Illinois Journal of Mathematics
Volume 56, Number 1, Spring 2012, Pages 177–194
S 0019-2082

EFFECTIVE COMPUTATION OF STRONG GRÖBNER BASES OVER EUCLIDEAN DOMAINS

DANIEL LICHTBLAU

Abstract. Buchberger and, independently, Kandri-Rody and Kapur, defined a strong Gröbner basis for a polynomial ideal over a Euclidean domain in a way that gives rise to canonical reductions. This retains what is perhaps the most important property of Gröbner bases over fields. A difficulty is that these can be substantially harder to compute than their field counterparts. We extend their results for computing these bases to give an algorithm that is effective in practice. In particular, we show how to use S-polynomials (rather than "critical pairs") so that the algorithm becomes quite similar to that for fields, and thus known strategies for the latter may be employed. We also show how Buchberger's important criteria for detection of unneeded S-polynomials can be extended to work over Euclidean domains. We illustrate with some examples.

1. Introduction

Since their introduction by Bruno Buchberger in the 1960s, the theory and application of Gröbner bases has been developed extensively. While the original version worked with polynomial rings defined over fields, this has been extended in different ways to other types of base ring such as Euclidean domains or principal ideal domains. Textbook expositions for this may be found in [2] and [1]. As might be expected, the less structured the base ring, the more problematic becomes the theory behind, and computation of, such bases. Moreover while the definitions for the field case are common throughout the literature, one encounters variations when working over other rings, motivated by the wish (or decreasing ability) to preserve various aspects

Received November 4, 2011; received in final form March 12, 2013.
2010 *Mathematics Subject Classification.* 11C08, 11C20, 11D04, 11R11, 13B25, 13P05, 13P10.

of the field case. One particular variant, proposed independently in [6], and [3], defines what is termed a “strong” Gröbner basis over the integers. As demonstrated in [7], this extends more generally to polynomial rings over Euclidean domains. Their motivation was to define these bases in such a way that canonical reductions to normal form are essentially unchanged from the field case. A by-product was that one also retains similarity to the field case in the algorithm for computing these. As a generalization, [12] developed similar ideas but in the setting of polynomial rings over effectively computable principal ideal domains. In this paper, we restrict attention to Euclidean domains firstly because that is the setting wherein one may preserve the notion of canonical forms, and secondly in order to have efficient computability with ring elements.

There are at least two reasons to want a strong Gröbner basis over a Euclidean domain. One is that as noted above we obtain canonical forms, and these are very useful in computations modulo polynomial ideals. The second is that reduction is now cheap; with a weak Gröbner basis one must compute greatest-common-divisors in the base ring in order to perform reduction (see [2]), whereas with a strong basis one only needs use a division algorithm (the price of course is that the basis computation itself may be more costly).

The theory behind strong bases was largely resolved by the early 1990s but details regarding efficient computation and preservation of the simplicity of Buchberger’s algorithm are scattered through the references. The intent of this paper is to gather this under one roof, so to speak, and to make explicit mention of any improvements and simplifications of which we are aware. For example, a straightforward reduction algorithm can be slow as it essentially emulates the extended GCD algorithm but with coefficient arithmetic carried over to polynomials. To remedy this [6], [7] use the extended GCD explicitly on coefficients, but make use of two types of S-polynomial and require a restriction on how reduction may be performed. Another issue is that with the notable exception of [10] the literature says relatively little about extending the Buchberger criteria for eliminating redundant S-polynomials ([1] also considered these criteria in some exercises, but in the context of what appears to be a very different algorithm for working over a PID). We give a form that is very much in the spirit of the field case in [3].

It will turn out that our basis is identical to the D-Gröbner basis discussed in Chapter 10 of [2] (they refer to the two types of polynomial as S-polynomials and G-polynomials). Our method will be computationally cheaper due to a more general reduction, availability of redundancy criteria, and fewer S-polynomials to consider. The algorithm we will discuss is implemented in the kernel of *Mathematica* [14]. General information about Gröbner bases in *Mathematica* may be found in [8].

The outline of this paper is as follows. First, we cover the basic definitions, when working in a polynomial ideal over a Euclidean domain, of term ordering, canonical rewriting, S-polynomials, and Gröbner bases. We then extend the theory presented in [3] and [7] so that it is more like the case where one works over a field. We next extend the well-known Buchberger criteria for detecting unnecessary S-polynomials in advance of performing actual (and often time consuming) reductions. We follow with several general examples. We then show some specialized applications that, among other things, connect these bases to important areas elsewhere in computational mathematics.

In the sequel, we restrict our attention mostly to the integers for clarity of exposition. Definitions and theorems in this paper extend readily to all Euclidean domains over which one can perform effective computation, provided one can canonically select elements in a way that will be made precise for the integer case below. It is then straightforward to adapt the ideas behind this case to the other common Euclidean rings for example, Gaussian integers or univariate polynomials.

2. Notation and definitions

First, we establish notation. We work in the polynomial ring of n indeterminates over the integers, $\mathbb{Z}[x_1,\ldots,x_n]$. A power product is a product of the form $\prod_{j=1}^{n}(x_j)^{e_j}$. A term, or monomial, is a power product times an integer coefficient (note that some authors define one or the other of these to be what we call a power product). We will typically denote monomials as $c_j t_j$ where c_j is an integer coefficient and t_j is a power product. One sees immediately that any polynomial in our ring can be written as a sum of terms with distinct power products; this is the usual definition of a polynomial in expanded form. Our typical usage of letters (possibly subscripted or otherwise annotated) in the sequel will be as follows: $\{a,b,c,d,e\}$ are coefficients in our ring, $\{f,g,h,p,q,r\}$ are polynomials, $\{i,j,k,m,n\}$ are integers, and $\{s,t,u,v\}$ are power products.

As in the field case we define well-founded orderings on the power products. Let $\{j_1,\ldots,j_n\}$ denote the (ordered) exponent vector of nonnegative integers for a given power product (that is, j_1 is the exponent of x_1, etc.). Suppose u, v, and w are any three such terms, 1 is the term with all exponents equal to zero, and sums of exponent vectors are performed element-wise and correspond to products of power products.

Definition 1. A total ordering among such exponent vectors (and hence among power products) is well founded provided:

(i) $1 < u$ for nonzero (u).
(ii) $u \le v \Longleftrightarrow u + w \le v + w$.

For example, we have the important "pure lexicographic" ordering wherein $\{j_1,\ldots,j_n\} > \{m_1,\ldots,m_n\}$ whenever $j_i = m_i$ for all $1 \le i < k \le n$ and $j_k > m_k$. For naming purposes, we will sometimes call a term ordering T. In the sequel when power products are compared it is always assumed that this is done with respect to a well-founded order.

