A New Incremental Algorithm for Computing Groebner Bases

Shuhong Gao [*] [†]
Dept. of Math. Sciences
Clemson University
Clemson, SC 29634-0975
sgao@clemson.edu

Yinhua Guan
Dept. of Math. Sciences
Clemson University
Clemson, SC 29634-0975
gguan@clemson.edu

Frank Volny IV
Dept. of Math. Sciences
Clemson University
Clemson, SC 29634-0975
fvolny@g.clemson.edu

ABSTRACT

In this paper, we present a new algorithm for computing Gröbner bases. Our algorithm is incremental in the same fashion as F5 and F5C. At a typical step, one is given a Gröbner basis G for an ideal $\mathbf{I}$ and any polynomial g, and it is desired to compute a Gröbner basis for the new ideal $\langle \mathbf{I}, g\rangle$, obtained from $\mathbf{I}$ by joining g. Let $(\mathbf{I} : g)$ denote the colon ideal of $\mathbf{I}$ divided by g. Our algorithm computes Gröbner bases for $\langle \mathbf{I}, g\rangle$ and $(\mathbf{I} : g)$ simultaneously. In previous algorithms, S-polynomials that reduce to zero are useless, in fact, F5 tries to avoid such reductions as much as possible. In our algorithm, however, these "useless" S-polynomials give elements in $(\mathbf{I} : g)$ and are useful in speeding up the subsequent computations. Computer experiments on some benchmark examples indicate that our algorithm is much more efficient (two to ten times faster) than F5 and F5C.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*Algebraic Algorithms*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*Computations on Polynomials*

Keywords

Gröbner basis, Buchberger's Algorithm, Colon ideal, F5 Algorithm

1. INTRODUCTION

In Buchberger's algorithm (1965, [1, 2, 3]), one has to reduce many "useless" S-polynomials (i.e. those that reduce to 0 via long division), and each reduction is time consuming. Faugère (1999, [9]) introduced a reduction method (F4) that can efficiently reduce many polynomials simultaneously; see also Joux and Vtse (2010, [11]) for a recent variant of F4. Lazard (1983, [13]) pointed out the connection between a Gröbner basis and linear algebra, that is, a Gröbner basis can be computed by Gauss elimination of Macaulay matrices (1902, [14]). This idea is implemented as XL type algorithms by Courtois et al. (2000,[5]), Ding et al. (2008, [7]), Mohammed et al. (2008–2009, [15, 16]), and Buchman et al. (2010, [4]). The linear algebra approach can be viewed as a fast reduction method. The main problem with these approaches is that the memory usage grows very quickly, and in practice the computation for even a small problem can not be done simply due to memory running out.

Faugère (2002, [10]) introduced the idea of signatures and rewriting rules that can detect many useless S-polynomials hence saving a significant amount of time that would be used in reducing them. By computer experiments, Faugère showed that his algorithm F5 is many times faster than previous algorithms. However, F5 seems difficult to both understand and implement. Eder and Perry (2009, [8]) simplified some of the steps in F5 and gave a variant called F5C which is almost always faster than F5. We should note that Sun and Wang (2009, [17]) also give a new proof and some improvement for F5.

Our main purpose of the current paper is to present a new algorithm that is both simpler and more efficient than F5 and F5C. Our algorithm is incremental just like F5 and F5C. Let $\mathbb{F}$ be any field and $R = \mathbb{F}[x_1, \cdots, x_n]$. Fix an arbitrary monomial order on R. At a typical iterative step, a Gröbner basis G for an ideal $\mathbf{I}$ in R is already computed, and it is desired to compute a Gröbner basis for the new ideal $\langle \mathbf{I}, g\rangle$ for a given polynomial $g \in R$. In F5, the basis G may not be reduced, thus containing many redundant polynomials. F5C is the same as F5 except that G is replaced by a reduced Gröbner basis in the next iterative step. Our algorithm will use a reduced Gröbner basis G as in F5C, but the crucial difference is that we introduce a so-called "super top-reduction" to detect "useless" polynomials. Furthermore, if there happens to be a polynomial that reduces to 0, it will be used to detect more "useless" polynomials. Hence reduction to 0 in our algorithm is not "useless" at all. In fact, it gives us a polynomial in the colon ideal

$$(\mathbf{I} : g) = \{u \in R : ug \in \mathbf{I}\}. \tag{1}$$

It is of independent interest to have an efficient algorithm for computing Gröbner bases for colon ideals of the form $(\mathbf{I} : g)$,

[*]The three authors were partially supported by National Science Foundation under grants DMS-0302549 and CCF-0830481, and National Security Agency under grant H98230-08-1-0030.
[†]We would also like to thank the referees for their very helpful comments and suggestions.

ISSAC 2010, 25–28 July 2010, Munich, Germany.

as it is a routine repeatedly used in primary decomposition, especially in separating components of different dimensions.

