*J. Symbolic Computation* (2004) **11**, 1–000

# Computing Zero-Dimensional Schemes [†]

J. ABBOTT*, M. KREUZER**, L. ROBBIANO*

**Department of Mathematics, University of Genova, Italy*
*** Fakultät für Mathematik, Universität Regensburg, Germany*

*(Received 27 October 2004)*

This paper is a natural continuation of Abbott *et al.* (2000) further generalizing the Buchberger-Möller algorithm to zero-dimensional schemes in both affine and projective spaces. We also introduce a new, general way of viewing the problems which can be solved by the algorithm: an approach which looks to be readily applicable in several areas. Implementation issues are also addressed, especially for computations over $\mathbb{Q}$ where a trace-lifting paradigm is employed. We give a complexity analysis of the new algorithm for fat points in affine space over $\mathbb{Q}$. Tables of timings show the new algorithm to be efficient in practice.

## 1. Introduction

Nowadays it is common knowledge that Gröbner bases and Buchberger's Algorithm are key ingredients in Computational Commutative Algebra, and are hence fundamental tools for applications in several fields both inside and outside Mathematics (see Buchberger (1985)). It is also well-known that the computation of a Gröbner basis can be time consuming due to its intrinsic complexity. Therefore many attempts have been made in recent years to find special situations in which the usual computational scheme of Buchberger's Algorithm can be improved.

For instance, in a recent paper (see Abbott *et al.* (2000)) we addressed the problem of computing the vanishing ideal of a set of reduced $K$-rational points, where $K$ is a field. In particular, we studied the case $K = \mathbb{Q}$. Our investigation was based on the Buchberger-Möller Algorithm (BM-algorithm) which improves the traditional scheme for computing intersections of ideals of points considerably (see Buchberger and Möller (1982)).

The first question we want to address now is the following. Is there a more general computational problem one of whose specializations is solved by the classical BM-algorithm? For this we let $P = K[x_1, \ldots, x_n]$ be the polynomial ring in $n$ indeterminates over a field $K$, we let $M$ be a $P$-module, and we let $\varphi : P \longrightarrow M$ be a homomorphism of $P$-modules. The task is to compute $\mathrm{Ker}(\varphi)$ efficiently.

Let us see how this general setting specializes to the case treated by the classical BM-algorithm. Let $\mathbf{p}_1, \ldots, \mathbf{p}_s \in \mathbb{A}^n_K$ and $\mathfrak{m}_1, \ldots, \mathfrak{m}_s$ be their associated maximal ideals. Let $M = \oplus_{i=1}^s P/\mathfrak{m}_i \cong K^s$, and let $\varphi : P \longrightarrow M$ be defined by $\varphi(f) = (f(p_1), \ldots, f(p_s))$.

[†] `abbott,robbiano@dima.unige.it` `martin.kreuzer@mathematik.uni-regensburg.de`

Then the problem of computing $\operatorname{Ker}(\varphi)$ is exactly the problem of computing $\bigcap_{i=1}^{s} \mathfrak{m}_i$, and indeed a good solution is the BM-algorithm.

A more general instance of the above computational problem is the computation of the vanishing ideal of a zero-dimensional scheme. Let $\mathfrak{m}_1, \ldots, \mathfrak{m}_s$ be maximal ideals in $P$, and for each $i$ let $\mathfrak{q}_i$ be an $\mathfrak{m}_i$-primary ideal. Then $I = \mathfrak{q}_1 \cap \cdots \cap \mathfrak{q}_s$ is the vanishing ideal of some zero-dimensional subscheme $\mathbb{X} \subseteq \mathbb{A}^n_K$. It is also the kernel of the canonical $P$-linear map $\pi : P \longrightarrow \oplus_{i=1}^{s} P/\mathfrak{q}_i$. In this case the methods based on Buchberger's Algorithm tend to be rather inefficient in practice and faster methods are needed.