Definition 2. We will regard our polynomials as sums of terms in descending term order. That is, if $p = c_1t_1 + \cdots + c_nt_n$ then we have $t_1 > t_2 > \cdots > t_n$; this depends, of course, on the particular choice of term order. The term c_1t_1 is denoted the "head" term. In the language of rewriting rules one says that the head term "reduces" to minus the sum of the remaining terms. For a pair of power products $v = \{k_1, \ldots, k_n\}$ and $w = \{m_1, \ldots, m_n\}$ we say that w divides v if $m_j \leq k_j$ for all $1 \leq j \leq n$. For abbreviation purposes we will write $\mathrm{HPP}[p] = t_1$, ("PP" for "power product"), $\mathrm{HCoeff}[p] = c_1$, and $\mathrm{HMonom}[p] = c_1t_1$.

We will assume the reader is familiar with the basic ideas of Gröbner bases in polynomial rings over fields. Good references for this include [1], [2], [3], [5]. Recall that one of several equivalent definitions is that one obtains a canonical form when reducing a given polynomial by such a basis. The various definitions are no longer equivalent when one works over a more general ring, and it is this particular one that gives rise to strong Gröbner bases when the base ring is a Euclidean domain. Before this can be described, we must first see what is meant by reduction, as this is altered from the field case.

First, we will impose an ordering on elements in the coefficient ring. For our later purposes, this too will be a total ordering, which we will denote by $\ll$. In particular, suppose our Euclidean norm on an element c in the ring is denoted by $|c|$. Then whenever $|c_1| < |c_2|$ we require that $c_1 \ll c_2$. For integers, we could for example use absolute values with ties broken by sign. So, following [6], [3], Section 8, we may take for our ordering $0 \ll 1 \ll --1 \ll 2 \ll --2 \ll \cdots$.

As will become clear, what we really require is a way to obtain unique minimal remainders in the division algorithm. This extra ordering suffices for that task.

Definition 3. Given a monomial $m = ct$ and a polynomial $p = \sum c_jt_j$ with t_1 the leading power product we say that p reduces m provided

(i) $t_1 \mid t$ (that is, we have $t = s_1t_1$).

(ii) Using the division algorithm to write $c = ac_1 + d$, we have $a \neq 0$ (or, equivalently, $|d| < |c|$). In this case we write $m \to m - as_1p$. More generally, we may allow any multiplier a such that the remainder satisfies $|d| < |c|$, but the quotient a from the division algorithm is the only one we use in actual practice.

Similarly if q and p are polynomials, we say p reduces q provided it reduces some monomial m of q. Note that reduction depends on term order in general. We make this explicit in a shorthand notation: if the resulting polynomial is r we write $q \xrightarrow{\{p,T\}} r$. Generally, we will be interested in head term reductions, but for purposes of obtaining canonical forms we will reduce lower terms as well. Also note that it is in reductions that minimal remainders become important: we require that the reduced polynomial be "smaller" either in head power product or coefficient. For the case of integers, there is a small subtlety that should be made explicit. Our division algorithm must work in such a way that the quotient of 2 and 3 is 1, with a remainder of -1 (because -1 is smaller than 2 in the Euclidean norm). This extends to other domains where the valuation provides a tie in terms of ordering. The method we develop largely avoids this issue by working with a GCD rather than iterated divisions.

Definition 4. Given a polynomial q and a set of polynomials F, we say that q is reducible by F if there is a polynomial p in F that reduces q. There may be many such, and one may get different reductions. The point of a Gröbner basis is that we will get a unique result once no further reductions can be applied, regardless of choices for reducing polynomials that were made along the way. If some chain of reductions from $q$leads to a polynomial r (regardless of whether it might be further reduced by F), we write $q \overset{\{F,T\}}{\longrightarrow} r$.

We now mention why this form of reduction is useful. As we will see, one can obtain a basis computation algorithm that is quite similar to that for fields. This is important if one is to write (almost) generic code that is at the same time optimized for different coefficient domains. Indeed, we want to use heuristics that are borrowed from the field case to the greatest extent possible, and the fewer departures from that case the more readily we are able to do this. This may also be carried beyond the Buchberger algorithm. Specifically, we note that there has been much work over the years to do Gröbner basis conversion. One such method in particular, the Gröbner walk [4], is readily extended to Euclidean domain base rings. Yet another reason to have this form of reduction is that it is fast; weak bases rely on slower GCD computations rather than division.

We now define two types of S-polynomial. Recall that the idea behind these in the field case is to combine head terms using the LCM of the lead power products, and then kill off the lead coefficient. In the Euclidean domain case, we can do this only if one lead coefficient divides the other, or if we will allow coefficient multipliers that are both nonunits. Moreover, we must allow for reducing rather than entirely removing head coefficients. For example, the pair $\{2x, 3y\}$ will, in contrast to the field case, give rise to the S-polynomial xy. While two flavors of S-polynomial marks a departure from the field case, we will see later how these may be used in an algorithm that is virtually identical to Buchberger's.

Definition 5 (S-polynomials). We are given polynomials $p_j = c_j t_j + r_j$ where $t_j = \mathrm{HPP}[p_j]$ for $j \in \{1,2\}$. Without loss of generality, we may assume $|c_1| \leq |c_2|$. Let $\{c, \{a_1, a_2\}\} = \mathrm{ExtendedGCD}[c_1, c_2]$ (that is, c is the GCD with $c = a_1c_1 + a_2c_2$). Let $t = \text{PolynomialLCM}[t_1, t_2]$ with cofactors $s_1$and s_2 so that $t = s_1t_1 = s_2t_2$. Finally, take $d =$ LCM $[c_1, c_2]$ with cofactors $b_1$and b_2 so that $d = b_1c_1 = b_2c_2$. With this, we define two types of S-polynomial:

$$\text{SPoly}_1[p_1, p_2] = a_1s_1p_1 + a_2s_2p_2 \, \text{SPoly}_2[p_1, p_2] = b_1s_1p_1 - b_2s_2p_2.$$

Note that the head term of $\text{SPoly}_1[p_1, p_2]$ had coefficient c and power product t, and in $\text{SPoly}_2[p_1, p_2]$ we have killed off that power product. Also note that when c_1 divides c_2 then $\text{SPoly}_1[p_1, p_2]$ is simply a power product multiple of p_1 (because $a_1 = 1$ and $a_2 = 0$). In this case it will obviously reduce to zero, and only $\text{SPoly}_2[p_1, p_2]$ will be of interest. Finally note that due to choices of cofactor, $\text{SPoly}_1$is not uniquely defined; this will not matter for our purposes and we merely require that an extended gcd algorithm exist. Anticipating later results, we now define, for each pair, a unique S-polynomial.