In Section 2, we shall present a relation between the Gröbner bases of $\langle \mathbf{I}, g\rangle$ and $(\mathbf{I} : g)$. This is based on the exact sequence of R-modules:

$$0 \longrightarrow R/(\mathbf{I} : g) \longrightarrow R/\mathbf{I} \longrightarrow R/\langle \mathbf{I}, g\rangle \longrightarrow 0$$

where the second morphism is defined by multiplication by g, which is injective by the definition in (1), and the third is the canonical morphism. The exactness of the sequence implies that

$$\dim_{\mathbb{F}}(R/\mathbf{I}) = \dim_{\mathbb{F}}(R/\langle \mathbf{I}, g\rangle) + \dim_{\mathbb{F}}(R/(\mathbf{I} : g)). \quad (2)$$

For an arbitrary ideal $\mathbf{I}$, we show in Section 2 how to compute $\mathbb{F}$-linear bases for all of these vector spaces from a given Gröbner basis for $\mathbf{I}$. In particular, we have the following result.

Theorem. *Suppose $\mathbf{I}$ is a zero-dimensional ideal in $R = \mathbb{F}[x_1, \cdots, x_n]$. Let $N = \dim_{\mathbb{F}}(R/\mathbf{I})$ (which is equal to the number of common solutions of $\mathbf{I}$ over the algebraic closure of $\mathbb{F}$, counting multiplicities). Then, given a Gröbner basis for $\mathbf{I}$ (under any monomial order) and a polynomial $g \in R$, Gröbner bases for $\langle \mathbf{I}, g\rangle$ and $(\mathbf{I} : g)$ can be computed deterministically using $O((nN)^3)$ operations in $\mathbb{F}$.*

The time complexity claimed by the theorem is of interest only when N is small compared to n (say $N = n^{\mathcal{O}(1)}$). For when N is large or ∞, we introduce an enhanced algorithm in Section 3. We shall define regular top-reductions and super top-reductions, as well as J-polynomials and J-signatures for any pair of polynomials. A J-polynomial means the joint of two polynomials, which is different from an S-polynomial but plays a similar role. Our algorithm is very similar to Buchberger's algorithm, where we replace S-polynomials by J-polynomials and "reduction" by "regular top-reduction". There are, however, two new features: (a) a super top-reduction is introduced to detect a useless J-polynomial, and (b) each reduction to zero gives a polynomial in $(\mathbf{I} : g)$ and is subsequently used in detecting future useless J-polynomials. We have implemented the resulting algorithm in Singular. In Section 4, we present some comparisons with F5 and F5C. Our computer experiments on several benchmark examples show that the new algorithm is more efficient, often two to ten times faster than F5 and F5C.

2. THEORY

We give a computational proof for the correspondence of linear bases for the equation (1) and the theorem mentioned in the previous section. The proof itself is more important than the theorem for our algorithm presented in the next section.

Let $\mathbf{I}$ be an arbitrary ideal in $R = \mathbb{F}[x_1, \ldots, x_n]$ and g any polynomial in R. Suppose we know a Gröbner basis G for $\mathbf{I}$ with respect to some monomial order $\prec$. Then we can find the standard monomial basis for $R/\mathbf{I}$:

$$B(\mathbf{I}) = \{x^{\alpha_1} = 1, x^{\alpha_2}, \ldots, x^{\alpha_N}\},$$

that is, $B(\mathbf{I})$ consists of all the monomials that are not reducible by $\mathrm{LM}(\mathbf{I})$.[1] Then $B(\mathbf{I})$ is a linear basis for $R/\mathbf{I}$ over $\mathbb{F}$. We assume the monomials in $B(\mathbf{I})$ are ordered in increasing order, that is, $x^{\alpha_i} \prec x^{\alpha_j}$ whenever $i < j$. Please note that when $\mathbf{I}$ is not 0-dimensional, N is ∞ and it is possible that there are infinitely many monomials between some two monomials in $B(\mathbf{I})$ (especially for lex order). The following proof is for an arbitrary ideal $\mathbf{I}$.

Suppose

$$\begin{pmatrix} x^{\alpha_1} \\ x^{\alpha_2} \\ \vdots \\ x^{\alpha_N} \end{pmatrix} \cdot g \equiv \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_N(x) \end{pmatrix} \pmod{G} \quad (3)$$

$$= A(x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_N})^T, \quad (4)$$

where $h_i \in \mathrm{span}_{\mathbb{F}}(x^{\alpha_1}, \ldots, x^{\alpha_N})$, $1 \le i \le N$, that is, each h_i is the normal form of $x^{\alpha_i} \cdot g \mod G$, and $A \in \mathbb{F}^{N\times N}$ is a matrix with the i^{th} row representing the coefficients of h_i, $1 \le i \le N$.

Note the matrix A in (4) has an important property that is useful for finding points (or solutions) of the algebraic variety defined by the ideal $\mathbf{I}$. In fact, when $\mathbf{I}$ is zero-dimensional, the eigenvalues of A correspond to the values of the polynomial g when evaluated at the points in the variety of $\mathbf{I}$ (and the corresponding eigenvectors are determined by the points alone, independent of g); for more details see Chapter 2 in [6].

Now apply the following row operations to both sides of (3) (equivalently (4)):

(R1) for $1 \le i < j \le N$ and $a \in \mathbb{F}$, subtract from the j^{th} row by the i^{th} row multiplied by a (i.e. $A_j := A_j - aA_i$),

(R2) for $a \in \mathbb{F}$ with $a \neq 0$, multiply the i^{th} row by a.