If the ideals $\mathfrak{q}_1, \ldots, \mathfrak{q}_s$ are described by the vanishing of certain "dual functionals", a suitable adaptation of the BM-algorithm was given in Marinari *et al.* (1993). But to find those functionals from systems of generators of $\mathfrak{q}_1, \ldots, \mathfrak{q}_s$ does not seem easy, especially if $K$ has finite characteristic. A more direct approach was suggested in Lakshman (1991), but the cost of computing local normal form vectors and the problem of the growth of coefficients in the case $K = \mathbb{Q}$ were ignored. The related problem of computing minimal generators has been studied in Cioffi (1998) and Cioffi and Orecchia (2001).

To get a better grip on this situation, and in order to put it in a suitable general framework, we start in Section 2 by studying $K$-linear, surjective maps $\varphi : P \longrightarrow K^\mu$, where $\mu \geq 1$; this was partly inspired by some ideas given in Mourrain (1999), and is further extended in Robbiano (2001). We show that $\operatorname{Ker}(\varphi)$ is a zero-dimensional ideal in $P$ if and only if $\varphi$ maps a polynomial to its normal form vector with respect to a tuple of polynomials whose residue classes are a $K$-vector space basis of $P/\operatorname{Ker}(\varphi)$. The important case is when $\varphi$ is explicitly computable. For instance, if $P/\operatorname{Ker}(\varphi)$ is generated by the residue classes of the terms in the complement of some leading term ideal and if $\varphi$ is constructed using the normal form map, it is explicitly computable. But we shall also see that there are cases where $\varphi$ is explicitly computable, but not of this type. The "change of basis" between two such maps having the same kernel is achieved by a generalization of the well-known FGLM-algorithm.

Then we present our first generalization of the BM-algorithm in Section 3. It computes the vanishing ideal of a zero-dimensional scheme $\mathbb{X} \subseteq \mathbb{A}^n_K$ as above. The zero-dimensional ideals whose intersection we want to compute are represented by normal form vector maps. To compute the local normal form vectors more efficiently, we show how to use a method similar to the one introduced in Faugère *et al.* (1989): the ideals $I_k$ are represented by the image of 1 in the basis of $P/I_k \cong K^{\mu_k}$ and by the matrices representing the multiplications by $x_1, \ldots, x_n$ in this basis. Then we exploit the fact that the algorithm operates only on terms of the form $t = x_j t'$ for which the local normal forms of $t'$ are already known.

As in Abbott *et al.* (2000), we are also able to extend the method to the computation of the vanishing ideal of a zero-dimensional scheme $\mathbb{X} \subseteq \mathbb{P}^n_K$. Here we have to intersect one-dimensional homogeneous saturated ideals in $K[x_0, \ldots, x_n]$. By proceeding degree by degree, we can reduce the problem to computations involving finite dimensional vector spaces. The problem is then to find a good stopping criterion which says that the result is complete after we have reached a certain degree. For the case of zero-dimensional subschemes of $\mathbb{P}^n_K$, we give two stopping criteria and the projective version of the BM-algorithm in Section 4.

As mentioned above, if we consider the problem of computing the vanishing ideal of $\mathbb{X} \subset \mathbb{A}^n_K$ over the base field $K = \mathbb{Q}$, we run into the additional problem of coefficient growth. This problem is addressed in Section 5, where we present a version of the general BM-algorithm using an approach a bit like the method of Gröbner traces. The brunt of the computation becomes the solving of some linear systems over $\mathbb{Q}$, a well-studied problem for which efficient algorithms already exist.

One case which is particularly important for applications in Algebraic Geometry is the case of schemes containing fat points: a fat point is a point in $\mathrm{Supp}(\mathbb{X})$ whose local ideal is a power of the associated maximal ideal. We examine this situation in Section 6. We exhibit particularly efficient normal form vector maps for this case, and proceed to analyse the complexity of the algorithm of Section 5 assuming use of these maps. We find that for simple points the new algorithm matches the complexity of the less general modular algorithm given in Abbott *et al.* (2000).