Definition 6. Again given polynomials $p_j = c_jt_j + r_j$ with $t_j = \text{HPP}[p_j]$ for $j \in \{1, 2\}$ and $|c_1| \leq |c_2|$. If c_1 divides c_2 then $\text{SPoly}[p_1, p_2] = \text{SPoly}_2[p_1, p_2]$, otherwise $\text{SPoly}[p_1, p_2] = \text{SPoly}_1[p_1, p_2]$. We remark that this is in essence "definition CP3" in [6]. It is also the efficient generalization of the definition from [3]. In that case one uses quotient and remainder to remove as much of the leading coefficient as possible from the lead term of the S-polynomial. When one lead coefficient divides the other it may be entirely removed and we have SPoly_2. When this does not happen, iterating the process emulates the Euclidean algorithm so after some number of steps we would obtain SPoly_1.

We are now ready to define a strong Gröbner basis.

Definition 7. A set of polynomials G in $\mathbb{Z}[x_1, \ldots, x_n]$ is called a strong Gröbner basis over (the base ring) $\mathbb{Z}$ and with respect to a given term ordering T if, given any polynomial $p \in \mathbb{Z}[x_1, \ldots, x_n]$, it has a canonical reduction by $\{G, T\}$. What this means is that no matter what polynomials from G we use at any given step in the process, when we can no longer reduce it we have a unique form. Restated, if $p \overset{\{G,T\}}{\longrightarrow} r_1$ and $p \overset{\{G,T\}}{\longrightarrow} r_2$ and neither r_1 nor r_2 can be further reduced by F, then $r_1 = r_2$.

Last, we will need a notion from the theory of Gröbner bases over principal ideal rings.

Definition 8. Given a set of polynomials $G = \{g_1, \ldots, g_n\} \subset \mathbb{Z}[x_1, \ldots, x_n]$ and a polynomial f with $f = \sum h_jg_j$. We call this a strong standard representation of f with respect to G provided $\text{HMonom}[f] = \text{HMonom}[h_jg_j]$ for some j and $\text{HPP}[h_kg_k] < \text{HPP}[f]$ for all $k \neq j$ (obviously this is with respect to some given term order).

We see that in a strong standard representation one kills off the head term with exactly one summand. There is also a notion of a weak standard representation, wherein we allow multiple terms with the same head power product, that is useful in construction of what are called "weak" Gröbner bases. These in turn may be used to construct strong Gröbner bases as in [1], [10]. We do not pursue that approach here. Instead we will work directly with strong standard representations. These in fact give rise to strong Gröbner bases over principal ideal rings. The characterization in that case is that all elements of the ideal have a strong standard representation; we lose canonical forms of arbitrary polynomials. Is is easy to see that existence of such representations is equivalent to one of the common characterizing features from the field case: G is a strong Gröbner basis for the ideal I provided that for any $f \in I$ there is some $g \in G$ with $\mathrm{HMonom}[g] \mid \mathrm{HMonom}[f]$ (we require now that both lead coefficient and power product of f be divisible by those of g).

3. Main results

We want to establish a type of Buchberger result connecting Gröbner bases to reduction of S-polynomials. We will do this in steps.

THEOREM 9. *Given a set of polynomials $G = \{g_1, \ldots, g_n\}$ in $A = \mathbb{Z}[x_1, \ldots, x_i]$ and a term order T. Let I be the ideal generated by G. Then following are equivalent.*

(i) *Every $g \in G$ has a strong standard representation.*

(ii) *Every $f \in A$ has a canonical reduction by $\{G, T\}$ (in other words, G is a Gröbner basis with respect to order T).*

Proof. (i)$\Longrightarrow$(ii) is similar to Lemma 10.22 and Theorem 10.23 in [2]. Suppose we have $f \overset{\{G,T\}}{\longrightarrow} h_1$ and $f \overset{\{G,T\}}{\longrightarrow} h_2$ with h_1 and h_2 both fully reduced. We need to show that $h_1 = h_2$. Since $h_1 - h_2 \in I$ it has a strong standard representation. Let $\mathrm{HMonom}[h_1 - h_2] = ct$ and $h_1 - h_2 = \sum q_j g_j$ be a strong standard representation with $\mathrm{HPP}[q_k g_k] = ct$. Let c_1 respectively, c_2 be the coefficient of t in h_1 respectively, h_2. First, suppose $c_1 = 0$. Then $\mathrm{HMonom}[h_2] = ct$ and hence h_2 is not fully reduced, contradicting our assumption. Thus $c_1 \neq 0$ and similarly we see that $c_2 \neq 0$. Hence $(c_1 - c_2)t$ reduces but neither $c_1 t$ nor $c_2 t$ reduces by G. Thus $b_k = \mathrm{HCoeff}[g_k]$ divides $(c_1 - c_2)$. Moreover, $\mathrm{Quotient}[c_1, b_k] = \mathrm{Quotient}[c_2, b_k] = 0$ for otherwise at least one of h_1 and h_2 could not be fully reduced. Thus, c_1 and c_2 are in the same residue class modulo b_k and so they are equal. This shows that the head term of $h_1 - h_2$ is zero, in other words $h_1 = h_2$ as desired.

(ii)$\Longrightarrow$(i) is similar in style to 10.8 of [2]. Suppose $f \in I, f = ct + r$ where $t = \mathrm{HPP}[f]$. By assumption of canonical reduction we have $f \overset{\{G,T\}}{\longrightarrow} 0$. Thus, we may write $f = \sum h_j g_j$ where $\mathrm{Max}_j[\mathrm{HPP}[h_j g_j]] = t$ (we remark that this is already a weak standard representation of f). Let $J = \{j : \mathrm{HPP}[h_j g_j] = t\}$. Assume for a contradiction that $m = \#J > 1$, that t is minimal among all power products (with respect to T) for which this happens, and that $|c|$ is minimal among coefficients for which there is no strong standard representation involving this head power product t. These assumptions are tenable because we work with well ordered monomials over a totally ordered Euclidean domain.

For notational convenience, assume without loss of generality that $J = \{1,\ldots,m\}$. Now let $\{\tilde{c},\{s_1,\ldots,s_m\}\} =$ ExtendedGCD[HCoeff[g_1], ..., HCoeff[g_m]] and $u_j = t/\operatorname{HPP}[g_m]$ for $1 \le j \le m$.

We next define $g = u_1 s_1 g_1 + \cdots + u_m s_m g_m$. Then by construction HMonom[g] $= \tilde{c}t$. If $|\tilde{c}| = |c|$ then $m = 1$ because we use Euclidean reduction that forces $|\operatorname{HCoeff}[h_j g_j]| \le |c|$ for $1 \le j \le m$, yet by construction as a GCD we have $|\tilde{c}| \le |\operatorname{HCoeff}[h_j g_j]|$ for $1 \le j \le m$. Thus, $|\tilde{c}| < |c|$.

