Computing Comprehensive Gröbner Systems and Comprehensive Gröbner Bases Simultaneously *

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

Yao Sun
KLMM
Academy of Mathematics and Systems Science,CAS
Beijing 100190, China
sunyao@amss.ac.cn

Dingkang Wang
KLMM
Academy of Mathematics and Systems Science,CAS
Beijing 100190, China
dwang@mmrc.iss.ac.cn

ABSTRACT

In Kapur et al (ISSAC, 2010), a new method for computing a comprehensive Gröbner system of a parameterized polynomial system was proposed and its efficiency over other known methods was effectively demonstrated. Based on those insights, a new approach is proposed for computing a comprehensive Gröbner basis of a parameterized polynomial system. The key new idea is not to simplify a polynomial under various specialization of its parameters, but rather keep track in the polynomial, of the power products whose coefficients vanish; this is achieved by partitioning the polynomial into two parts–*nonzero* part and *zero* part for the specialization under consideration. During the computation of a comprehensive Gröbner system, for a particular branch corresponding to a specialization of parameter values, nonzero parts of the polynomials dictate the computation, i.e., computing S-polynomials as well as for simplifying a polynomial with respect to other polynomials; but the manipulations on the whole polynomials (including their zero parts) are also performed. Gröbner basis computations on such pairs of polynomials can also be viewed as Gröbner basis computations on a module. Once a comprehensive Gröbner system is generated, both nonzero and zero parts of the polynomials are collected from every branch and the result is a *faithful* comprehensive Gröbner basis, to mean that every polynomial in a comprehensive Gröbner basis belongs to the ideal of the original parameterized polynomial system. This technique should be applicable to other algorithms for computing a comprehensive Gröbner system as well, thus producing both a comprehensive Gröbner system as well as a faithful comprehensive Gröbner basis of a parameterized polynomial system simultaneously. The approach is exhibited by adapting the recently proposed method for computing a comprehensive Gröbner system in (ISSAC, 2010) for computing a comprehensive Gröbner basis. The timings on a collection of examples demonstrate that this new algorithm for computing comprehensive Gröbner bases has better performance than other existing algorithms.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms

General Terms

Algorithms

Keywords

Gröbner basis, comprehensive Gröbner basis, comprehensive Gröbner system.

1. INTRODUCTION

The concept of a comprehensive Gröbner basis was introduced by Weispfenning [16] as a special basis of a parametric polynomial system such that for every possible specialization of its parameters, the basis obtained from the comprehensive Gröbner basis serves as a Gröbner basis of the ideal generated by the specialization of the parametric polynomial system (see also [7] where a related concept of a parametric Gröbner basis is introduced). In that paper, Weispfenning gave an algorithm for computing a comprehensive Gröbner basis from a comprehensive Gröbner system, consisting of Gröbner bases for various specializations of the parameters. In this paper, we show how both comprehensive Gröbner system and comprehensive Gröbner basis of a parametric polynomial system can be constructed together. The key idea is to retain terms in polynomials even when parameters are specialized, resulting in vanishing of the coefficients of these terms.

To illustrate the key idea, let us consider Example 8.4 from [17] where Weispfenning defined the concept of a canonical comprehensive Gröbner basis of a parametric polynomial system to mimic the concept of a reduced Gröbner basis of a polynomial system determined by the associated ideal and term order. Suppose there are two polynomials $f, g \in k[u, v][x, y]$:

$$f = y + ux + v, \quad g = uy + x + v.$$

*The first author is supported by the National Science Foundation award CCF-0729097 and the last two authors are supported by NKBRPC 2011CB302400, NSFC 10971217 and 60821002/F02.

ISSAC'11, June 8–11, 2011, San Jose, California, USA.

Further, suppose we are interested in computing Gröbner basis with the lexicographic order induced by y > x.

Clearly, f can be used to simplify g, resulting in

$$h = g - uf = (1-u^2)x - uv + v.$$

In fact, g can be deleted without any loss of generality. Based on the specialization of u and v, the leading power product of h is either x or 1.

For the branch where $(1-u^2) \neq 0$, the nonzero part of h is $(1-u^2)x+(-uv+v)$. Since both f and h have noncomparable leading power products, $\{f, h\}$ constitutes a Gröbner basis for this branch for those specializations satisfying $(1-u^2) \neq 0$.

For the branch, where $(1-u^2) = 0$ and $(-uv+v) \neq 0$ for all those specializations of u and v, the nonzero part of h is $(-uv+v)$ and the zero part of h is $(1-u^2)x$. For this branch, a Gröbner basis is $\{h\}$, since the leading power product of the nonzero part of h is 1, which reduces every other power product. If h is simplified using the specializations of u and v, the Gröbner basis would have been $\{1\}$. However, such a Gröbner basis is not *faithful*, since 1 is not in $\langle f, g\rangle$. But to maintain faithfulness, we keep h instead.

Finally, for the branch where $(1-u^2) = 0$ and $(-uv+v) = 0$, h vanishes completely. And, the nonzero part of f is itself, since the leading coefficient of f is 1. A Gröbner basis for this branch is $\{f\}$; if the specialization of u and v had been used to simplify f, we have $\{y + x + v\}$ as a Gröbner basis.

Using the proposed algorithm, a comprehensive Gröbner system consists of three branches: a branch corresponding to specializations satisfying $(1-u^2) \neq 0$, for which $\{f, h\}$ is a Gröbner basis for a 0-dimensional specialization; another branch, corresponding to the specialization satisfying $(1-u^2) = 0, (-uv+v) \neq 0$ (which can be further simplified to $u + 1 = 0, v \neq 0$), for which $\{h\}$ is a Gröbner basis for the ideal generated by 1; the last branch corresponds to the specialization $(1-u^2) = 0, (-uv+v) = 0$, for which $\{f\}$ is a Gröbner basis for the one dimensional ideal.

The key difference between the output of this algorithm and other algorithms including our algorithm in [9], is that a Gröbner basis in every branch in a comprehensive Gröbner system is a subset of the original ideal, and hence contributes to a comprehensive Gröbner basis.

A *faithful* comprehensive Gröbner basis for the above system can be easily constructed by taking the union of Gröbner bases along all the branches; for every possible specialization, there is exactly one branch generating a Gröbner basis for the specialized ideal; furthermore, by construction, all the polynomials are in the ideal of the original system. For the above example, a comprehensive Gröbner basis is $\{f, h\}$.[1]

Based on the ideas illustrated for the above example, we propose in this paper, an algorithm for simultaneously computing a comprehensive Gröbner system as well as the associated comprehensive Gröbner basis that is faithful. This algorithm builds on our recently proposed algorithm [9] for computing a comprehensive Gröbner system as its foundation. The key difference between the new algorithm and the previous algorithm is that unlike in the old algorithm, during computations, the zero part of a polynomial under a specialization is also kept in a tuple representation so as to recover the original polynomial when needed. Specifically, when computing a comprehensive Gröbner system of the set $F \subset k[U][X]$, we use a tuple $(q, \bar{q}) \in (k[U][X])^2$ to replace each polynomial $p = q + \bar{q}$ in the computation, with the following properties: (i) $p \in \langle F\rangle$, and (ii) $\bar{q}$ is 0 under the specialization of parameters being considered. When a comprehensive Gröbner system of F is obtained, then for each 2-tuple $(g, \bar{g})$ in this comprehensive Gröbner system, we recover the faithful polynomial $g + \bar{g}$; this way, a comprehensive Gröbner basis of F is obtained simultaneously with the comprehensive Gröbner system.