This means that we only apply row operations *downward* as one would perform Gauss elimination (to equation (4)) to get a triangular matrix. For example, suppose x^β is the leading monomial of $h_1(x)$. We can use $h_1(x)$ to eliminate the term x^β in all $h_j(x)$, $2 \le j \le N$. In fact, we only need to eliminate it if it's the leading term. Then continue with the leading monomial of the resulting $h_2(x)$ and so on. Since a monomial order is a well ordering, there is no infinite decreasing sequence of monomials, hence each $h_i(x)$ needs only be reduced by finitely many rows above it (even if there are infinitely many rows about the row of $h_i(x)$). Therefore, using downward row operations, the right hand side of (3) can be transformed into a quasi-triangular form, say

$$\begin{pmatrix} u_1(x) \\ u_2(x) \\ \vdots \\ u_N(x) \end{pmatrix} \cdot g \equiv \begin{pmatrix} v_1(x) \\ v_2(x) \\ \vdots \\ v_N(x) \end{pmatrix} \pmod{G}, \quad (5)$$

where $u_i(x)$ and $v_i(x)$ are in $\mathrm{span}_{\mathbb{F}}(x^{\alpha_1}, \ldots, x^{\alpha_N})$, and for each $1 \le i, j \le N$ with $v_i(x) \neq 0$ and $v_j(x) \neq 0$, we have $\mathrm{LM}(v_i(x)) \neq \mathrm{LM}(v_j(x))$, i.e. the nonzero rows of the right hand side have distinct leading monomials.

Since row operations are downward only, and the $B(\mathbf{I})$ are written in increasing order, we have that each $u_i(x)$ is monic and

$$\mathrm{LM}(u_i(x)) = x^{\alpha_i},\ 1 \le i \le N.$$

Let

$$G_0 = G \cup \{u_i(x) : 1 \le i \le N \text{ with } v_i(x) = 0\}, \text{ and}$$
$$G_1 = G \cup \{v_i(x) : 1 \le i \le N\}.$$

[1] We say that a polynomial f is reducible by a set of polynomials G if $\mathrm{LM}(f)$ is divisible by $\mathrm{LM}(g)$ for some $g \in G$.

Certainly, $G_1 \subseteq \langle \mathbf{I}, g\rangle$ and $G_0 \subseteq (\mathbf{I} : g)$ (as $u_i(x) \cdot g \in \mathbf{I}$ whenever $v_i(x) = 0$). We prove the following:

(a) G_0 is a Gröbner basis for $(\mathbf{I} : g)$, and

(b) G_1 is a Gröbner basis for $\langle \mathbf{I}, g\rangle$.

Since (5) is obtained from (3) by downward row operations, there is an upper triangular nonsingular matrix $M \in \mathbb{F}^{N\times N}$ (with each row containing only finitely many nonzero entries) such that

$$(u_1(x), \ldots, u_N(x))^T = M\,(x^{\alpha_1}, \ldots, x^{\alpha_N})^T,$$

and

$$(v_1(x), \ldots, v_N(x))^T = M\,(h_1(x), \ldots, h_N(x))^T.$$

Even though N could be infinite, M does have an inverse M^{-1} with each row containing only finitely many nonzero entries. For any $w(x) \in R/\mathbf{I}$, we can write it as

$$w(x) = \sum_{i=1}^{N} w_i x^{\alpha_i},\ w_i \in \mathbb{F}, \tag{6}$$

where there are only finitely many nonzero w_i's. Let

$$(c_1, \ldots, c_N) = (w_1, \ldots, w_N)M^{-1} \in \mathbb{F}^N.$$

Note that the vector $(c_1, \ldots, c_N)$ contains only finitely many nonzero entries, as it is a linear combination of finitely many rows of M^{-1}. Then we have

$$\begin{aligned} w(x) &= (w_1, \ldots, w_N)M^{-1}M(x^{\alpha_1}, \ldots, x^{\alpha_N})^T \\ &= (c_1, \ldots, c_N)(u_1(x), \ldots, u_N(x))^T, \end{aligned}$$

i.e.

$$w(x) = \sum_{i=1}^{N} c_i u_i(x), \tag{7}$$

and

$$\begin{aligned} w(x)\cdot g &= (w_1, \ldots, w_N)(x^{\alpha_1}, \ldots, x^{\alpha_N})^T \cdot g \\ &\equiv (w_1, \ldots, w_N)M^{-1}M(h_1(x), \ldots, h_N(x))^T \\ &= (c_1, \ldots, c_N)(v_1(x), \ldots, v_N(x))^T, \end{aligned}$$

i.e.