By minimality of $|c|$ there is a strong standard representation $g = \sum q_j g_j$ with $\operatorname{HPP}[q_k g_k] = t$ and $\operatorname{HPP}[q_j g_j] < t$ for all $j \ne k$. As $c = \operatorname{HCoeff}[h_1 g_1] + \cdots + \operatorname{HCoeff}[h_m g_m]$ and $\tilde{c} = \operatorname{GCD}[\operatorname{HCoeff}[g_1],\ldots,\operatorname{HCoeff}[g_m]]$ we see that $\tilde{c} \mid c$, so we have $c = d\tilde{c}$ for some d. Finally let $\tilde{f} = f - dg$. Then $\operatorname{HPP}[\tilde{f}] < t$ and hence $\tilde{f}$ has a strong standard representation by our minimality hypothesis which we write as $\tilde{f} = \sum p_j g_j$. But then $d\sum q_j g_j + \sum p_j g_j$ is seen to be a strong standard representation of f. □

THEOREM 10. *Given a set of polynomials G in $\mathbb{Z}[x_1,\ldots,x_n]$ and a term order T, the following are equivalent.*

(i) *G is a Gröbner basis with respect to term order T.*

(ii) *For every pair of polynomials $\{p_1,p_2\} \subset G$ we have* $\operatorname{SPoly}_1[p_1,p_2] \overset{\{G,T\}}{\longrightarrow} 0$ *and* $\operatorname{SPoly}_2[p_1,p_2] \overset{\{G,T\}}{\longrightarrow} 0$.

(iii) *For every pair of polynomials $\{p_1,p_2\} \subset G$ we have* $\operatorname{SPoly}[p_1,p_2] \overset{\{G,T\}}{\longrightarrow} 0$.

We use both types of S-polynomial in the second equivalent statement because it is a bit easier to show that this yields a Gröbner basis. We then show that the third statement is equivalent to the second. This is useful among other reasons because one wants to retain the original Buchberger algorithm intact to the extent possible, and having one rather than two S-polynomials for a given pair clearly furthers this goal.

Proof of Theorem 10. (i) $\Longrightarrow$ (ii) is from the definition of a Gröbner basis. We now show (ii) $\Longrightarrow$ (i) (this is similar to Theorem 10.11 in [2]).

Suppose $G = \{g_1,\ldots,g_n\}$ and f is in the ideal generated by G. We may write $f = \sum h_j g_j$. Let $t = \operatorname{Max}_j[\operatorname{HPP}[h_j g_j]]$, $J = \{j : \operatorname{HPP}[h_j g_j] == t\}$, and $\tilde{t} = \operatorname{HPP}[f]$. We may assume t is minimal among such representations. Let $m = \#J$. If $m = 1$ and $t = \tilde{t}$, then we have a strong standard representation, so we assume otherwise. If $t = \tilde{t}$ then obviously $m > 1$. On the other hand, if $t > \tilde{t}$ then we require at least two terms in the representation to have power product of t in order to kill off that term. Hence, $m > 1$. Reordering if necessary, without loss of generality we may assume $J = \{1,\ldots,m\}$.

We now set up some notation. Write $g_j = c_j t_j + r_j$ and $h_j = b_j s_j + q_j$ where $t_j = \mathrm{HPP}[g_j]$ and $s_j = \mathrm{HPP}[h_j]$. Note that $s_j t_j = t$ for $1 \le j \le m$. Let $t_{1,2} = \mathrm{PolynomialLCM}[t_1, t_2]$, $v = t/t_{1,2}$, $u_1 = t_{1,2}/t_1$, and $u_2 = t_{1,2}/t_2$. From this we see at once that $s_1 = u_1 v$ and $s_2 = u_2 v$. We will assume for a contradiction that $|b_1 c_1| + \cdots + |b_m c_m|$ is minimal among all representations of f that have a largest power product of t. Again, such a representation must exist for well ordered monomials over a totally ordered Euclidean domain.

Let $\{c, \{d_1, d_2\}\} = \mathrm{ExtendedGCD}[c_1, c_2]$, $e_1 = \mathrm{LCM}[c_1, c_2]/c_1$, $e_2 = -\mathrm{LCM}[c_1, c_2]/c_2$. So e_1 and e_2 are minimal in norm such that $e_1 c_1 + e_2 c_2 = 0$.

In terms of these definitions, we have

$$\mathrm{Spoly}_1[g_1, g_2] = d_1 u_1 g_1 + d_2 u_2 g_2\, \mathrm{Spoly}_2[g_1, g_2] = e_1 u_1 g_1 + e_2 u_2 g_2.$$

Now $b_1 c_1 + b_2 c_2 = dc$ for some d; moreover there exists e such that $b_1 = dd_1 + ee_1$ and $b_2 = dd_2 + ee_2$. Since $b_1 \ne 0$ and $b_2 \ne 0$ by construction, and $c = \mathrm{GCD}[c_1, c_2]$, it follows that $|b_1 c_1| + |b_2 c_2| > |dc|$.

We now have

$$\begin{aligned} h_1 g_1 + h_2 g_2 &= (dd_1 + ee_1) u_1 v g_1 + q_1 g_1 + (dd_2 + ee_2) u_2 v g_2 + q_2 g_2 \\ &= dv\, \mathrm{SPoly}_1[g_1, g_2] + ev\, \mathrm{SPoly}_2[g_1, g_2] + (q_1 g_1 + q_2 g_2). \end{aligned}$$

By hypothesis, the S-polynomials reduce to zero. Now $v\, \mathrm{HPP}[\mathrm{SPoly}_2[g_1, g_2]] < t$, $\mathrm{HPP}[q_1 g_1] < t$, and $\mathrm{HPP}[q_2 g_2] < t$. Moreover, $\mathrm{HCoeff}[dv\, \mathrm{SPoly}_1[g_1, g_2]] = dc$. We thus have a representation of $h_1 g_1 + h_2 g_2$ as a sum $\sum p_k g_k$ whereby, letting $K = \{k :\ \mathrm{HPP}[p_k g_k] = t\}$, we obtain

$$\sum_{k \in K} \left|\mathrm{HCoeff}[p_k g_k]\right| = |dc| < |b_1 c_1| + |b_2 c_2|.$$

But then we may use this representation to replace $h_1 g_1 + h_2 g_2$ in the representation of f, and this contradicts minimality of $|b_1 c_1| + \cdots + |b_m c_m|$. Thus (ii) $\Longrightarrow$ (i).

Since (ii) is stronger than (iii) it is clear that (ii) $\Longrightarrow$ (iii). We show (iii) $\Longrightarrow$ (ii).