Generally, a comprehensive Gröbner basis for a given polynomial set F is harder to compute than a comprehensive Gröbner system of F. The difficulty of computing a comprehensive Gröbner basis of F is that, all the polynomials in this comprehensive Gröbner basis should be faithful polynomials, i.e., these polynomials should belong to the ideal $\langle F\rangle$, while the polynomials in a comprehensive Gröbner system of F are not necessarily faithful polynomials. Therefore, the algorithms for computing comprehensive Gröbner systems usually have better performance than those for comprehensive Gröbner bases. Consequently, a feasible method for computing comprehensive Gröbner bases is to compute comprehensive Gröbner systems first, and then transform all polynomials in the comprehensive Gröbner systems to faithful polynomials. Unfortunately, this transformation is usually expensive when the computation of comprehensive Gröbner systems is finished. So the goal of the new technique is to make this transformation easier. The proposed idea of retaining polynomials from the ideal of the original polynomial system while computing Gröbner bases along different branches can be used in all algorithms for computing comprehensive Gröbner systems, including Weispfenning's [16], Kapur's [7], Montes' [11, 10, 13], Wang's [2], Suzuki-Sato's [15], Nabeshima's [14] as well as our recently proposed algorithm [9].

Comprehensive Gröbner basis and Gröbner system constructions have been found useful in many engineering applications which can be modeled using parameterized polynomial systems; see [4, 6, 11] for examples of some applications. These constructions have also been found useful for automated geometry theorem proving [2] and automated geometry theorem discovery [12], as well as more recently, for computing loop invariants in program analysis [8]. Solving parametric polynomial systems has also been investigated by Chou and Gao [5] and Chen et al. [1] using the characteristic set construction, as well as by Wibmer [18] using Gröbner cover.

The paper is organized as follows. We give some notations and definitions in Section 2. The new technique mentioned above is described in Section 3. We propose a new algorithm for computing comprehensive Gröbner bases in Section 4. A simple example illustrates the proposed algorithm in Section 5. Empirical data and comparison with other existing algorithms are presented in Section 6. Concluding remarks follow in Section 7.

2. NOTATIONS AND DEFINITIONS

Let k be a field, R be the polynomial ring $k[U]$ in the parameters $U = \{u_1, \cdots, u_m\}$, and $R[X]$ be the polynomial ring over the parameter ring R in the variables $X =$

[1] An interested reader would notice that this result is different from the one reported in [17]. In fact, the canonical comprehensive Gröbner basis reported there for the same order is a proper superset of the above result, suggesting that after all, the definition in [17] does not quite capture the notion of minimality and hence, canonicity.

$\{x_1, \cdots, x_n\}$ where $X \cap U = \emptyset$, i.e. X and U are disjoint sets.

Let $PP(X)$, $PP(U)$ and $PP(U, X)$ be the sets of power products of X, U and $X \cup U$ respectively. $\prec_{X,U}$ is an admissible block term order on $PP(U, X)$ where $U \ll X$. The orders $\prec_X$ and $\prec_U$ are the restrictions of $\prec_{X,U}$ on $PP(X)$ and $PP(U)$ respectively.

For a polynomial $f \in R[X] = k[U][X]$, the leading power product, leading coefficient and leading monomial of f w.r.t. the order $\prec_X$ are denoted by $\mathrm{lpp}_X(f)$, $\mathrm{lc}_X(f)$ and $\mathrm{lm}_X(f)$ respectively. Since f can also be regarded as an element of $k[U, X]$, in this case, the leading power product, leading coefficient and leading monomial of f w.r.t. the order $\prec_{X,U}$ are denoted by $\mathrm{lpp}_{X,U}(f)$, $\mathrm{lc}_{X,U}(f)$ and $\mathrm{lm}_{X,U}(f)$ respectively. For f, we always have $\mathrm{lm}_X(f) = \mathrm{lc}_X(f)\mathrm{lpp}_X(f)$ and $\mathrm{lm}_{X,U}(f) = \mathrm{lc}_{X,U}(f)\mathrm{lpp}_{X,U}(f)$.

Given a field L, a specialization of R is a homomorphism $\sigma : R \longrightarrow L$. In this paper, we always assume L to be an algebraically closed field containing k and we only consider the specializations induced by the elements in L^m. That is, for $\bar{a} \in L^m$, the induced specialization $\sigma_{\bar{a}}$ is defined as follows.

$$\sigma_{\bar{a}} : f \longrightarrow f(\bar{a})$$

where $f \in R$. Every specialization $\sigma : R \longrightarrow L$ extends canonically to a specialization $\sigma : R[X] \longrightarrow L[X]$ by applying σ coefficient-wise.

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

Definition 2.1 (CGS) *Let F be a subset of $R[X]$, $A_1, \cdots, A_l$ be algebraically constructible subsets of L^m, $G_1, \cdots, G_l$ be subsets of $R[X]$, and S be a subset of L^m such that $S \subseteq A_1 \cup \cdots \cup A_l$. A finite set $\mathcal{G} = \{(A_1, G_1), \cdots, (A_l, G_l)\}$ is called a* **comprehensive Gröbner system** *on S for F, if $\sigma_{\bar{a}}(G_i)$ is a Gröbner basis of the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[X]$ for $\bar{a} \in A_i$ and $i = 1, \cdots, l$. Each (A_i, G_i) is called a branch of $\mathcal{G}$. If $S = L^m$, then $\mathcal{G}$ is simply called a comprehensive Gröbner system for F.*

For an $F \subset R = k[U]$, the variety defined by F in L^m is denoted by $V(F)$. In this paper, the constructible set A_i always has the form: $A_i = V(E_i) \setminus V(N_i)$ where E_i, N_i are subsets of $k[U]$. Particularly, we call E_i and N_i equality constraints and disequality constraints respectively. Clearly, if the set $A_i = V(E_i) \setminus V(N_i)$ is empty, the branch (A_i, G_i) is redundant.

Definition 2.2 (CGB) *Let F be a subset of $R[X]$ and S be a subset of L^m. A finite subset G in $R[X]$ is called a* **comprehensive Gröbner basis** *on S for F, if $\sigma_{\bar{a}}(G)$ is a Gröbner basis of the ideal $\langle \sigma_{\bar{a}}(F) \rangle$ in $L[X]$ for each $\bar{a}$ in S. If $S = L^m$, then G is simply called a comprehensive Gröbner basis for F.*
A comprehensive Gröbner basis G of F is called **faithful** *if in addition, every element of G is also in $\langle F \rangle$.*

A typical approach to compute a comprehensive Gröbner basis of F is to first compute a comprehensive Gröbner system of F and then further process it to generate a comprehensive Gröbner basis. It follows from the above definitions of a comprehensive Gröbner system and a comprehensive Gröbner basis that given a comprehensive Gröbner system $\mathcal{G} = \{(A_1, G_1), \cdots, (A_l, G_l)\}$ on S for $F \subset R[X]$, if $G_i \subset \langle F \rangle$ for $i = 1, \cdots, l$, then the set $G_1 \cup \cdots \cup G_l$ is a comprehensive Gröbner basis on S for F. However, in almost all the known algorithms for computing a comprehensive Gröbner system, G_i is typically never a subset of the ideal $\langle F \rangle$, since polynomials get simplified based on parameter specialization. The main challenge is thus to recover $G'_i \subset \langle F \rangle$ such that $\sigma_{\bar{a}}(G_i) = \sigma_{\bar{a}}(G'_i)$ for $\bar{a} \in A_i$. In the next section, we propose a new technique to obtain the G'_i's efficiently during the computation of a comprehensive Gröbner system.