$$w(x)\cdot g \equiv \sum_{i=1}^{N} c_i v_i(x) \pmod{G}. \tag{8}$$

For (a), to prove that G_0 is a Gröbner basis for $(\mathbf{I} : g)$, it suffices to show that each $f \in (\mathbf{I} : g)$ can be reduced to zero by G_0 via long division. Indeed, for any $f \in (\mathbf{I} : g)$, since G is a Gröbner basis, f can be reduced by G to some $w(x)$ as in (6). Then, by (7) and (8), we have

$$f \equiv w(x) \equiv \sum_{i=1}^{N} c_i u_i(x) \pmod{G},$$

and

$$f \cdot g \equiv w \cdot g \equiv \sum_{i=1}^{N} c_i v_i(x) \pmod{G}.$$

As $f \in (\mathbf{I} : g)$, we have $f \cdot g \in \mathbf{I}$, so $f \cdot g \equiv 0 \pmod{G}$. This implies that $\sum_{i=1}^{N} c_i v_i(x) = 0$, hence $c_i = 0$ whenever $v_i(x) \neq 0$, as the nonzero $v_i(x)$'s have distinct leading monomials. Thus

$$f \equiv w(x) \equiv \sum_{u_i \in G_0} c_i u_i(x) \pmod{G}. \tag{9}$$

This implies that f can be reduced to 0 by G_0 via long division. Therefore, G_0 is a Gröbner basis for $(\mathbf{I} : g)$.

For (b), for any $f \in \langle \mathbf{I}, g\rangle$, there exists $w(x)$ of the form (6) such that

$$f \equiv w(x) \cdot g \pmod{G}.$$

By (8),

$$f \equiv w(x)\cdot g \equiv \sum_{i=1}^{N} c_i v_i(x) = \sum_{v_i(x)\neq 0} c_i v_i(x) \pmod{G}. \tag{10}$$

Hence f can be reduced to 0 by $G \cup \{v_i(x) : 1 \le i \le N\}$ via long division. This shows that G_1 is a Gröbner basis for $\langle \mathbf{I}, g\rangle$.

Now we explicitly describe $B(\mathbf{I} : g)$ and $B(\langle \mathbf{I}, g\rangle)$, the standard monomial bases for $R/(\mathbf{I} : g)$ and $R/\langle \mathbf{I}, g\rangle$, respectively. We first show that

$$B(\mathbf{I} : g) = \{x^{\alpha_j} : 1 \le j \le N \text{ and } v_j(x) \neq 0\}. \tag{11}$$

Since $\mathbf{I} \subseteq (\mathbf{I} : g)$, we have

$$B(\mathbf{I} : g) \subseteq B(\mathbf{I}) = \{x^{\alpha_1}, \ldots, x^{\alpha_N}\}.$$

Recall that $\mathrm{LM}(u_j(x)) = x^{\alpha_j},\ 1 \le j \le N$. For each $1 \le j \le N$, if $v_j(x) = 0$, then $u_j(x) \in G_0$, so $x^{\alpha_j} \notin B(\mathbf{I} : g)$. If $v_j(x) \neq 0$, we claim that there is no $f \in (\mathbf{I} : g)$ such that $\mathrm{LM}(f) = x^{\alpha_j}$. Suppose otherwise. Then $f \equiv w(x) \pmod{G}$ for some $w(x)$ as in (6) and $\mathrm{LM}(w(x)) = \mathrm{LM}(f) = x^{\alpha_j}$. By (9), x^{α_j} must be equal to the leading monomial of some $u_i(x) \in G_0$, hence $u_j(x) \in G_0$. This contradicts the assumption that $v_j(x) \neq 0$. Hence (11) holds.

Next we claim that

$$B(\langle \mathbf{I}, g\rangle) = B(\mathbf{I}) \setminus \{\mathrm{LM}(v_i(x)) : 1 \le i \le N\}. \tag{12}$$

This holds, as the equation (10) implies that the leading monomial of any $f \in \langle \mathbf{I}, g\rangle$ is either divisible by $\mathrm{LM}(G)$ or equal to some $\mathrm{LM}(v_i(x))$, where $v_i(x) \neq 0,\ 1 \le i \le N$.

Now back to the proof of the theorem. The equation (2) follows from the equations (11) and (12), as the leading monomials of the nonzero $v_i(x)$ are distinct and are contained in $B(\mathbf{I})$. When $\mathbf{I}$ is zero-dimensional, the normal forms $h_i(x)$ in (3) can be computed in time cubic in nN, say by using the border basis technique [12], and Gauss elimination also needs cubic time. Hence the claimed time complexity follows.

Finally, we make a few observations concerning the above proof. They will be the basis for our algorithm below.

- $\mathrm{LM}(u_i(x)) = x^{\alpha_i}$, so u_i is not divisible by $\mathrm{LM}(G)$, for all $1 \le i \le N$. The monomial x^{α_i} is an index for the corresponding row in (3), which will be called a signature.
- For any i with $v_i(x) \neq 0$, $\mathrm{LM}(u_i(x))$ is not divisible by $\mathrm{LM}(G_0)$. This follows from (11).
- In the process of computing the Gröbner bases, whenever we get some $u \cdot g \equiv 0 \pmod{G}$, we add u to G_0. So we never need to consider any u' such that $\mathrm{LT}(u')$ is divisible by $\mathrm{LT}(u)$.

- Both G_0 and G_1 have many redundant polynomials. We do not want to store most of them.

We need to decide which rows to store and how to perform row operations while many rows are missing. In the next section, we shall introduce regular top-reductions to emulate the row operations above and super top-reductions to detect rows that need not be stored.