Let $p_j = c_j t_j + r_j$ with $\mathrm{HPP}[p_j] = t_j$ for $j \in \{1, 2\}$. Assume without loss of generality that $|c_1| \le |c_2|$. If $c_1 \mid c_2$ then $\mathrm{SPoly}_1[p_1, p_2]$ is trivially a product of $p_2$and hence known to reduce, and thus we need only use $\mathrm{SPoly}_2[p_1, p_2]$. So we may suppose that $c_1 \nmid c_2$. Let $\{c, \{a_1, a_2\}\} = \mathrm{ExtendedGCD}[c_1, c_2]$ with $c_1 = d_1 c$ and $c_2 = d_2 c$. Note that $a_1 d_1 + a_2 d_2 = 1$and in particular a_1 and a_2 are relatively prime. Let $t = \mathrm{PolynomialLCM}[t_1, t_2]$ with $s_1 = t/t_1$ and $s_2 = t/t_2$. Then

$$\begin{aligned} q &= \mathrm{SPoly}_1[p_1, p_2] \\ &= a_1 c_1 s_1 t_1 + a_1 s_1 r_1 + a_2 c_2 s_2 t_2 + a_2 s_2 r_2 \\ &= ct + a_1 s_1 r_1 + a_2 s_2 r_2 \end{aligned}$$

and

$$\begin{aligned}&\mathrm{SPoly}_2[p_1,p_2]\\&\quad=(d_2c_1s_1t_1+d_2s_1r_1)-(d_1c_2s_2t_2+d_1s_2r_2)\\&\quad=d_2s_1r_1-d_1s_2r_2.\end{aligned}$$

Thus,

$$\begin{aligned}h_1&=\mathrm{SPoly}_2[p_1,q]=c_1s_1t_1+s_1r_1-d_1(ct+a_1s_1r_1+a_1s_2r_2)\\&=(1-d-1a_1)s_1r_1-d_1a_2s_2r_2=a_2d_2s_1r_1-a_2d_1s_1r_2\\&=a_2\,\mathrm{SPoly}_2[p_1,p_2]\end{aligned}$$

and similarly

$$h_2=\mathrm{SPoly}_2[p_2,q]=a_1\,\mathrm{SPoly}_2[p_1,p_2].$$

Also by Definition 6 it is clear that $\mathrm{SPoly}_2[p_j,q]=\mathrm{SPoly}[p_j,q]$ for $j\in\{1,2\}$.

Since a_1 and a_2 are relatively prime, we obtain $\mathrm{SPoly}_1[h_1,h_2]=\mathrm{SPoly}_2[p_1,p_2]$. This shows that, provided $\mathrm{SPoly}_1[p_1,p_2]$ is not trivial, we will eventually obtain $\mathrm{SPoly}_2[p_1,p_2]$ by iterating SPoly. Hence for any pair $\{p_1,p_2\}$ we need only use $\mathrm{SPoly}[p_1,p_2]$ as given in Definition 6. □

From the theorems above, it is now not hard to see that our bases are the same as the D-bases of [2], [12]. What is very different is the mode of computation insofar as we allow Euclidean reduction of lead coefficients rather than insist on divisibility. This will tend to make them smaller sooner, and thus offers an advantage in efficiency. Note that this only applies when working over a Euclidean domain, and so the algorithm in the above references has the advantage of greater generality, albeit ours has greater flexibility in choices of reducing polynomial.

There are other ways to improve computational efficiency. It is known from long experience that the common bottleneck to the algorithm is the reduction of S-polynomials. Buchberger himself was the first to give criteria under which certain S-polynomials could be ignored (see [3] and references therein). We will show how his criteria from the field case can be adapted to Euclidean domain base rings.

THEOREM 11 (Theorem 3 (Buchberger's criterion 1)). *Suppose* $p_j=c_jt_j+r_j$ *with* $\mathrm{HPP}[p_j]=t_j$ *for* $j\in\{1,2\}$ *and* $c_1\mid c_2$. *Suppose further that the lead power products* t_1 *and* t_2 *are coprime, that is,* $\mathrm{PolynomialLCM}[t_1,t_2]=t_1t_2$. *Then* $\mathrm{SPoly}[p_1,p_2]$ *will reduce to zero and hence is superfluous.*

Note that we are using $\mathrm{SPoly}_2[p_1,p_2]$ in this case. While the divisibility requirement for lead coefficients might seem unduly strong, one will observe that the algorithm proceeds in such a way as to make coefficients small with respect to Euclidean norm. Thus in practice this requirement seems not to be terribly restrictive.

Proof of Theorem 11. $\mathrm{SPoly}[p_1,p_2]=c_1t_1p_2-c_2t_2p_1=(p_1-r_1)p_2-(p_2-r_2)p_1=r_2p_1-r_1p_2$. As t_1 divides the head term of r_2p_1 while t_2 does not, and t_2 divides the head term of r_1p_2 while t_1 does not, these do not collapse further. But clearly $r_2p_1\xrightarrow{p_1}0$ and $r_1p_2\xrightarrow{p_2}0$, so $\mathrm{SPoly}[p_1,p_2]\longrightarrow 0$. □

THEOREM 12 (Buchberger's criterion 2). *Given $p_j=c_jt_j+r_j$ with $\mathrm{HPP}[p_j]=t_j$ for $j\in\{1,2,3\}$ where $\mathrm{PolynomialLCM}[t_1,t_2]$ is divisible by t_3. Suppose $\mathrm{SPoly}[p_1,p_3]$ and $\mathrm{SPoly}[p_2,p_3]$ have strong standard representations (thus far these are the conditions for criterion 2 to be in effect in the field case). If either $c_1\mid c_3\mid c_2$ or $c_3\mid c_1\mid c_2$ then $\mathrm{SPoly}[p_1,p_2]$ will have a strong standard representation and hence is superfluous.*

Note that again we are working with $\mathrm{SPoly}_2[p_1,p_2]$. Obviously the roles of p_1 and p_2 can be interchanged. Moreover, while the divisibility conditions again appear to be restrictive, in general one obtains a lot of polynomials with a unit as lead coefficient, so these conditions are not uncommon relative to their field case counterpart.

Proof of Theorem 12. Let $t=\mathrm{PolynomialLCM}[t_1,t_2]$. Assume inductively that if f and g have strong standard representations, and $\mathrm{HPP}[f]<t$, $\mathrm{HPP}[g]<t$, then so does $f+g$.