3. A POLYNOMIAL AS A TUPLE UNDER PARAMETER SPECIALIZATION

As mentioned in the introduction and illustrated using an example, the key new idea in our approach is to keep track of polynomials in $\langle F \rangle$ while computing various Gröbner bases under different parameter specializations. If some terms in these polynomials vanish due to specialization of parameters during the computation of a comprehensive Gröbner system, this information can be kept by splitting the polynomial into the nonzero part and the zero part under the specialization.

A polynomial $p \in \langle F \rangle$ is replaced along a branch of a comprehensive Gröbner system computation for a specialization of parameters from a constructible set A_i, by a tuple $(q, \bar{q})$ such that (i) $p = q + \bar{q}$, and further, (ii) for every parameter specialization σ from A_i, $\sigma(\bar{q})$ is 0. We call $(q, \bar{q})$ an **admissible tuple representation** of p in the ideal $\langle F \rangle$ w.r.t. constructible set A_i.[2]

Let us observe some properties of admissible tuple representation of polynomials from an ideal. Given admissible tuple representations $(p, \bar{p})$ and $(q, \bar{q})$ of $p + \bar{p}$ and $q + \bar{q}$ in $\langle F \rangle$, w.r.t. A_i, $(p + q, \bar{p} + \bar{q})$ is an admissible tuple representation of $p + q + \bar{p} + \bar{q}$ in the ideal generated by $p + \bar{p}$ and $q + \bar{q}$ w.r.t. A_i. Furthermore, given a polynomial r, $(r \cdot p, r \cdot \bar{p})$ is an admissible tuple representation of $r \cdot p + r \cdot \bar{p}$ in the ideal generated by $p + \bar{p}$ w.r.t. A_i.

Let us now consider operations on polynomials and parameter specializations performed while computing a comprehensive Gröbner system. In Gröbner basis computations, there are two key steps – simplification of a polynomial by another polynomial and S-polynomial construction from a pair of distinct polynomials. In addition, we modify parametric constraints by adding disequalities and equalities on parameters, and modify the tuple representation of polynomials under consideration.

Particularly, if a parametric constraint h is added to a constructible set $A_i = V(E_i) \setminus V(N_i)$, with $E_i, N_i \subset k[U]$, the new constructible set A'_i should be nonempty, i.e., $A'_i = V(E'_i) \setminus V(N'_i) \neq \emptyset$ where either $E'_i = E_i \cup \{h\}, N'_i = N_i$ if the constraint is $h = 0$ or $E'_i = E_i, N'_i = \{n \cdot h \mid n \in N_i\}$ if the constraint is $h \neq 0$. Typically, h is the leading coefficient of the polynomial q in a tuple $(q, \bar{q})$ in a computation. If A_i is extended by adding $h \neq 0$, then the tuple is not changed; otherwise, if $h = 0$ is added as a new parameter constraint to E_i of A_i, then the above tuple is replaced by $(q', \bar{q}')$ by moving all terms in q that vanish to $\bar{q}'$ such that $q + \bar{q} = q' + \bar{q}'$ and the leading coefficient of q' is not always zero for the specializations from A'_i and $\bar{q}'$ is 0 w.r.t.

[2]We decided not to include an additional condition on an admissible tuple representation that $\mathrm{lc}_X(q) \neq 0$ wrt A_i, because this property is not preserved under addition. However, as the reader would observe later, this third condition is satisfied by tuples generated in the algorithms below.

A_i'. These are admissible tuple representations. We can make the leading coefficients of first components of tuples always nonzero w.r.t. some parametric constraints by using the method in [7, 11, 2].

For a constructible set A_i and two admissible tuple representations $\mathbf{p} = (p, \bar{p}), \mathbf{q} = (q, \bar{q})$ of $p + \bar{p}$ and $q + \bar{q}$, respectively, assuming both $\mathrm{lc}_X(p)$ and $\mathrm{lc}_X(q)$ are nonzero w.r.t. A_i, their **S-polynomial** is defined to be

$$\frac{\mathrm{lc}_X(q)L_{pq}}{\mathrm{lpp}_X(p)} \cdot (p, \bar{p}) - \frac{\mathrm{lc}_X(p)L_{pq}}{\mathrm{lpp}_X(q)} \cdot (q, \bar{q}),$$

where $L_{pq} = \mathrm{lcm}(\mathrm{lpp}_X(p), \mathrm{lpp}_X(q))$. Clearly, the S-polynomial of $\mathbf{p}$ and $\mathbf{q}$ is also an admissible tuple representation. And the polynomial corresponding to the above tuple is in the ideal of $\{p + \bar{p}, q + \bar{q}\}$.

Similarly, along a branch corresponding to a constructible set A_i, assuming $\mathrm{lpp}_X(g)$ divides $\mathrm{lpp}_X(f)$ and $\mathrm{lc}_X(g)$ is nonzero w.r.t. A_i, the result of reducing (simplifying) $\mathbf{f} = (f, \bar{f})$ by $\mathbf{g} = (g, \bar{g})$ is:

$$\begin{aligned} & \mathrm{lc}_X(g)\mathbf{f} - \tfrac{\mathrm{lm}_X(f)}{\mathrm{lpp}_X(g)} \cdot \mathbf{g} \\ = \; & (\mathrm{lc}_X(g)f - \tfrac{\mathrm{lm}_X(f)}{\mathrm{lpp}_X(g)}g, \mathrm{lc}_X(g)\bar{f} - \tfrac{\mathrm{lm}_X(f)}{\mathrm{lpp}_X(g)}\bar{g}), \end{aligned}$$

which is an admissible tuple representation of the simplified polynomial in the ideal of $\{f + \bar{f}, g + \bar{g}\}$.

In algorithms for computing a comprehensive Gröbner system from F, if we use the above admissible tuple representation of polynomials in F and perform the above S-polynomial and reduction as defined above on tuples, then, for each branch, we get a finite set of admissible tuples such that their first components constitute a Gröbner basis of F under the parameter specialization belonging to A_i. Furthermore, these constructions produce tuples such that the polynomials corresponding to them, obtained by adding the two components of the tuple, are in the ideal $\langle F \rangle$. In this way, a faithful Gröbner basis is generated for every branch corresponding to A_i.

3.1 Manipulating Tuple Representations of Polynomials using Module Operations

As the reader might have noticed, it suffices to perform various Gröbner basis operations only on the first component of the tuple representation of a polynomial from the input ideal to generate a comprehensive Gröbner system. However, to compute a comprehensive Gröbner basis consisting of faithful polynomials from the input ideal, the same operations have to be recorded on the second component also, even though computations on the second components do not affect the overall computation of a Gröbner basis along a branch under a specialization. A Gröbner basis implementation that also provides information about how the elements of a Gröbner basis can be obtained from the input basis (i.e., the representation of each element of a Gröbner basis in terms of the input basis), can be used to derive the required information about the second components; hence, in this way, the faithful polynomial corresponding to the first component in a Gröbner basis along a particular branch can be generated.