3. ALGORITHM

Our algorithm computes a Gröbner basis for $(\mathbf{I} : g)$ in the process of computing a Gröbner basis for $\langle \mathbf{I}, g\rangle$. The Gröbner basis for $(\mathbf{I} : g)$ is stored in the list H in the algorithm described in figure 1. If one does not need a Gröbner basis for $(\mathbf{I} : g)$, one is free to retain only the leading monomials of H. This improves efficiency when only the Gröbner basis for $\langle \mathbf{I}, g\rangle$ is required. We provide Singular code for this version at `http://www.math.clemson.edu/~sgao/code/g2v.sing`.

Let $R = \mathbb{F}[x_1, \cdots, x_n]$ with any fixed monomial order $\prec$ as above. Let $G = \{f_1, f_2, \ldots, f_m\}$ be any given Gröbner basis for $\mathbf{I}$ and let $g \in R$. Consider all pairs $(u, v) \in R^2$ satisfying

$$ug \equiv v \pmod{G}. \tag{13}$$

Certainly, $G \subset \langle \mathbf{I}, g\rangle$ and $G \subset (\mathbf{I} : g)$. That is, we have the trivial solutions $(f_1, 0), (f_2, 0), \ldots, (f_m, 0)$ and

$$(0, f_1), (0, f_2), \ldots, (0, f_m). \tag{14}$$

The first nontrivial solution for (13) is $(1, g)$.

We need to introduce a few concepts before proceeding. For any pair $(u, v) \in R^2$, $\mathrm{LM}(u)$ is called the **signature** of (u, v). We make the convention that $\mathrm{LM}(0) = 0$. Our definition of signature is similar in purpose to that of Faugère [10]. To simulate the row operation (R1), we introduce the concept of regular top-reduction. Our regular top-reduction is similar to the top-reduction used by Faugère [10], but our use of super top-reduction below seems to be new. We say that (u_1, v_1) is **top-reducible** by (u_2, v_2) if

(i) $\mathrm{LM}(v_2) \mid \mathrm{LM}(v_1)$, and

(ii) $\mathrm{LM}(tu_2) \preceq \mathrm{LM}(u_1)$ where $t = \dfrac{\mathrm{LM}_{(v_1)}}{\mathrm{LM}_{(v_2)}}$.

The corresponding **top-reduction** is then

$$(u_1, v_1) - ct(u_2, v_2) \equiv (u_1 - ctu_2, v_1 - ctv_2) \pmod{G},$$

where $c = \mathrm{LC}(v_1)/\mathrm{LC}(v_2)$. The effect of a top-reduction is that the leading monomial in the v-part is canceled. A top-reduction is called **super**, if

$$\mathrm{LM}(u_1 - ctu_2) \prec \mathrm{LM}(u_1),$$

that is, the leading monomial in the u-part is also canceled. A super top-reduction happens when

$$\mathrm{LM}(tu_2) = \mathrm{LM}(u_1) \text{ and } \frac{\mathrm{LC}(u_1)}{\mathrm{LC}(u_2)} = \frac{\mathrm{LC}(v_1)}{\mathrm{LC}(v_2)}.$$

A top-reduction is called **regular** if it is not super. The signature is preserved by regular top-reductions, but not by super top-reductions.

In our algorithm, we only perform regular top-reductions. We also keep all the u monic (or 0 for trivial solutions). Hence, for each regular top-reduction of (u_1, v_1) by (u_2, v_2) where u_1 and u_2 are monic, we perform the following steps:

- $u := u_1 - ctu_2$, and $v := v_1 - ctv_2$ where $t = \dfrac{\mathrm{LM}_{(v_1)}}{\mathrm{LM}_{(v_2)}}$ and $c = \mathrm{LC}(v_1)/\mathrm{LC}(v_2)$;
- if $\mathrm{LM}(u_1) = t\mathrm{LM}(u_2)$ then $u := u/(1 - c)$ and $v := v/(1 - c)$;
- $u := \mathrm{Normal}(u, G)$ and $v := \mathrm{Normal}(v, G)$, the normal forms of u and v modulo G.

Note that, if $\mathrm{LM}(u_1) = t\mathrm{LM}(u_2)$ and $c = 1$, then (u_1, v_1) is super top-reducible by (u_2, v_2). We never perform super top-reductions in our algorithm. In the case that (u_1, v_1) is not regular top-reducible by other pairs known but is super top-reducible, we discard the pair (u_1, v_1), which corresponds to a row in the equation (5) that needs not be stored (in this case v_1 is redundant in G_1).