Although we are not going to address them directly in this paper, we mention that for the general computational problem described above there are other cases where suitable generalizations of the BM-algorithm appear feasible. For instance, if we view $M = P' = K[y_1, \ldots, y_m]$ as a $P$-module via a $K$-algebra homomorphism $\varphi$ given by $\varphi(x_i) = f_i(y_1, \ldots, y_m)$ for $i = 1, \ldots, n$, then the general task specializes to the implicitization problem. Again it is known that the general implicitization problem is hard, but there are special cases which can be treated more directly. For instance in the case of toric ideals, an *ad hoc* approach was used in Bigatti *et al.* (1999) to tame the intrinsic difficulties of elimination theory.

All algorithms described in this paper are implemented in the system CoCoA which is available from `http://cocoa.dima.unige.it` (see Capani *et al.* (1998)). Some experimental data based on our implementations are reported in Section 7.

## 2. Theoretical Preliminaries

In this section we describe some material which forms the theoretical background of our algorithms. For an extended presentation of this material we refer to Robbiano (2001). Throughout we assume that $P = K[x_1, \dots, x_n]$ is a polynomial ring over a field $K$ and that $\mathbb{T}^n$ is the monoid of terms (power products) of $P$. At various points it will be handy to refer to a basis for $K^\mu$, so let $e_1, \dots, e_\mu$ be one with the convention that $(a_1, \dots, a_\mu) \in K^\mu$ refers to the vector $\sum_i a_i e_i$.

Example 2.1. Let $I \subset P$ be a zero-dimensional ideal, let $\mu = \dim_K(P/I)$, let $\sigma$ be a term ordering on $\mathbb{T}^n$, and let $\mathcal{O}_\sigma = (t_1, \dots, t_\mu)$ be a tuple whose components are precisely the terms in $\mathbb{T}^n$ which are not contained in the leading term ideal $\mathrm{LT}_\sigma(I)$. By the Macaulay Basis Theorem (see for instance Kreuzer and Robbiano (2000), Thm. I.5.7), the tuple $(\bar{t}_1, \dots, \bar{t}_\mu)$ of the residue classes of the elements of $\mathcal{O}_\sigma$ is a $K$-basis of $P/I$.

Let $G$ be a $\sigma$-Gröbner basis of $I$. For every polynomial $f \in P$, the Division Algorithm with respect to $G$ yields its normal form $\mathrm{NF}_{\sigma,I}(f) = \sum_{i=1}^{\mu} a_i t_i$, where $a_1, \dots, a_\mu \in K$. The canonical surjective map $\pi : P \longrightarrow P/I$ satisfies $\pi(f) = \sum_{i=1}^{\mu} a_i \bar{t}_i$. Our choice of basis for $P/I$ also fixes a canonical isomorphism $P/I \longrightarrow K^\mu$, and combining this with $\pi$ we obtain a $K$-linear, surjective map $\mathrm{NFV}_{\mathcal{O}_\sigma} : P \longrightarrow K^\mu$ sending $f \mapsto (a_1, \dots, a_\mu)$ and whose kernel is precisely the ideal $I$.

Example 2.2. Let $\{(0,0),\ (0,-1),\ (1,0),\ (1,1),\ (-1,1)\} \subset \mathbb{A}_2(\mathbb{Q})$, let $I \subset P = \mathbb{Q}[x,y]$ be the vanishing ideal of this set of points, and let $\mathcal{O} = (1,\, x,\, y,\, x^2,\, y^2)$. Since the evaluation matrix of those terms at the given points has determinant $-4$, we conclude that the tuple $(\bar{1},\, \bar{x},\, \bar{y},\, \bar{x}^2,\, \bar{y}^2)$ is a $\mathbb{Q}$-basis of $P/I$. Therefore, as before, there is a $K$-linear, surjective map $\mathrm{NFV}_{\mathcal{O}} : P \longrightarrow K^5$ which sends every polynomial $f$ to the uniquely defined tuple $(a_1, a_2, a_3, a_4, a_5)$ such that the residue class of $f$ in $P/I$ is $a_1 + a_2\bar{x} + a_3\bar{y} + a_4\bar{x}^2 + a_5\bar{y}^2$.