In the absence of such information available about Gröbner basis elements in terms of the input basis, existing implementations of Gröbner basis algorithms on modules can be used instead, since all the operations on admissible tuple representations can be converted to basic module operations. Most of the terminologies on "module" in this section can be found in Chapter 5 of [3].

Let F be a subset of $R[X]$ and A_i be a constructible set. Then

$$\mathbf{M}(F, A_i) = \{(p, \bar{p}) \mid p + \bar{p} \in \langle F \rangle \text{ and } \sigma_{\bar{a}}(\bar{p}) = 0 \text{ for all } \bar{a} \in A_i\}$$

is the set of all admissible tuple representations of polynomials from $\langle F \rangle$ w.r.t. A_i. Clearly, $\mathbf{M}(F, A_i) \subset (R[X])^2$ is an $R[X]$-module with the following basic operations:

1. for $\mathbf{p} = (p, \bar{p}), \mathbf{q} = (q, \bar{q}) \in \mathbf{M}(F, A_i)$, $\mathbf{p} + \mathbf{q} = (p + q, \bar{p} + \bar{q}) \in \mathbf{M}(F, A_i)$, and
2. for $\mathbf{p} = (p, \bar{p}) \in \mathbf{M}(F, A_i)$ and $r \in R[X]$, $r \cdot \mathbf{p} = (r \cdot p, r \cdot \bar{p}) \in \mathbf{M}(F, A_i)$.

Since $\mathbf{M}(F, A_i)$ is a module, we can use general definitions of the S-polynomial and reduction in a module. To make these definitions consistent with those defined on tuples in last subsection, it suffices to extend the term order defined on $R[X]$ to the free $R[X]$-module $(R[X])^2$ in a POT (position over term) fashion with $(1, 0) \succ (0, 1)$.

An important operation for computing a comprehensive Gröbner system is simplifying $(p, \bar{p}) \in \mathbf{M}(F, A_i)$ w.r.t. A_i. As mentioned earlier, we can simplify $(p, \bar{p})$ to $(p', \bar{p}')$ by moving all terms in p that vanish to $\bar{p}$ such that $p + \bar{p} = p' + \bar{p}'$ and the leading coefficient of p' is not always zero for the specializations from A_i and $\bar{p}'$ is 0 w.r.t. A_i. This simplification can also be expressed using module operations. Assume $A_i = V(E) \setminus V(N)$ where $E, N \subset R$ and $\langle E \rangle$ is radical. Then simplifying $(p, \bar{p})$ w.r.t. A_i is equivalent to reducing $(p, \bar{p})$ by the set $\{(e, -e) \mid e \in E\} \subset \mathbf{M}(F, A_i)$. For example, let $F = \{ax^2 + bx + a + 1\} \subset \mathbb{Q}[a, b][x]$, $A_i = V(E) = V(\{a, b - 1\})$ and $\mathbf{p} = (ax^2 + bx + a + 1, 0) \in \mathbf{M}(F, A_i)$. Then $\mathbf{p} = (ax^2 + bx + a + 1, 0)$ can be reduced to $(x + 1, ax^2 + bx - x + a)$ as follows:

$$(ax^2 + bx + a + 1, 0) - (x^2 + 1) \cdot (a, -a) - x \cdot (b - 1, 1 - b)$$

$$= (x + 1, ax^2 + bx - x + a).$$

Notice that the result is also an element in $\mathbf{M}(F, A_i)$, since $(a, -a), (b - 1, 1 - b) \in \mathbf{M}(F, A_i)$.

4. A NEW ALGORITHM FOR COMPUTING A COMPREHENSIVE GRÖBNER BASIS

The algorithm proposed in [9] for computing a comprehensive Gröbner system is adapted so as to work on the tuple representation of polynomials. The output of the new algorithm is a comprehensive Gröbner system with the property that every branch is disjoint vis a vis specializations, and the output along each branch is a Gröbner basis for $\langle F \rangle$ under the specialization. Tuple representation of the output is used to extract polynomials from $\langle F \rangle$. Hence a faithful comprehensive Gröbner basis can be found by taking the union of the outputs along all branches. The correctness and termination of the new algorithm that outputs a comprehensive Gröbner system as well as a comprehensive Gröbner basis follow from the correctness and termination of the algorithm proposed in [9] for computing a comprehensive Gröbner system.

The algorithm for computing a comprehensive Gröbner system in [9] uses the following theorem; the definition below is used in this theorem.

Definition 4.1 (Minimal Dickson Basis) *For a polynomial set G in $k[U,X]$ and an admissible block order with $U \ll X$, we say $F \subset k[U,X]$, denoted by* MDBasis(G)*, is a minimal Dickson basis of G, if*

1. *F is a subset of G,*
2. *for every polynomial $g \in G$, there is some polynomial $f \in F$ such that $\mathrm{lpp}_X(g)$ is a multiple of $\mathrm{lpp}_X(f)$, i.e. $\langle \mathrm{lpp}_X(F)\rangle = \langle \mathrm{lpp}_X(G)\rangle$, and*
3. *for any two distinct $f_1, f_2 \in F$, neither $\mathrm{lpp}_X(f_1)$ is a multiple of $\mathrm{lpp}_X(f_2)$ nor $\mathrm{lpp}_X(f_2)$ is a multiple of $\mathrm{lpp}_X(f_1)$.*

A minimal Dickson basis of a set may not be unique.

Theorem 4.2 (Kapur-Sun-Wang, 2010) *Let G be a Gröbner basis of the ideal $\langle F\rangle \subset k[U,X]$ w.r.t. an admissible block order with $U \ll X$. Let $G_r = G \cap k[U]$ and $G_m =$* MDBasis$(G \setminus G_r)$*. If σ is a specialization from $k[U]$ to L such that*

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g\in G_m} \mathrm{lc}_X(g) \in k[U]$,*

then $\sigma(G_m)$ is a (minimal) Gröbner basis of $\langle\sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$.

The theorem below serves as a basis of the proposed algorithm for computing a comprehensive Gröbner basis. The set E below refers to the set of equality constraints. It establishes that along a branch, for a specialization satisfying E, the first components of the tuple representation of polynomials constitute a comprehensive Gröbner basis of $\langle F\rangle$ and furthermore, every polynomial obtained by adding the two components in the tuple representation is in $\langle F\rangle$ ensuring faithfulness.