Now we introduce a new concept of so-called J-pair for any two pairs of polynomials. Initially, we have the trivial solution pairs in (14) and the pair

$$(1, v), \quad \text{where } v = \mathrm{Normal}(g, G), \text{ assuming } v \neq 0.$$

We find new solution pairs that are not top-reducible by the known pairs, hence must be stored. For any monomial t, consider the pair $t(1, v)$. If $t(1, v)$ is not top-reducible by any $(0, f)$ where $f \in G$, then $t(1, v) \bmod G$ is super top-reducible by $(1, v)$, hence we don't need to store this pair. However, if $t(1, v)$ top-reducible by some $(0, f)$ where $f \in G$, then the new pair after reduction by $(0, f)$ may not be top-reducible by $(1, v)$ any more, hence it must be stored. This means we find the smallest monomial t so that the pair $t(1, v)$ is top-reducible by some $(0, f)$. This can happen only if $t\mathrm{LM}(v)$ is divisible by $\mathrm{LM}(f)$ for some $f \in G$. Hence t should be such that $t\mathrm{LM}(v) = \mathrm{lcm}(\mathrm{LM}(v), \mathrm{LM}(f))$. We consider all these t given by $f \in G$. More generally, suppose we have computed a list of solution pairs

$$(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k), \tag{15}$$

including the pairs in (14). We consider all pairs $t(u_i, v_i)$, $1 \leq i \leq k$, that may be top-reducible by some pair in (15). The t must come from $\mathrm{lcm}(\mathrm{LM}(v_i), \mathrm{LM}(v_j))$ for some $j \neq i$. This leads us to the concept of a joint pair from any two pairs as defined below.

Let (u_1, v_1) and (u_2, v_2) be two pairs of polynomials with v_1 and v_2 both nonzero. Let

$$\mathrm{lcm}(\mathrm{LM}(v_1), \mathrm{LM}(v_2)) = t, \quad t_1 = \frac{t}{\mathrm{LM}(v_1)}, \quad t_2 = \frac{t}{\mathrm{LM}(v_2)}.$$

Find $\max(t_1\mathrm{LM}(u_1), t_2\mathrm{LM}(u_2))$, say equal to $t_i\mathrm{LM}(u_i)$. Then

- $t_i\mathrm{LM}(u_i)$ is called the **J-signature** of the two pairs;
- $t_i v_i$ is called the **J-polynomial** of the two pairs;
- $t_i(u_i, v_i) = (t_i u_i, t_i v_i)$ is called the **J-pair** of the two pairs;

where J means "joint". In comparison, the S-polynomial of v_1 and v_2 is $t_1 v_1 - (c_1/c_2) t_2 v_2$ where $c_i = \mathrm{LC}(v_i)$. Hence our J-polynomials are related to S-polynomials. Notice that the J-signature of (u_1, v_1) and (u_2, v_2) is the same as the signature of the J-pair of (u_1, v_1) and (u_2, v_2).

The basic idea of our algorithm is as follows. Initially, we have the pair $(1, g) \bmod G$ and the trivial pairs in (14). From these pairs, we form all J-pairs and store them in a list JP. Then take the smallest J-pair from JP and repeatedly

perform regular top-reductions until it is no longer regular top-reducible. If the v part of the resulting pair is zero, then the u part is a polynomial in $(\mathbf{I} : g)$, and we store this polynomial. If the v part is nonzero, then we check if the resulting J-pair is super top-reducible. If so, then we discard this J-pair; otherwise, we add this pair to the current Gröbner basis and form new J-pairs and add them to JP. Repeat this process for each pair in JP. The algorithm is described more precisely in Figure 1 below. In the algorithm, we include two options: in first option we only keep the leading monomials of u's and there is no need to update u's in each regular top-reduction, so we compute a Gröbner basis for $\mathrm{LM}(\mathbf{I} : g)$; in the second option, we actually update u in each regular top-reduction as specified above, so we compute a Gröbner basis for $(\mathbf{I} : g)$.

It can be proved that, when JP is empty, $\mathrm{LM}(H)$ is a Gröbner basis for $\mathrm{LM}(\mathbf{I} : g)$ and V is a Gröbner basis for $\langle \mathbf{I}, g \rangle$, which may not be minimal. Also, for each solution (u, v) to (13), we have either $\mathrm{LM}(u)$ is reducible by H, or (u, v) can be top-reduced to $(0, 0)$ by (U, V) (using both regular and super top-reductions). The proof of the algorithm will be included elsewhere for a more general version of this algorithm that needs not be incremental.

It should be remarked that in our algorithm we always pick the J-pair with minimal signature to reduce. This is to emulate the downward row operations of the matrix. The algorithm may not work if one uses another strategy, say picking J-pairs with minimal total degree in the v part.

4. COMPARISONS AND CONCLUSIONS

In order to determine how our algorithm compared to, say F5 and F5C, we computed Gröbner basis for various benchmark examples as provided in [8]. We used the examples and algorithm implementation for F5 and F5C provided by the URL in [8] which was all implemented in the Singular computer algebra system. Our implementation was meant to mirror the F5C implementation in terms of code structure and Singular kernel calls. For example, both implementations use the procedure "reduce" to compute normal form of a polynomial modulo a Gröbner basis. Reasonable differences were unavoidable though. For example, F5C uses Quicksort while $\mathrm{G}^2\mathrm{V}$ performs one step of a Mergesort in the function "insertPairs".

All examples considered were over the field of 7583 elements with the graded reverse lexicographic ordering. In addition to the usual wall clock times, several other measures of performance were considered, namely

1. Wall clock time (from a single run),
2. Extraneous generators,
3. Memory usage,
4. Count of J-pairs or S-pairs reduced, and
5. Count of normal forms computed.

The run-times and ratios of run-times are presented in Table 1. One can see that, for these examples, our algorithm is two to ten times faster than F5 and F5C.

