. Suzuki. 1992. Three new algorithms for multivariate polynomial GCD. *Journal of Symbolic Computation* 13, 4 (1992), 395–411.

[19] A. Suzuki and Y. Sato. 2006. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In *Proceedings of the 2006 ACM International Symposium on Symbolic and Algebraic Computation*. 326–331.

[20] V. Weispfenning. 1992. Comprehensive Gröbner bases. *Journal of Symbolic Computation* 14, 3 (1992), 669–683.

[21] R. Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings of EUROSAM'79*. Springer-Verlag, 216–226.