Theorem 4.3 *Let F be a set of polynomials in $k[U,X]$, E be a subset of $k[U]$, and $\mathbf{M}$ be a $k[U,X]$-module generated by $\{(f,0) \mid f \in F\} \cup \{(e,-e) \mid e \in E\}$. Suppose $\mathbf{G}$ is a Gröbner basis of the module $\mathbf{M}$ w.r.t. an order extended from $\prec_{X,U}$ in a position over term fashion with $(0,1) \prec (1,0)$, where $\prec_{X,U}$ is an admissible block order with $U \ll X$.*

Denote $G^{1st} = \{g \mid (g,\bar g) \in \mathbf{G}\}$, $G_r = G^{1st} \cap k[U]$ and $G_m =$ MDBasis$(G^{1st} \setminus G_r)$*. $\mathbf{G}_m$ is a subset of $\mathbf{G}$ such that $\{(g,\bar g) \in \mathbf{G}_m \mid g \in G_m\}$. If σ is a specialization from $k[U]$ to L such that*

1. *$\sigma(g) = 0$ for $g \in G_r$, and*
2. *$\sigma(h) \neq 0$, where $h = \prod_{g\in G_m} \mathrm{lc}_X(g) \in k[U]$,*

then

(1). for each $(g,\bar g) \in \mathbf{G}_m$, $g + \bar g \in \langle F\rangle$ and $\sigma(\bar g) = 0$, and

(2). $\{\sigma(g+\bar g) \mid (g,\bar g) \in \mathbf{G}_m\}$ is a minimal Gröbner basis of $\langle\sigma(F)\rangle$ in $L[X]$ w.r.t. $\prec_X$.

That is, $\{(V(G_r) \setminus V(h), G_m)\}$ is comprehensive Gröbner system on $V(G_r) \setminus V(h)$ for F, and $\{g + \bar g \mid (g,\bar g) \in \mathbf{G}_m\}$ is a comprehensive Gröbner basis on $V(G_r) \setminus V(h)$ for F.

PROOF. For (1), we first show $E \subset \langle G_r\rangle$. Since $\mathbf{G}$ is a Gröbner basis of the module $\mathbf{M}$ generated by $\{(f,0) \mid f \in F\} \cup \{(e,-e) \mid e \in E\}$ w.r.t. an order extended from $\prec_{X,U}$ in a POT fashion with $(0,1) \prec (1,0)$, we next show G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$ w.r.t. $\prec_{X,U}$. For any $h \in \langle F \cup E\rangle$, we have $h = \sum_{f\in F} p_f f + \sum_{e\in E} q_e e$ where $p_f, q_e \in k[U,X]$, so $(h, -(\sum_{e\in E} q_e e)) = \sum_{f\in F} p_f (f,0) + \sum_{e\in E} q_e (e,-e) \in \mathbf{M}$. As $\mathbf{G}$ is a Gröbner basis for $\mathbf{M}$, there exists $(g,\bar g) \in \mathbf{G}$ such that $\mathrm{lpp}(g)$ divides $\mathrm{lpp}(h)$, which means G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$. Besides, $G_r = G^{1st} \cap k[U] \subset \langle F \cup E\rangle$, so we have $E \subset \langle G_r\rangle \subset k[U]$ since $\prec_{X,U}$ is a block order with $U \ll X$.

Notice that $\mathbf{G}_m$ is a subset of the module $\mathbf{M}$; for each $(g,\bar g) \in \mathbf{G}_m$, we have

$$\begin{pmatrix} g \\ \bar g \end{pmatrix} = \sum_{f\in F} p_f \begin{pmatrix} f \\ 0 \end{pmatrix} + \sum_{e\in E} q_e \begin{pmatrix} e \\ -e \end{pmatrix},$$

where $p_f, q_e \in k[U,X]$. So $g+\bar g = (\sum_{f\in F} p_f f + \sum_{e\in E} q_e e) + \sum_{e\in E} q_e(-e) = \sum_{f\in F} p_f f \in \langle F\rangle$, and $\bar g = \sum_{e\in E} q_e(-e)$. Since $E \subset \langle G_r\rangle$, then $\sigma(\bar g) = 0$.

For (2), G^{1st} is a Gröbner basis for the ideal $\langle F \cup E\rangle$ w.r.t. $\prec_{X,U}$ as shown above, $G_r = G^{1st} \cap k[U]$ and $G_m =$ MDBasis$(G^{1st} \setminus G_r)$, so $\sigma(G_m) = \{\sigma(g+\bar g) \mid (g,\bar g) \in \mathbf{G}_m\}$ is a minimal Gröbner basis of $\langle\sigma(F)\rangle$ by Theorem 4.2.

Therefore, combined with (1) and (2), $\{(V(G_r)\setminus V(h), G_m)\}$ is comprehensive Gröbner system on $V(G_r)\setminus V(h)$ for F, and $\{g + \bar g \mid (g,\bar g) \in \mathbf{G}_m\}$ is a comprehensive Gröbner basis on $V(G_r) \setminus V(h)$ for F. ∎

We emphasize that, in the above theorem, we do not necessarily need to compute a whole Gröbner basis for the module $\mathbf{M}$, what we really need is a $\mathbf{G} \subset \mathbf{M}$ such that $G^{1st} = \{g \mid (g,\bar g) \in \mathbf{G}\}$ is a Gröbner basis for the ideal $\langle F \cup E\rangle$.

4.1 Algorithm

Now, we give an algorithm for computing comprehensive Gröbner bases. The correctness of the algorithm is a direct result of the above theorem. Its termination also can be proved in a same way as in [9].

In order to keep the presentation simple, we have deliberately avoided tricks and optimizations such as factoring h below. All the tricks suggested in [9] can be used here as well. In fact, our implementation incorporates fully these optimizations.

The following algorithm computes a comprehensive Gröbner basis on $V(E) \setminus V(N)$ for $F \subset k[U,X]$.

Algorithm CGB(E,N,F)
Input: (E,N,F): E, N, finite subsets of $k[U]$; F, a finite subset of $k[U,X]$.
Output: a comprehensive Gröbner basis of the set F on $V(E) \setminus V(N)$.

1. $\mathcal{CGS}$:= CGSMain(E,N,F), where $\mathcal{CGS}$ is a finite set of 3-tuples $(E_i, N_i, \mathbf{G}_i)$ such that $\{(V(E_i)\setminus V(N_i), G_i^{1st})\}$, where $G_i^{1st} = \{g \mid (g,\bar g) \in \mathbf{G}_i\}$, constitutes a comprehensive Gröbner system on $V(E)\setminus V(N)$ for F, and for each $(g,\bar g) \in \mathbf{G}_i$, $g + \bar g \in \langle F\rangle$ and $\sigma(\bar g)$ is 0 for every parameter specialization σ from $V(E_i) \setminus V(N_i)$.
2. Return $\{g + \bar g \mid (g,\bar g) \in \mathbf{G}_i$ for all $i\}$.

Below we assume that all Gröbner basis computations are done in $(k[U,X])^2$ using the order extended by $\prec_{X,U}$ in a POT fashion with $(1,0) \succ (0,1)$.

Algorithm CGSMain(E,N,F)
Input: (E,N,F): E, N, finite subsets of $k[U]$; F, a finite subset of $(k[U,X])^2$.
Output: $\mathcal{CGS}$: a finite set of 3-tuples $(E_i,N_i,\mathbf{G}_i)$ such that $\{(V(E_i)\backslash V(N_i), G_i^{1st})\}$, where $G_i^{1st} = \{g \mid (g,\bar{g}) \in \mathbf{G}_i\}$, constitutes a comprehensive Gröbner system on $V(E)\backslash V(N)$ for F, and for each $(g,\bar{g}) \in \mathbf{G}_i$, $g+\bar{g} \in \langle F\rangle$ and $\sigma(\bar{g})$ is 0 for every parameter specialization σ from $V(E_i) \setminus V(N_i)$.