F5, F5C and our algorithm $\mathrm{G}^2\mathrm{V}$ are all incremental. That is, given a list of polynomials $g_1, \ldots, g_m$, a Gröbner basis is computed for $\langle g_1, g_2, \ldots, g_i \rangle$ for $i = 1, 2, \ldots, m$. Hence, in each iteration, all three algorithms are given a polynomial

Test Case (#generators)	F5	F5C	$\mathrm{G}^2\mathrm{V}$
Katsura5 (22)	1.48	0.93	0.36
Katsura6 (41)	2.79	2.34	0.37
Katsura7 (74)	30.27	22.76	4.64
Katsura8 (143)	290.97	177.74	29.88
Schrans-Troost (128)	1180.08	299.65	21.34
F633 (76)	30.93	29.87	2.06
Cyclic6 (99)	28.44	22.06	5.65
Cyclic7 (443)	4591.20	2284.05	732.33

Test Case (#generators)	$\mathrm{F5}/\mathrm{G}^2\mathrm{V}$	$\mathrm{F5C}/\mathrm{G}^2\mathrm{V}$
Katsura5 (22)	4.11	2.58
Katsura6 (41)	7.54	6.32
Katsura7 (74)	6.52	4.91
Katsura8 (143)	9.74	5.95
Schrans-Troost (128)	55.30	14.04
F633 (76)	15.01	14.50
Cyclic6 (99)	5.03	3.90
Cyclic7 (443)	6.27	3.12

Table 1: Run-times in seconds and ratios of run-times for various test cases in Singular 3.1.0.6 on an Intel Core 2 Quad 2.66 GHz. The #generators refers to a reduced Gröbner basis.

Test Case (#generators)	F5	F5C	$\mathrm{G}^2\mathrm{V}$
Katsura5 (22)	61	44	63
Katsura6 (41)	74	65	52
Katsura7 (74)	185	163	170
Katsura8 (143)	423	367	335
Schrans-Troost (128)	643	399	189
F633 (76)	237	217	115
Cyclic6 (99)	202	183	146
Cyclic7 (443)	1227	1006	658

Table 2: The number of generators in the Gröbner basis in the last iteration but before computing a reduced Gröbner basis. Of course, F5 never computes the reduced Gröbner basis.

$g \in R$ and a Gröbner basis G for some ideal $\mathbf{I}$, and they compute a Gröbner basis for $\langle \mathbf{I}, g \rangle$. The computed Gröbner basis is not necessarily reduced, and any redundant polynomials in the basis will result in extra S-polynomials or J-polynomials to be reduced. Fewer generators at any given time means that fewer S-polynomials or J-polynomials need to be considered. F5 uses G as it was computed, so may not be reduced, however, F5C and our algorithm always replace G by a reduced Gröbner basis. Table 2 lists the number of polynomials in the Gröbner bases that were output by each algorithm on the last iteration of each example. Computation time is not the only limiting factor in a Gröbner basis computation. Storage requirements also limit computation. Table 3 lists the maximum amount of memory each algorithm needed in the processing of examples. Again, we cannot make generalizations from the memory results because this is only one possible implementation of each algorithm in one possible CAS.

The last two criteria were also measured, but the results were not nearly as interesting. Each algorithm outperformed the other (and usually not by much) in nearly half the examples.

Input:	$G = [f_1, f_2, \ldots, f_m]$, a Gröbner basis for an ideal $\mathbf{I}$, and g a polynomial.
Output:	A Gröbner basis for $\langle \mathbf{I}, g \rangle$, and a Gröbner basis for $\mathrm{LM}(\mathbf{I} : g)$ or for $(\mathbf{I} : g)$.
Variables:	U a list of monomials for $\mathrm{LM}(u)$ or of polynomials for u; V a list of polynomials for v; H a list for $\mathrm{LM}(u)$ or u so that $u \in (\mathbf{I} : g)$ found so far, JP a list of pairs (t, i), where t is a monomial so that $t(u_i, v_i)$ is the J-pair of (u_i, v_i) and (u_j, v_j) for some $j \neq i$. We shall refer (t, i) as the J-pair of (u_i, v_i) and (u_j, v_j).
Step 0.	$U = [0, \ldots, 0]$ with length m, and $V = [f_1, \ldots, f_m]$ (so that $(u_i, v_i) = (0, f_i)$, $1 \leq i \leq m$); $H = [\mathrm{LM}(f_1), \mathrm{LM}(f_2), \ldots, \mathrm{LM}(f_m)]$ or $H = [f_1, f_2, \ldots, f_m]$; Compute $v = \mathrm{Normal}(g, G)$; If $v = 0$, then append 1 to H and return V and H (stop the algorithm); else append 1 to U and v to V; $JP = [\,]$, an empty list; For each $1 \leq i \leq m$, compute the J-pair of the two pairs $(u_{m+1}, v_{m+1}) = (1, v)$ and $(u_i, v_i) = (0, f_i)$, such a J-pair must be of the form $(t_i, m+1)$, insert $(t_i, m+1)$ into JP whenever t_i is not reducible by H. (store only one J-pair for each distinct J-signature).
Step 1.	Take a minimal (in signature) pair (t, i) from JP, and delete it from JP.
Step 2.	Reduce the pair $t(u_i, v_i)$ repeatedly by the pairs in (U, V), using regular top-reductions, say to get (u, v), which is not regular top-reducible.
Step 3a.	If $v = 0$, then append $\mathrm{LM}(u)$ or u to H and delete every J-pair (t, ℓ) in JP whose signature $t\mathrm{LM}(u_\ell)$ is divisible by $\mathrm{LM}(u)$.
Step 3b.	If $v \neq 0$ and (u, v) is super top-reducible by some pair (u_j, v_j) in (U, V), then **discard** the pair (t, i).
Step 3c.	Otherwise, append u to U and v to V, form new J-pairs of (u, v) and (u_j, v_j), $1 \leq j \leq \#U - 1$, and insert into JP all such J-pairs whose signature are not reducible by H (store only one J-pair for each distinct J-signature).
Step 4.	While JP is not empty, go to step 1.
Return:	V and H.