Define power product multipliers $u_{j,k}=\mathrm{PolynomialLCM}[t_j,t_k]/t_j$. Then

$$\mathrm{SPoly}[p_1,p_2]=\frac{c_2}{c_1}u_{1,2}p_1-u_{2,1}p_2.$$

First, we assume $c_3\mid c_1\mid c_2$. Then $\mathrm{SPoly}[p_1,p_3]=u_{1,3}p_1-\frac{c_1}{c_3}u_{3,1}p_3$ and

$$\mathrm{SPoly}[p_2,p_3]=u_{2,3}p_1-\frac{c_2}{c_3}u_{3,2}p_3.$$

Since $t_3\mid\mathrm{PolynomialLCM}[t_1,t_2]$ we know that $u_{1,3}\mid u_{1,2}$ and similarly $u_{2,3}\mid u_{2,1}$. We thus may write

$$\begin{aligned}&\frac{c_2}{c_1}\frac{u_{1,2}}{u_{1,3}}\,\mathrm{SPoly}[p_1,p_3]-\frac{u_{2,1}}{u_{2,3}}\,\mathrm{SPoly}[p_2,p_3]\\&\quad=\frac{c_2}{c_1}u_{1,2}p_1-\frac{c_2}{c_3}\frac{u_{1,2}}{u_{1,3}}u_{3,1}p_3-\left(u_{1,2}p_1-\frac{c_2}{c_3}\frac{u_{2,1}}{u_{2,3}}u_{3,2}p_3\right)\\&\quad=\mathrm{SPoly}[p_1,p_2]-\frac{c_2}{c_3}p_3\left(\frac{u_{1,2}}{u_{1,3}}u_{3,1}-\frac{u_{2,1}}{u_{2,3}}u_{3,2}\right).\end{aligned}$$

Now $u_{1,2}t_1=u_{2,1}t_2$, $u_{1,3}t_1=u_{3,1}t_3$, and $u_{2,3}t_2=u_{3,2}t_3$. This implies $\frac{u_{1,2}}{u_{1,3}}u_{3,1}=\frac{u_{2,1}t_2}{u_{3,1}t_3}u_{3,1}=\frac{u_{2,1}t_2}{t_3}$ and similarly $\frac{u_{2,1}}{u_{2,3}}u_{3,2}=\frac{u_{2,1}t_2}{t_3}$. Hence, the parenthesized term vanishes, and so

$$\mathrm{SPoly}[p_1,p_2]=\frac{c_2}{c_1}\frac{u_{1,2}}{u_{1,3}}\,\mathrm{SPoly}[p_1,p_3]-\frac{u_{2,1}}{u_{2,3}}\,\mathrm{SPoly}[p_2,p_3].$$

Now use the hypothesis that each summand has a strong standard representation with head power product smaller than t. Then so does the sum.

The case where $c_1 \mid c_3 \mid c_2$ is similar. For the first step, one instead shows

$$\mathrm{SPoly}[p_1,p_2] = \frac{c_2}{c_3}\frac{u_{1,2}}{u_{1,3}}\,\mathrm{SPoly}[p_1,p_3] - \frac{u_{2,1}}{u_{3,2}}\,\mathrm{SPoly}[p_2,p_3].$$

□

A more general treatment of this criterion may be found in [10], based on generating sets of homogeneous syzygy modules. We use the version of Theorem 12 version because it is simple to code; as it is in essence the usual Buchberger criterion 2 one can adapt "standard" code for the field case with only minor modification (as indeed is done in the *Mathematica* implementation).

One will note that the criteria above pertain to the second type of S-polynomial, and naturally it would be nice to have a criterion for eliminating as redundant an S-polynomial of the first type. There is such a criterion implicit in Theorem 10.11 of [2].

THEOREM 13. *Given* $p_j = c_j t_j + r_j$ *with* $\mathrm{HPP}[p_j] = t_j$ *for* $j \in \{1,2,3\}$ *with* $t_{i,j} = \mathrm{PolynomialLCM}[t_i,t_j]$. *Suppose* $t_3 \mid t_{1,2}$. *Let* $\{c,\{a_1,a_2\}\} = \mathrm{ExtendedGCD}[c_1,c_2]$ *and further suppose* $c_3 \mid c$. *In other words, the head monomial of* p_3 *divides the head monomial of* $\mathrm{SPoly}[p_1,p_2]$ (*the latter is top-D-reducible, in the terminology of* [2]). *Then* $\mathrm{SPoly}[p_1,p_2]$ *is redundant.*

Proof. Let $u_{i,j} = t_{i,j}/t_i$ for $j \in \{1,2,3\}$. Let $c/c_3 = d$ and $t_{1,2}/t_3 = v$. Then

$$\mathrm{SPoly}[p_1,p_2] = a_1u_{1,2}p_1 + a_2u_{2,1}p_2 = ct_{1,2} + a_1u_{1,2}r_1 + a_2u_{2,1}r_2.$$

Also

$$\begin{aligned}\mathrm{SPoly}[p_1,p_3] &= \mathrm{SPoly}_2[p_1,p_3]\\ &= (c_1t_1+r_1)u_{1,3} - (c_1/c_3)(c_3t_3+r_3)u_{3,1}\\ &= u_{1,3}r_1 - (c_1/c_3)u_{3,1}r_3\end{aligned}$$

and similarly

$$\mathrm{SPoly}[p_2,p_3] = u_{2,3}r_2 - (c_2/c_3)u_{3,2}r_3.$$

Now

$$\frac{u_{1,2}u_{3,1}}{u_{1,3}} = \frac{(t_{1,2}/t_1)(t_{1,3}/t_3)}{t_{1,3}/t_1} = t_{1,2}/t_3 = v$$

and a similar computation shows that $\frac{u_{2,1}u_{3,2}}{u_{2,3}} = v$. Also

$$a_1\frac{c_1}{c_3} + a_2\frac{c_2}{c_3} - d = \frac{1}{c_3}(a_1c_1 + a_2c_2) - d = \frac{c}{c_3} - d = 0.$$

Hence,

$$\begin{aligned}
&\mathrm{SPoly}[p_1,p_2] - dvp_3 - a_1\frac{u_{1,2}}{u_{1,3}}\,\mathrm{SPoly}[p_1,p_3] - a_2\frac{u_{2,1}}{u_{2,3}}\,\mathrm{SPoly}[p_2,p_3]\\
&\quad = ct_{1,2} + a_1u_{1,2}r_1 + a_2u_{2,1}r_2 - (ct_{1,2} + dvr_3)\\
&\qquad - \left(a_1u_{1,2}r_1 - a_1\frac{c_1}{c_3}\,\frac{u_{1,2}u_{3,1}}{u_{1,3}}r_3\right) - \left(a_2u_{2,1}r_2 - a_2\frac{c_2}{c_3}\,\frac{u_{2,1}u_{3,2}}{u_{2,3}}r_3\right)\\
&\quad = \left(a_1\frac{c_1}{c_3}\,\frac{u_{1,2}u_{3,1}}{u_{1,3}} + a_2\frac{c_2}{c_3}\,\frac{u_{2,1}u_{3,2}}{u_{2,3}} - dv\right)r_3 = 0.
\end{aligned}$$

We have thus a strong standard representation of $\mathrm{SPoly}[p_1,p_2]$ and this suffices to show that it is redundant. □

We now have our algorithm, and it is virtually the same as the Buchberger algorithm for polynomial rings over fields. It is stated as follows. List all pairs of polynomials, marking as processed all those that the criteria warrant. Iteratively select a pair whose S-polynomial is not yet marked, reduce it, and if the result is not zero, form new pairs. Again use the criteria to mark redundant pairs. Continue this iteration until there are no more pairs to process, at which point all S-polynomials can be reduced to zero. Termination in a finite number of steps is proven, for example, as in [7], by noting that $\mathbb{Z}[x_1,\ldots,x_n]$ is Noetherian and hence an ascending chain condition applies to its ideals.