1. If inconsistent(E, N), then return $\emptyset$.
2. Otherwise, $\mathbf{G}_0$:= ReducedGröbnerBasis $(\{(f,0) \mid f \in F\} \cup \{(e,-e) \mid e \in E\})$.
3. $\mathbf{G} := \mathbf{G}_0 \setminus \{(g,\bar{g}) \in \mathbf{G}_0 \mid g = 0\}$ and $G^{1st} := \{g \mid (g,\bar{g}) \in \mathbf{G}\}$.
4. If there exists $(1,\bar{g}) \in \mathbf{G}$, then return $\{(E,N,\{(1,\bar{g})\})\}$.
5. Let $\mathbf{G}_r := \{(g,\bar{g}) \in \mathbf{G} \mid g \in k[U]\}$ and $G_r := \{g \mid (g,\bar{g}) \in \mathbf{G}_r\}$.
6. If inconsistent(E, $G_r \times$ N), then $\mathcal{CGS} := \emptyset$, else $\mathcal{CGS} := \{(E, G_r \times N, \mathbf{G}_r)\}$.
7. If inconsistent(G_r, N), then return $\mathcal{CGS}$.
8. Otherwise, let G_m := MDBasis$(G^{1st} \setminus G_r)$ and $\mathbf{G}_m := \{(g,\bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r \mid g \in G_m\}$.
9. if consistent(G_r, N$\times\{$h$\}$), then $\mathcal{CGS} := \mathcal{CGS} \cup \{(G_r, N \times \{h\}, \mathbf{G}_m)\}$, where $h = \mathrm{lcm}\{h_1,\cdots,h_k\}$ and $\{h_1,\cdots,h_k\} = \{\mathrm{lc}_X(g) \mid g \in G_m\}$.
10. Return $\mathcal{CGS} \cup \bigcup_{h\in[h_1,\cdots,h_k]}$ CGSMain$(G_r \cup \{h_i\}, N \times \{h_1h_2\cdots h_{i-1}\}, \{g+\bar{g} \mid (g,\bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r\})$.

In the above algorithm, $A \times B = \{fg \mid f \in A, g \in B\}$. Also, for the case $i = 1$, $N \times \{h_1h_2\cdots h_{i-1}\} = N$. inconsistent(E, N) returns true if $V(E)\backslash V(N)$ is empty, false otherwise. The above steps 2 and 3 present a method to get $\mathbf{G}$ such that G^{1st} is a Gröbner basis for $\langle F \cup E\rangle$. We can also get such $\mathbf{G}$ by using Suzuki-Sato's trick in [15]. The inconsistency check is performed using techniques discussed in detail in [9]; their discussion is omitted here because of lack of space.

Compared with Suzuki-Sato's algorithm for computing a comprehensive Gröbner basis [15], the new algorithm has three advantages, most of which are inherited from our algorithm for computing a comprehensive Gröbner system [9]. First, as should be evident from the description, polynomials are never generated for the case when $V(E)\backslash V(N)$ is empty; so many useless computations are avoided. Second, recursive calls on the CGSMain are made only for the cases when the leading coefficients of G_m are nonzero instead of having to consider the leading coefficients of the whole $G^{1st} \setminus G_r$; thus many unnecessary branches are avoided, because typically, the size of G_m is smaller than the size of $G^{1st} \setminus G_r$. Finally, while recursively calling the CGSMain function, the intermediate result $\{g+\bar{g} \mid (g,\bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r\}$ is used in the new algorithm, instead of using the original F as input as in the Suzuki-Sato's algorithm, which should also contribute to the speedup of the proposed algorithm. Because of these advantages, the proposed algorithm has a much better performance than the Suzuki-Sato algorithm as well as other existing algorithms, as shown in the experimental results in section 6.

5. A SIMPLE EXAMPLE

The proposed algorithm is illustrated using the same example discussed in [9] primarily to help an interested reader to see the differences between the algorithm in [9] and the new algorithm of this paper. The discussion here is however self-contained.

Example 5.1 *Let $F = \{ax-b, by-a, cx^2-y, cy^2-x\} \subset \mathbb{Q}[a,b,c][x,y]$, with the block order $\prec_{X,U}$, $\{a,b,c\} \ll \{x,y\}$; within each block, $\prec_X$ and $\prec_U$ are graded reverse lexicographic orders with $y < x$ and $c < b < a$, respectively.*

At the beginning, $F = \{ax-b, by-a, cx^2-y, cy^2-x\}$, $E = \emptyset$ and $N = \{1\}$. We compute a comprehensive Gröbner system for $\{(f,0) \mid f \in F\} \in (\mathbb{Q}[a,b,c][x,y])^2$ using the tuple representation, so that along every branch, for every polynomial in a Gröbner basis, we can also extract the original polynomial from the input ideal generated by F to maintain faithfulness of the output.

(1) The set $V(E)\backslash V(N)$ is not empty. The reduced Gröbner basis of the $\mathbb{Q}[a,b,c][x,y]$-module $\langle (f,0) \mid f \in F\rangle \subset (\mathbb{Q}[a,b,c][x,y])^2$ w.r.t. the order extended by $\prec_{X,U}$ in POT fashion, is

$$\mathbf{G}_0 = \mathbf{G} = \{(x^3-y^3,0), (cx^2-y,0), (ay^2-bc,0), (cy^2-x,0),$$

$$(ax-b,0), (bx-acy,0), (a^2y-b^2c,0), (by-a,0), (a^6-b^6,0),$$

$$(a^3c-b^3,0), (b^3c-a^3,0), (ac^2-a,0), (bc^2-b,0)\},$$

with $\mathbf{G}_r = \{(g,\bar{g}) \in \mathbf{G} \mid g \in \mathbb{Q}[a,b,c]\} = \{(a^6-b^6,0), (a^3c-b^3,0), (b^3c-a^3,0), (ac^2-a,0), (bc^2-b,0)\}$. Denote $G^{1st} = \{g \mid (g,\bar{g}) \in \mathbf{G}\}$ and $G_r = \{g \mid (g,\bar{g}) \in \mathbf{G}_r\}$.

It is easy to see that $(V(E)\setminus V(G_r))\setminus V(N)$ is not empty. This implies that $\{\emptyset, G_r, \mathbf{G}_r\}$ is a trivial branch of the comprehensive Gröbner system for F.

(2) $G^{1st}\backslash G_r = \{x^3-y^3, cx^2-y, ay^2-bc, cy^2-x, ax-b, bx-acy, a^2y-b^2c, by-a\}$; G_m = MDBasis$(G^{1st} \setminus G_r) = \{bx-acy, by-a\}$ and $\mathbf{G}_m = \{(bx-acy,0), (by-a,0)\}$. Further, $h = \mathrm{lcm}\{\mathrm{lc}_X(bx-acy), \mathrm{lc}_X(by-a)\} = b$. This gives us another branch of comprehensive system for F corresponding to the case when all polynomials in G_r are 0 and $b \neq 0$: $(G_r, \{b\}, \mathbf{G}_m)$. Notice that $V(G_r) \setminus V(b)$ is not empty.

(3) The next branch to consider is when $b = 0$. The Gröbner basis of $G_r \cup \{b\}$ is $\{a^3, ac^2-a, b\}$, which is the input E' in the recursive call of CGSMain, with the other input being $N' = \{1\}$ and $F' = \{g+\bar{g} \mid (g,\bar{g}) \in \mathbf{G} \setminus \mathbf{G}_r\}$.