But the map $\mathrm{NFV}_{\mathcal{O}}$ is not induced by a map of the form $\mathrm{NF}_{\sigma,I}$, since the components of $\mathcal{O}$ do not form a set $\mathbb{T}^2 \setminus \mathrm{LT}_\sigma(I)$ for some term ordering $\sigma$. To prove this, consider the polynomial $f = x^2 + xy - x - \frac{1}{2}y^2 - \frac{1}{2}y$. It is in $I$, since it vanishes at the five points. For any term ordering $\sigma$, we have $x^2 >_\sigma x$ and $y^2 >_\sigma y$. If $x >_\sigma y$, then $x^2 >_\sigma xy >_\sigma y^2$. And if $y >_\sigma x$, then $y^2 >_\sigma xy >_\sigma x^2$. This means that there are only two possibilities for the leading term of $f$: either it is $x^2$ or $y^2$. In either case we have that $\mathrm{LT}_\sigma(f)$ is both a component of $\mathcal{O}$, and an element of $\mathrm{LT}_\sigma(I)$.

For instance, examples of this second type, which do not come from Gröbner bases, arise in the study of Design of Experiments — see Caboara and Robbiano (2001). As has already been noted by Stetter, and more recently recalled in Mourrain (1999), such non-Gröbner examples are also important for symbolic-numeric solving. These situations motivate the following definition.

Definition 2.3. Let $I$ be a zero-dimensional ideal in $P$, let $\pi : P \longrightarrow P/I$ be the canonical map, let $\mu = \dim_K(P/I)$, and let $\mathcal{O} = (t_1, \dots, t_\mu)$ be a tuple of polynomials such that $\overline{\mathcal{O}} = (\bar{t}_1, \dots, \bar{t}_\mu)$ is a basis of $P/I$ as a $K$-vector space. The vector $(a_1, \dots, a_\mu) \in K^\mu$ such that $\pi(f) = a_1\bar{t}_1 + \cdots + a_\mu\bar{t}_\mu$ is called the **normal form vector** of $f$ with respect to $\mathcal{O}$ and is denoted by $\mathrm{NFV}_{\mathcal{O}}(f)$. The corresponding map $\mathrm{NFV}_{\mathcal{O}} : P \longrightarrow K^\mu$ is called the **normal form vector map** with respect to $\mathcal{O}$. To ease notation the dependence of $\mathrm{NFV}_{\mathcal{O}}$ on the ideal $I$ is not written out.

Clearly for actual computations we shall need explicit normal form vector maps, *i.e.* ones given by concrete algorithms. The next step is to characterize zero-dimensional ideals via normal form vector maps.

Proposition 2.4. *Let $\mu \geq 1$, and let $\varphi : P \longrightarrow K^\mu$ be a $K$-linear, surjective map. The following conditions are equivalent.*

- *a) The kernel of $\varphi$ is a zero-dimensional ideal in $P$.*
- *b) The map $\varphi$ is a normal form vector map, i.e. $\varphi = \mathrm{NFV}_{\mathcal{O}}$ for some choice of $I$ and $\mathcal{O}$.*
- *c) The map $\varphi$ is the composition of a normal form vector map $\mathrm{NFV}_{\mathcal{O}_\sigma}$, where $\mathcal{O}_\sigma$ is the complement of some leading term ideal, with a linear base change $K^\mu \longrightarrow K^\mu$.*

Proof. First we show that $a) \Rightarrow b)$. For every $i = 1, \ldots, \mu$ we select a polynomial $t_i \in P$ such that $\varphi(t_i) = e_i \in K^\mu$. Hence we define a $K$-linear map $\psi : K^\mu \longrightarrow P$ such that $\varphi \circ \psi$ is the identity on $K^\mu$. Let us write $I$ for the zero-dimensional ideal $\mathrm{Ker}(\varphi)$. Since $P = I \oplus \psi(K^\mu)$, every polynomial $f$ can be uniquely represented as $f = g + \sum_{i=1}^{\mu} a_i t_i$, where $g \in I$ and $a_1, \ldots, a_\mu \in K$. Thus we get $\varphi(f) = \varphi(g) + \sum_{i=1}^{\mu} a_i \varphi(t_i) = \sum_{i=1}^{\mu} a_i e_i$, and therefore $\varphi = \mathrm{NFV}_{\mathcal{O}}$ for $\mathcal{O} = (t_1, \ldots, t_\mu)$.