Figure 1: Algorithm

Test Case (#generators)	F5	F5C	G^2V
Katsura5 (22)	1359	828	1255
Katsura6 (41)	1955	1409	1254
Katsura7 (74)	8280	4600	5369
Katsura8 (143)	40578	20232	20252
Schrans-Troost (128)	130318	50566	32517
F633 (76)	3144	2720	2824
Cyclic6 (99)	2749	2280	1789
Cyclic7 (443)	48208	23292	24596

Table 3: The maximum amount of memory (in KiB) Singular 3.1.0.6 used from startup to the conclusion of the Gröbner basis computation. Memory amounts obtained with "memory(2);".

In conclusion, we presented a precise relationship among the degrees of the ideals $\mathbf{I}$, $\langle \mathbf{I}, g \rangle$ and $(\mathbf{I} : g)$, and a connection between the Gröbner bases of $\langle \mathbf{I}, g \rangle$ and $(\mathbf{I} : g)$. This allowed us to design a new algorithm, which is conceptually simpler and yet more efficient than F5 and F5C.

5. REFERENCES

[1] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, PhD thesis, Innsbruck, 1965.

[2] B. Buchberger, "A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Basis," In Proc. EUROSAM 79 (1979), vol. 72 of Lect. Notes in Comp. Sci., Springer Verlag, 3–21.

[3] B. Buchberger, "Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory," In Recent trends in multidimensional system theory, Ed. Bose, 1985.

[4] J. Buchmann, D. Cabarcas, J. Ding, M. S. E. Mohamed, "Flexible Partial Enlargement to Accelerate Grobner Basis Computation over F2," AFRICACRYPT 2010, May 03-06, 2010, Stellenbosch, South Africa, to be published in LNCS by Springer

[5] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations," in Proceedings of International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT), volume 1807 of Lecture Notes in Computer Science, pages 392–407, Bruges, Belgium, May 2000. Springer.

[6] D. Cox, J. Little and D. O'Shea, *Using algebraic geometry*, Graduate Texts in Mathematics, 185. Springer-Verlag, New York, 1998.

[7] J. Ding, J. Buchmann, M. S. E. Mohamed, W. S. A. M. Mohamed, R. Weinmann, "Mutant XL," First International Conference on Symbolic Computation and Cryptography, SCC 2008

[8] C. Eder and J. Perry, "F5C: a variant of Faugère's F5 algorithm with reduced Gröbner bases," arXiv: 0906.2967v5, July 2009.

[9] J.-C. Faugère, "A new efficient algorithm for computing Gröbner bases (F4)," Effective methods in algebraic geometry (Saint-Malo, 1998), J. Pure Appl. Algebra 139 (1999), no. 1-3, 61–88.

[10] J.-C. Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)," Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, 75–83 (electronic), ACM, New York, 2002.

[11] A. Joux and V. Vitse, "A variant of the F4 algorithm," preprint 2010. http://eprint.iacr.org/2010/158.

[12] A. Kehrein and M. Kreuzer, "Computation of Border Bases," J. Pure Appl. Algebra 205 (2006), 279–295.

[13] D. Lazard, "Gaussian Elimination and Resolution of Systems of Algebraic Equations," in Proc. EUROCAL 83 (1983), vol. 162 of Lect. Notes in Comp. Sci, 146–157.

[14] F. Macaulay, "Some formulae in elimination," Proceedings of London Mathematical Society (1902), 3–38.

[15] M. S. E. Mohamed, D. D. Cabarcas, J. Ding, J. Buchmann and S. Bulygin, "MXL3: An efficient algorithm for computing Groebner bases of zero-dimensional ideals," the 12th International Conference on Information Security and Cryptology (ICISC 2009), Dec. 2009, Seoul, Korea. (To be published in LNCS by Springer).

[16] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, J. Buchmann, "MXL2: Solving Polynomial Equations over GF(2) Using an Improved Mutant Strategy," PQCrypto 2008: 203-215, LNCS 5299, Springer 2008

[17] Y. Sun and D.K. Wang, "A New Proof of the F5 Algorithm," preprint 2009. http://www.mmrc.iss.ac.cn/pub/mm28.pdf/06-F5Proof.pdf