Since $V(E')\setminus V(N')$ is not empty, we can compute the reduced Gröbner basis for $\{(f,0) \mid f \in F'\} \cup \{(a^3,-a^3), (ac^2-a,-ac^2+a), (b,-b)\}$. By removing the tuples whose first component is 0, we get $\mathbf{G}' = \{(x^3-y^3,0), (cx^2-y,0), (cy^2-x,0), (a,-by), (b,-b)\}$ of which $\mathbf{G}'_r = \{(a,-by), (b,-b)\}$. Similarly, denote $G'^{1st} = \{g \mid (g,\bar{g}) \in \mathbf{G}'\}$ and $G'_r = \{g \mid (g,\bar{g}) \in \mathbf{G}'_r\}$. It is easy to check the set $V(E') \setminus V(G'_r)$ is empty, so no element in $\mathbf{G}'_r$ contributes to the comprehensive Gröbner system.

Next, $G'_m = \{cx^2-y, cy^2-x\}$, $\mathbf{G}'_m = \{(cx^2-y,0), (cy^2-x,0)\}$ and $h' = \mathrm{lcm}(\mathrm{lc}_X(cx^2-y), \mathrm{lc}_X(cy^2-x)) = c$. This results in another branch: $(G'_r, \{c\}, \mathbf{G}'_m)$.

(4) For the case when $h' = c = 0$, the set $E'' = \{a, b, c\}$ which is the Gröbner basis of $G'_r \cup \{c\}$. $N'' = \{1\}$ and $F'' = \{x^3 - y^3, cx^2 - y, cy^2 - x\}$. Computing the reduced Gröbner basis for $\{(f, 0) \mid f \in F''\} \cup \{(a, -a), (b, -b), (c, -c)\}$ and removing the tuples whose first component is 0, we get $\mathbf{G}'' = \{(x, -cy^2), (y, -cx^2), (a, -a), (b, -b), (c, -c)\}$. Then, $\mathbf{G}''_r = \{(a, -a), (b, -b), (c, -c)\}$, $G_m = \{x, y\}$ and $\mathbf{G}''_m = \{(x, -cy^2), (y, -cx^2)\}$. Further, $h'' = \mathrm{lcm}(\mathrm{lc}_X(x), \mathrm{lc}_X(y)) = 1$. Similarly, denote G'' and G''_r as before. This gives the last branch: $(G''_r, \{1\}, \mathbf{G}''_m)$.

Since $h'' = 1$, no more branches are generated and the algorithm terminates. Thus, we obtain a comprehensive Gröbner system for F:

$$\begin{cases} \{(a^6 - b^6, 0), (a^3c - b^3, 0), & \text{if } a^6 - b^6 \neq 0 \text{ or } a^3c - b^3 \neq 0 \\ (b^3c - a^3, 0), (ac^2 - a, 0), & \text{or } b^3c - a^3 \neq 0 \text{ or } ac^2 - a \neq 0 \\ (bc^2 - b, 0)\}, & \text{or } bc^2 - b \neq 0, \\ \{(bx - acy, 0), (by - a, 0)\}, & \text{if } a^6 - b^6 = a^3c - b^3 \\ & = b^3c - a^3 = ac^2 - a \\ & = bc^2 - b = 0 \text{ and } b \neq 0, \\ \{(cx^2 - y, 0), (cy^2 - x, 0)\} & \text{if } a = b = 0 \text{ and } c \neq 0, \\ \{(x, -cy^2), (y, -cx^2)\} & \text{if } a = b = c = 0. \end{cases}$$

An interested reader would observe comparing the above output with the output from [9] that except for the last branch, the outputs are the same. In [9], the last branch for the case when $a = b = c = 0$, the Gröbner basis is: $\{x, y\}$, whereas in the above the Gröbner basis is: $\{x - cy^2, y - cx^2\}$, when the tuple representation is replaced by the corresponding polynomials from the ideal of F. $x - cy^2$ is the faithful polynomial from the ideal of F corresponding to the output element x in [9]; similarly, $y - cx^2$ is the faithful polynomial corresponding to y.

A comprehensive Gröbner basis of F, after removing the duplicate ones, can be obtained directly from the above comprehensive Gröbner system. That is $\{a^6 - b^6, a^3c - b^3, b^3c - a^3, ac^2 - a, bc^2 - b, bx - acy, by - a, cx^2 - y, cy^2 - x\}$.

6. IMPLEMENTATION AND COMPARATIVE PERFORMANCE

The proposed algorithm has been implemented on the computer algebra system *Singular*. The implementation has been experimented on a number of examples from different application domains including geometry theorem proving and computer vision, and it has been compared with implementations of other algorithms. Since the proposed algorithm uses the new technique and basic module operations, it is efficient and can compute comprehensive Gröbner basis for most problems in a few seconds. In particular, we have been successful in solving the famous P3P problem for pose-estimation from computer vision, which is investigated by Gao et al [6] using the characteristic set method; see the polynomial system below.

The following table shows a comparison of our implementation on Singular with other existing algorithms for computing comprehensive Gröbner bases, including: Suzuki-Sato algorithm implemented by Nabeshima in Risa/Asir (package PGB, ver20090915) and the function "cgb" for computing comprehensive Gröbner bases in Reduce (package RedLog). The versions of Singular, Risa/Asir and Reduce are ver3-1-2, ver20090715 and free CSL version, respectively.

The implementation has been tried on many examples including Examples F6 and F8 from [14]. Many of these examples could be solved very quickly. To generate complex examples, we modified problems F2, F3, F4, F5 and F8 in [14], and labeled them as S1, S2, S3, S4 and S5. As stated above, we also tried the famous P3P problem from computer vision. The polynomials for these problems are given below:

F6: $F = \{x^4 + ax^3 + bx^2 + cx + d, 4x^3 + 3ax^2 + 2bx + c\}, X = \{x\}, U = \{a, b, c, d\}$;

F8: $F = \{ax^2 + by, cw^2 + z, (x - z)^2 + (y - w)^2, 2dxw - 2by\}, X = \{x, y, z, w\}, U = \{a, b, c, d\}$;

S1: $F = \{ax^2y^3 + by + y, x^2y^2 + xy + 2x, ax^2 + by + 2\}, X = \{x, y\}, U = \{a, b, c\}$;

S2: $F = \{ax^4 + cx^2 + y, bx^3 + x^2 + 2, cx^2 + dx + y\}, X = \{x, y\}, U = \{a, b, c, d\}$;

S3: $F = \{ax^3y + cxz^2, x^4y + 3dy + z, cx^2 + bxy, x^2y^2 + ax^2, x^5 + y^5\}, X = \{x, y, z\}, U = \{a, b, c, d\}$;

S4: $F = \{ax^2y + bx + y^3, ax^2y + bxy + cx, y^2 + bx^2y + cxy\}, X = \{x, y\}, U = \{a, b, c\}$;

S5: $F = \{ax^2 + byz + c, cw^2 + by + z, (x - z)^2 + (y - w)^2, 2dxw - 2byz\}, X = \{x, y, z, w\}, U = \{a, b, c, d\}$;

P3P: $F = \{(1 - a)y^2 - ax^2 - py + arxy + 1, (1 - b)x^2 - by^2 - qx + brxy + 1\}, X = \{x, y\}, U = \{p, q, r, a, b\}$.