Now we prove that $b) \Rightarrow c)$. Assume there exist an ideal $I$ and a tuple $\mathcal{O} = (t_1, \ldots, t_\mu)$ of polynomials for which $\overline{\mathcal{O}} = (\bar{t}_1, \ldots, \bar{t}_\mu)$ is a basis of $P/I$ as a $K$-vector space. Let $\sigma$ be a term ordering on $\mathbb{T}^n$, and let $\mathcal{O}_\sigma = (\tau_1, \ldots, \tau_\mu)$ be a tuple whose components are the terms in $\mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$. As already pointed out in Example 2.1, the tuple $\overline{\mathcal{O}}_\sigma$ of the residue classes $(\overline{\tau}_1, \ldots, \overline{\tau}_\mu)$ of the elements of $\mathcal{O}_\sigma$ is a $K$-basis of $P/I$. Therefore $\overline{\mathcal{O}} = \overline{\mathcal{O}}_\sigma \cdot M$ for some invertible matrix $M$. Hence by their definitions $\mathrm{NFV}_{\mathcal{O}}(f) = M \cdot \mathrm{NFV}_{\mathcal{O}_\sigma}(f)$ for every $f \in P$; that is $\varphi$ is the composition of $\mathrm{NFV}_{\mathcal{O}_\sigma}$ with the change of basis given by the matrix $M$.

Now we prove that $c) \Rightarrow a)$. Assume $c)$. Clearly the kernel of the composition is the kernel of $\mathrm{NFV}_{\mathcal{O}_\sigma}$. This latter is precisely the ideal implicit in the normal form vector map, which is a zero-dimensional ideal. □

This proposition allows us to represent a zero-dimensional ideal in our later algorithms by a normal form vector map. Given two normal form vector maps corresponding to the same zero-dimensional ideal, we can change from one to the other as follows.

Remark 2.5. Let $I \subset P$ be a zero-dimensional ideal, and let $\mathcal{O} = (t_1, \ldots, t_\mu)$ be a tuple of polynomials whose residue classes form a basis of $P/I$ as a $K$-vector space.

- a) Given an explicit $\mathrm{NFV}_{\mathcal{O}}$, if we have a tuple $\mathcal{O}_\sigma = \mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$ for some term ordering $\sigma$, we can calculate $\mathrm{NFV}_{\mathcal{O}}(t)$ for all components $t$ of $\mathcal{O}_\sigma$, and we obtain the matrix $M$ in the proof of Proposition 2.4. Thus we can compute $\mathrm{NFV}_{\mathcal{O}_\sigma}(f)$ for all $f \in P$ without having to find a $\sigma$-Gröbner basis of $I$.
- b) Given an explicit $\mathrm{NFV}_{\mathcal{O}}$ and a term ordering $\sigma$, but no corresponding tuple $\mathcal{O}_\sigma$, we can still find such a tuple without resorting to Buchberger's Algorithm. Namely, we can apply the Algorithm GBM of Theorem 3.1 in the case $s = 1$. We get a Change of Base Algorithm which generalizes the standard FGLM-algorithm (see Faugère *et al.* (1989), Section 3), because it does not require a Gröbner basis as input.

We conclude with two results we shall need in the next section where we show how to represent a zero-dimensional ideal using commuting matrices. These results are inspired by Mourrain (1999).