One will note that our algorithm puts a certain emphasis on SPoly_1, wherein the lead coefficient is the GCD of the leading coefficients of the critical pair. This is in contrast to algorithms in [1], [2], [10], [12]; where the emphasis is more on SPoly_2 in which, as with the field case, one entirely kills off a leading coefficient. Given the dearth of available implementations, it is an open question as to which approach is computationally more effective in general.

4. Some special cases

Before proceeding to examples, we will discuss an important special class of Euclidean domains. While one can show that the theory developed above carries over in a general way, for the specific and very important case where our base ring is the set of univariate polynomials in x over a (computable) field $\mathbb{F}$ one can do better. Suppose we are given a set of polynomials in some set of indeterminates over $\mathbb{F}[x]$. One augments the indeterminates with x, extending the term order so that every power of x is smaller than all power products containing other variables. One next computes a Gröbner basis for the input in this setting of polynomials in one more variable over $\mathbb{F}$. Theorem 4.5.12 in [1] shows that this is in fact a strong Gröbner basis for the ideal over the original base ring $\mathbb{F}[x]$ (this fact had also been mentioned in [7]). Our experience is that the benefits of computing a basis over a field outweigh the efficiencies developed for working over a Euclidean domain, hence we use this field computation tactic in *Mathematica*. We show applications to working over such polynomial rings in the examples.

The ring $\mathbb{Z}_{p^n}$ is an example of a finite-chain ring. For polynomials over such a ring, we could use the results presented in [11]. They show, among other things, that weak and strong Gröbner bases for ideals over such rings are equivalent. They also present a structure theorem for the univariate case and apply it to cyclic codes. One can instead regard the ring as a quotient of $\mathbb{Z}[x]$ and use the computational methods of this paper.

5. Examples

We first show some simple examples adapted from [1]. For purposes of assessing speed, we note that all timings were done with version 9 of *Mathematica* running on a 3.1 GHz Intel processor running under a 64 bit version of the Linux operating system.

For the first example, we wish to compute a basis for an ideal in the polynomial ring $\mathbb{Z}[\sqrt{-5}][x,y]$. Note that our base ring, $\mathbb{Z}[\sqrt{-5}]$, is not a Euclidean domain (or even a unique factorization domain). In such cases one may resort to a common tactic of adding a new variable and defining polynomial so that in effect we work over a quotient ring; in this example it will be $\mathbb{Z}[x,y,\alpha]/\{\alpha^2+5\}$. So our base ring will be the integers and we have added a variable and a polynomial relation equating that variable to $\sqrt{-5}$ (up to a conjugate, as these are indistinguishable to this method without further variables and defining polynomials). For this to work as desired, we must have the new variable ordered lexicographically lower than all others. We then remove the first polynomial from the basis, which, due to this ordering, is exactly the defining polynomial for that algebraic extension element.

Example 14. **Rest[GroebnerBasis[{$2xy-\alpha y,(1+\alpha)x^2-xy,\alpha^2+5$}, {$x,y,\alpha$}, CoefficientDomain → Integers]]**

$$25y+10y^2-5y\alpha, 15y+5y^2+y^2\alpha, -25y+xy+5y^3+12y\alpha,$$
$$6x^2+10y+5y^2-3y\alpha, x^2-25y+5y^3+x^2\alpha+12y\alpha.$$

The basis in that reference is a bit different due to different notions of coefficient handling, but the one above serves the same purposes.

As a second example, we will find a basis for the ideal intersection $\{3x-2,5y-3\}\cap\{xy-6\}$ in $\mathbb{Z}[x,y]$. This may be done as below. Note that we again use and subsequently eliminate an auxiliary variable, this time ordered lexicographically greater than the others (specifying it as the third argument tells *GroebnerBasis* it is to be eliminated).

Example 15. **GroebnerBasis[Flatten[{$w\{3x-2,5y-3\}$, $(1-w)\{xy-6\}$}], {x,y}, w, CoefficientDomain → Integers, MonomialOrder → EliminationOrder]**

$$\{18-30y-3xy+5xy^2, 12-18x-2xy+3x^2y, 6-6y-7xy+xy^2+x^2y^2\}.$$

Again, we do not obtain the identical basis due to differences in basis definition. Specifically, theirs does not have our third polynomial; this is because they find a weak Gröbner basis and that requires fewer polynomials.

To get some idea of algorithm speed, we now show a more strenuous computation.

Example 16. **polys** $= \{7x^2y^2 + 8xy^2 + 3xz - 11, 11y^2z + 4x^2y + xyz^2 + 2, 5x^2yz + x^2 + 2z^2 + 5z, 7xyz + 3xy + 5x + 4y + 7\}$;

Timing[gbdlex = GroebnerBasis[polys, $\{x, y, z\}$**, CoefficientDomain → Integers, MonomialOrder → DegreeLexicographic]]**

$$\{0.220000, \{344756404173555623362363962704362811959 26, 1089845251315182396260633050875076267021 9 + z, -635532288772540533781010561988733318423 4 + y, -147609871996376010904521540962105125937 21 + x\}\}.$$

A similar basis computed over the field of rationals is about two orders of magnitude faster using the same hardware and software (the result, as might be expected, is $\{1\}$ because we started with more polynomials than variables). So the fact that the Euclidean domain case takes almost two orders of magnitude longer is not entirely a surprise insofar as the eventual result contains much more information. If we remove the first polynomial, then the tasks are in some sense more similar and correspondingly the relative time ratio of computing over the rationals vs. the integers drops to under one order of magnitude.

An application of finding bases over the integers was pointed out to the author around 1996 by Dan Grayson (private communication), and in fact was implemented by him in *Mathematica* around 1988 (using the Gröbner basis over integers algorithm from [6]). Given a system of $n+1$ polynomials in n unknowns, find a modulus m such that the system is exactly determined modulo m, and return all solutions (which lie in $(\mathbb{Z}_m)^n$). With reference to the previous example, the above system is seen to be exactly determined in the quotient ring $\mathbb{Z}_{34475640417355562336236396270436281195926}$.

A related application is to do computations involving ideals defined over quotient rings that may contain zero divisors. As an example we will find all solutions in the ring $\mathbb{Z}_{5072012170009}$ to the system below.