For all these examples, the term orders used on X are graded reverse lexicographic orders.

Table 1: Timings

Exa.	Algorithm	time(sec.)	#polys
F6	New(S)	0.310	7
	cgb(R)	0.590	6
	SuzukiSato(A)	*error*	–
F8	New(S)	0.650	28
	cgb(R)	> 1h	–
	SuzukiSato(A)	0.6708	284
S1	New(S)	0.120	8
	cgb(R)	> 1h	–
	SuzukiSato(A)	*error*	–
S2	New(S)	0.165	9
	cgb(R)	10.520	38
	SuzukiSato(A)	*error*	–
S3	New(S)	4.515	62
	cgb(R)	28.845	84
	SuzukiSato(A)	> 1h	–
S4	New(S)	5.410	27
	cgb(R)	50.180	39
	SuzukiSato(A)	> 1h	–
S5	New(S)	18.034	58
	cgb(R)	329.169	59
	SuzukiSato(A)	> 1h	–
P3P	New(S)	14.440	50
	cgb(R)	> 1h	–
	SuzukiSato(A)	> 1h	–

In Table 1, entries labelled with New(S) is the proposed algorithm implemented in Singular; (R) and (A) stand for Reduce and Risa/Asir, respectively. The column "#polys" is the number of polynomials in the comprehensive Gröbner basis output by the implementations. The label "*error*"

is included if an implementation ran out of memory or broke down. The timings were obtained by running the implementations on Core i5 4 × 2.8GHz with 4GB memory running Windows 7.

As is evident from Table 1, the proposed algorithm has better performance in contrast to other algorithms.

7. CONCLUDING REMARKS

In this paper, we have adapted the algorithm proposed in [9] for computing a comprehensive Gröbner system of a parameterized polynomial system F such that the new algorithm not only produces a comprehensive Gröbner system of F but it also generates a comprehensive Gröbner basis of F. The main idea is to use polynomials from the ideal generated by F during the computation along various branches corresponding to constructible sets specializing parameters in the algorithm in [9]. Polynomials from $\langle F \rangle$ are represented as tuples, with the first component corresponding to its nonzero part under the specialization and the second component being zero under the specialization. The key steps of a Gröbner basis computation including reduction of a polynomial by another polynomial and S-polynomial construction, are performed on these tuple representations; these steps can also be viewed as computing Gröbner basis of a submodule over $R[X]^2$.

The new algorithm produces a comprehensive Gröbner system, in which each branch is a finite set of tuples along a constructible set (which is specified by a finite set of equalities over parameters and a finite set of disequalities over parameters), with the properties (i) the constructible sets constitute a partition over the set of parameter specializations under consideration, and (ii) for every parameter specialization in the constructible set of the branch, the second component of every tuple is 0 under the specialization and the leading coefficient of the first component in every tuple is nonzero under the specialization, and most importantly, (iii) the sum of the first component and the second component in the tuple is in the ideal generated by the input F. For generating a comprehensive Gröbner system, the second component of these tuples do not give any useful information and can hence be discarded. Using these second components however, a comprehensive Gröbner basis is the union over every branch of the set of polynomials obtained by adding the two components of each tuple. Further, such a comprehensive Gröbner basis is faithful since all the polynomials in the basis are also in the ideal.

The above construction can be used to adapt all known algorithms for computing a comprehensive Gröbner system. We believe that various optimization criteria to discard redundant computations can also be integrated in the proposed algorithm.

Using insights discussed above, we are investigating the design of a new algorithm for computing a minimal comprehensive Gröbner basis of a parametric polynomial systems, which will be reported in a forthcoming paper. Using this notion, we are able to define a canonical minimal comprehensive Gröbner basis, unlike the notion in Weispfenning [17], where a canonical comprehensive Gröbner basis is defined but it does not have the property of being minimal.

8. REFERENCES

[1] C. Chen, O. Golubitsky, F. Lemaire, M. Moreno Maza, and W. Pan. Comprehensive triangular decomposition. In Proceedings of CASC'07, Lect. Notes in Comp. Sci., Springer, Berlin, vol. 4770, 73-101, 2007.

[2] X.F. Chen, P. Li, L. Lin, and D.K. Wang. Proving geometric theorems by partitioned-parametric Gröbner bases. In Proceeding of Automated Deduction in Geometry (ADG) 2004, Lect. Notes in Comp. Sci., Springer, Berlin, vol. 3763, 34-43, 2005.

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

[4] B. Donald, D. Kapur, and J.L. Mundy(eds.). Symbolic and numerical computation for artificial intelligence. Computational Mathematics and Applications, Academic Press Ltd., London, 1992.

[5] X.S. Gao and S.C. Chou. Solving parametric algebraic systems. In Proceedings of ISSAC'1992, ACM Press, New York, 335–341, 1992.

[6] X.S. Gao, X.R. Hou, J.L. Tang, and H.F. Chen. Complete solution classification for the Perspective-Three-Point problem. IEEE Tran. on PAMI, vol. 25, no. 8, 930-943, 2003.

[7] D. Kapur. An approach for solving systems of parametric polynomial equations. Principles and Practice of Constraint Programming (eds. Saraswat and Van Hentenryck). MIT Press, Cambridge, 1995.

[8] D. Kapur. A quantifier-elimination based heuristic for automatically generating inductive assertions for programs. J. Syst. Sci. Complex., Vol. 19, No. 3, 307-330, 2006.

[9] D. Kapur, Y. Sun, and D.K. Wang. A new algorithm for computing comprehensive Gröbner systems. In Proceedings of ISSAC'2010, ACM Press, New York, 29-36, 2010.

[10] M. Manubens and A. Montes. Improving DISPGB algorithm using the discriminant ideal. J. Symb. Comp., 41, no. 11, 1245-1263. 2006.

[11] A. Montes. A new algorithm for discussing Gröbner basis with parameters. J. Symb. Comp., vol. 33, 1-2, 183-208, 2002.

[12] A. Montes and T. Recio. Automatic discovery of geometry theorems using minimal canonical comprehensive Gröbner systems. In Proceeding of Automated Deduction in Geometry (ADG) 2006, Lecture Notes in Artificial Intelligence, Springer, Berlin, Heidelberg, vol. 4869, 113-138, 2007.

[13] A. Montes and M. Wibmer. Gröbner bases for polynomial systems with parameters. J. Symb. Comp., vol. 45, no. 12, 1391-1425, 2010.

[14] K. Nabeshima. A speed-up of the algorithm for computing comprehensive Gröbner systems. In Proceedings of ISSAC'2007, ACM Press, New York, 299-306, 2007.

[15] A. Suzuki and Y. Sato. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In Proceedings of ISSAC'2006, ACM Press, New York, 326-331, 2006.

[16] V. Weispfenning. Comprehensive Gröbner bases. J. Symb. Comp., vol. 14, no. 1, 1-29, 1992.

[17] V. Weispfenning. Canonical comprehensive Gröbner bases. J. Symb. Comp., vol. 36, no. 3-4, 669-683, 2003.

[18] M. Wibmer. Gröbner bases for families of affine or projective schemes. J. Symb. Comp., vol. 42, no. 8, 803-834, 2007.