Proposition 2.6. *Let $\mu \geq 1$, let $\varphi : P \longrightarrow K^\mu$ be a $K$-linear, surjective map whose kernel is a zero-dimensional ideal $I$ in $P$, and let $\mathbf{w} = \varphi(1)$. Then there exist uniquely defined, pairwise commuting matrices $M_1, \ldots, M_n$ in $\operatorname{Mat}_\mu(K)$ such that*

*a) $\varphi(x_i f) = M_i \cdot \varphi(f)$ for all $f \in P$ and all $i = 1, \ldots, n$.*
*b) $\varphi(f) = f(M_1, \ldots, M_n) \cdot \mathbf{w}$ for all $f \in P$.*

Proof. We explain how to construct the matrices $M_i$; the rest of the proof is simple algebra. Let $g_1, \ldots, g_\mu$ be polynomials such that $\varphi(g_k) = e_k \in K^\mu$ for $k = 1, \ldots, \mu$. We define $M_i$ to be the matrix whose columns are the vectors $\varphi(x_i g_1), \ldots, \varphi(x_i g_\mu)$. □

Definition 2.7. The pairwise commuting matrices $M_1, \ldots, M_n$ described in the above proposition are called the **multiplication matrices** of $\varphi$.

Here is a sort of converse to Proposition 2.6

Proposition 2.8. *Let $\mu \geq 1$, let $\mathbf{w} \in K^\mu$ be a non-zero vector, and let $M_1, \ldots, M_n$ be pairwise commuting matrices in $\operatorname{Mat}_\mu(K)$.*

*a) There exists a unique map $\varphi : P \longrightarrow K^\mu$ with the following properties:*

*$a_1$) $\varphi(1) = \mathbf{w}$,*
*$a_2$) $\varphi$ is $K$-linear,*
*$a_3$) $\varphi(x_i f) = M_i \varphi(f)$ for all $f \in P$ and each $i = 1, \ldots, n$.*

*b) We have $\varphi(f) = f(M_1, \ldots, M_n) \cdot \mathbf{w}$ for all $f \in P$.*
*c) The kernel of $\varphi$ is a zero-dimensional ideal.*

Proof. The first part of the proof is straightforward algebra with the observation that the commutativity of the matrices $M_i$ is necessary for $\varphi$ to be well-defined.

To show that $\operatorname{Ker}(\varphi)$ is an zero-dimensional ideal, let $f \in \operatorname{Ker}(\varphi)$. For any $x_i$ we have $\varphi(x_i f) = M_i \varphi(f) = 0$. By induction and the $K$-linearity of $\varphi$, we deduce that $gf \in \operatorname{Ker}(\varphi)$ for any $g \in P$. Now it is clear that $\operatorname{Ker}(\varphi)$ is an ideal; it is zero-dimensional because $P/I$ is a finite-dimensional $K$-vector space. □

## 3. The Generalized BM-Algorithm

In Section 2 we established a correspondence between zero-dimensional ideals and normal form vector maps. In particular, we showed how knowledge of the Gröbner basis of such an ideal enabled us to determine a corresponding normal form vector map $\mathrm{NFV}_{\mathcal{O}}$. In this section we present a generalization of the BM-algorithm which determines directly a Gröbner basis for the intersection of a finite number of zero-dimensional ideals where each ideal is represented by a normal form vector map.

In the following, let $K$ be a field, let $P = K[x_1, \ldots, x_n]$, let $s \geq 1$, and for $i = 1, \ldots, s$ let $\mathrm{NFV}_{\mathcal{O}_i} : P \longrightarrow K^{\mu_i}$ be a normal form vector map representing a zero-dimensional ideal $I_i \subseteq P$. Recall that this means $\mu_i = \dim_K(P/I_i)$ and $I_i = \mathrm{Ker}(\mathrm{NFV}_{\mathcal{O}_i})$, and that $\mathcal{O}_i$ is a set of polynomials whose residue classes form a $K$-basis of $P/I_i$.

Our goal is to compute a Gröbner basis of the ideal $I = \bigcap_{i=1}^{s} I_i$. We present here the generalized BM-Algorithm: in outline, we consider all power products in increasing order (according to the term-ordering), for each power product we seek a linear dependency of its normal form vector on those of smaller power products, if there is a dependency then we get a new Gröbner basis element, otherwise we place the power product in the quotient basis for use in finding future linear dependencies.