Example 17. **gb = GroebnerBasis[**$\{5072012170009, -4984359602099 + x^2 - 3y^2 - 9xz, -1780431462965 + 7xy + 5y^3 + z^2, -4585397367278 + x^3 - 3y^2 + z - 12z^3\}$, $\{x, y, z\}$**, CoefficientDomain → Integers]**

$$\{5072012170009, 1174872829454 + 12173501962z - 1363165624472z^2 + 1654998137452z^3 + 928181308002z^4 - 239795324199z^5$$

$$\begin{aligned}
&-1646238538583z^{6}-982686930325z^{7}-1734356432441z^{8}\\
&-1928316724538z^{9}+2384106829761z^{10}-2266219400230z^{11}\\
&-139245405743z^{12}+895384068341z^{13}+161928956428z^{14}\\
&+2194204640034z^{15}-1243172466690z^{16}-1196909984892z^{17}+z^{18},\\
&2247545052503+y+788535951374z+2214230166342z^{2}\\
&+955710141543z^{3}+2160238766386z^{4}-2474194692542z^{5}\\
&-1684716364278z^{6}+2157370757916z^{7}-1072725791722z^{8}\\
&+1173330106507z^{9}-1057647942280z^{10}-1511353993603z^{11}\\
&+1327624312048z^{12}-581007814126z^{13}+1772345363132z^{14}\\
&-185000519654z^{15}-1538648034589z^{16}-456160565195z^{17},\\
&-899617339822+x+2209081769554z-509675450156z^{2}\\
&+566438534091z^{3}+1828943883971z^{4}-1778487828359z^{5}\\
&-1120529181700z^{6}+1238816552216z^{7}-1898793743218z^{8}\\
&+1286010808749z^{9}+893019914153z^{10}+172896055599z^{11}\\
&+1872411543380z^{12}+1420313673322z^{13}-880454763764z^{14}\\
&-1202867057825z^{15}-1977589465047z^{16}-2210999439349z^{17}\}.
\end{aligned}$$

To obtain solutions, one would proceed exactly as if working over a field. Specifically, we first find roots of the univariate polynomial, then back substitute each solution to solve for the remaining variables. We show the first step explicitly. This involves root finding in a quotient ring of the integers. The principles behind this are well known (factor the modulus, find roots modulo each prime factor, lift to accommodate powers of primes, use the Chinese Remainder Algorithm to combine roots modulo powers of primes). The "hard" step, computationally speaking, is often the factorization of the modulus.

$\mathbf{Roots[gb[[2]] == 0}, \boldsymbol{z}, \mathbf{Modulus \to gb[[1]]]}$

$$z=99999\|z=1848935269876\|z=3102255902823.$$

This functionality is now built into *Mathematica*, in the function *Reduce*:

$\mathbf{gb = Reduce[\{5072012170009,}$
$\boldsymbol{-4984359602099+x^{2}-3y^{2}-9xz,-1780431462965+7xy+5y^{3}+z^{2},}$
$\boldsymbol{-4585397367278+x^{3}-3y^{2}+z-12z^{3}\}==0,}$
$\boldsymbol{\{x,y,z\}}\mathbf{, Modulus \to 5072012170009]}$

$$\begin{aligned}
&\{0.050000,(x=77777\&\&y=88888\&\&z=99999)\|\\
&(x=1712760123092\&\&y=3989577716979\&\&z=1848935269876)\|\\
&(x=2127801384642\&\&y=3379908964470\&\&z=3102255902823)\}.
\end{aligned}$$

Another area of application for Gröbner bases over the integers is in computations with finitely presented groups, as discussed in Chapter 10 of [13]. Among other tools one requires a module Gröbner basis. Further applications include Hensel lifting of univariate polynomials, computation of matrix Hermite and Popov normal forms when elements lie in a Euclidean domain, bivariate modular polynomial factorization, and computing small generators of ideals in quadratic number rings. We will illustrate these with numerous examples in a separate report [9].

6. Summary

We have presented an algorithm for computing a strong Gröbner basis over a Euclidean domain that is essentially identical to Buchberger's method for the case where the base ring is a field. In particular, we have retained the S-polynomial reduction approach as well as the Buchberger criteria for elimination of redundant S-polynomials. Several basic examples were presented to illustrate diverse applications of this technology for example, working over quotient rings and solving nonlinear systems over rings. We show a number of specialized applications of these bases, for example, to compute Hensel lifts in a univariate polynomial ring and to find matrix Hermite and Popov normal forms, in [9].

Acknowledgment. I thank an anonymous referee from an earlier draft for pointing out several typographical errors in an earlier draft of this paper, another for drawing my attention to the related work by Norton and Sălăgean, and Ana Sălăgean for providing a preprint of their work. I also thank Dan Grayson for discussing this topic with me many years ago.

References

[1] W. Adams and P. Loustaunau, *An introduction to Gröbner bases*, Amer. Math. Soc., Providence, RI, 1994. MR 1287608

[2] T. Becker, H. Kredel and V. Weispfenning, *Gröbner bases: A computational approach to commutative algebra*, Springer, London, 1993. MR 1213453

[3] B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Chapter 6, Reidel Publishing Company, Dodrecht, 1985, pp. 1084–2322.

[4] S. Collart, M. Kalkbrener, and D. Mall, *Converting bases with the Gröbner walk*, J. Symbolic Comput. **24** (1997), 465–469. MR 1484492

[5] D. A. Cox, J. Little and D. O'Shea, *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*, 3rd ed., Undergraduate texts in mathematics, Springer, New York, 2007. MR 2290010

[6] A. Kandri-Rody and D. Kapur, *Algorithms for computing Gröbner bases of polynomial ideals over various Euclidean rings*, EUROSAM, Springer, Berlin, 1984, pp. 195–206. MR 0779126

[7] A. Kandri-Rody and D. Kapur, *Computing a Gröbner basis of a polynomial ideal over a Euclidean domain*, J. Symbolic Comput. **6** (1988), 37–57. MR 0961369

[8] D. Lichtblau, *Gröbner bases in Mathematica 3.0*, Math. J. **6** (1996), 81–88; available at `http://library.wolfram.com/infocenter/Articles/2179/`.

[9] D. Lichtblau, *Applications of strong Gröbner bases over Euclidean domains*, Int. J. Algebra **7** (2013), 369–390.

[10] H. M. Möller, *On the construction of Gröbner bases using syzygies*, J. Symbolic Comput. **6** (1988), 345–359. MR 0988422

[11] G. H. Norton and A. Sălăgean, *Strong Gröbner bases and cyclic codes over a finite-chain ring*, Electron. Notes Discrete Math. **6** (2001), 240–250. MR 1985246

[12] L. Pan, *On the d-bases of polynomial ideals over principal ideal domains*, J. Symbolic Comput. **7** (1989), 55–69. MR 0984271

[13] C. Sims, *Computation with finitely presented groups*, Cambridge Univ. Press, Cambridge, 1994. MR 1267733

[14] I. Wolfram Research, Champaign, Illinois, *Mathematica 9*; available at `http://www.wolfram.com`, 2012.

Daniel Lichtblau, Wolfram Research, Inc., 100 Trade Centre Dr, Champaign, IL 61820, USA

E-mail address: `danl@wolfram.com`