Theorem 3.1. **(Algorithm GBM)**
*Let $\sigma$ be a term ordering on $\mathbb{T}^n$. Consider the following instructions.*

**GBM1** *Start with empty lists $G = [\,]$, $\mathcal{O} = [\,]$, a list $L = [1]$, and a matrix $M = (m_{ij})$ over $K$ with $\mu = \mu_1 + \cdots + \mu_s$ columns and initially zero rows.*

**GBM2** *If $L$ is empty, return the pair $[G, \mathcal{O}]$ and stop. Otherwise choose the power product $t = \min_\sigma(L)$ and remove it from $L$.*

**GBM3** *Compute the vector $\mathbf{v} = \mathrm{NFV}_{\mathcal{O}_1}(t) \oplus \cdots \oplus \mathrm{NFV}_{\mathcal{O}_s}(t) \in K^{\mu}$.*

**GBM4** *Reduce $\mathbf{v}$ against the rows of $M$ to obtain*

$$\mathbf{v}^* = \mathbf{v} - \sum_i a_i \mathbf{m}_i \qquad \text{with } a_i \in K$$

*where $\mathbf{m}_i = (m_{i1}, \ldots, m_{i\mu})$ is the $i^{\text{th}}$ row of the matrix $M$.*

**GBM5** *If $\mathbf{v}^* = \mathbf{0}$, then append the polynomial $t - \sum_i a_i t_i$ to the list $G$, where $t_i$ is the $i^{\text{th}}$ power product in the list $\mathcal{O}$. Continue with step* GBM2.

**GBM6** *Otherwise $\mathbf{v}^* \neq (0, \ldots, 0)$, so append the vector $\mathbf{v}$ as a new row to $M$. Append the corresponding term $t$ to the list $\mathcal{O}$. Add to $L$ those elements of $\{x_1 t, \ldots, x_n t\}$ which are neither multiples of an element of $L$ nor of $\{\mathrm{LT}_\sigma(g) \mid g \in G\}$. Continue with step* GBM2.

*This is an algorithm which computes a pair $(G, \mathcal{O})$ such that $G$ is a list of polynomials in $P$ forming the reduced $\sigma$-Gröbner basis of $I = \bigcap_{i=1}^{s} I_i$ and $\mathcal{O}$ is a list whose components are precisely the elements of $\mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$.*

Proof. In Algorithm GBM the reduced Gröbner basis is accumulated into the variable $G$; in the main loop $G$ contains those elements whose leading term is smaller than $t$. A quotient basis of power products in accumulated into $(O)$, it contains only power products known not to be reducible by any Gröbner basis element; at the end of the algorithm is contains all such power products. The list $L$ helps identify quickly the next power product to consider in step GBM2.

First we exhibit termination. In each iteration either step GBM5 is performed or step GBM6. By its construction the matrix $M$ always has linearly independent rows, and hence step GBM6, which adjoins a row to $M$, can be performed only finitely many times (at most $\mu$ times). By Dickson's lemma step GBM5 can be performed only finitely many times; the noetherianity of $P$ implies this too. Thus the algorithm performs only finitely many iterations, and each iteration clearly involves only a finite amount of computation.

To exhibit correctness we use induction on the iterations of the algorithm: we shall show that if the values of $G$ and $\mathcal{O}$ are correct at the start of an iteration then they are still correct at the end of the iteration. Let $B$ denote the reduced Gröbner basis for the intersection, and $Q = \mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$ be the set of all power products not divisible by the leading term of some element of $B$.

If $L$ is not empty then it contains a minimal element $t$. So at the start of the iteration we have that the list $G$ contains all elements of $B$ whose leading term is $\sigma$-smaller than $t$, and the list $\mathcal{O}$ contains all elements of $Q$ which are $\sigma$-smaller than $t$.

We must show that in each iteration the power product $t$ is either added to $\mathcal{O}$ or gives a new reduced Gröbner basis element as appropriate. The case $t = 1$ is trivial. Now consider the case $t > 1$. In step GBM5, if the vector $\mathbf{v}^*$ is zero then the polynomial $t - \sum_i a_i t_i$ is an element of $I_i$ for each $i$, and thus also of their intersection. Consequently $B$ contains an element whose leading term divides $t$, but by definition of $L$ no element of $G$ does this. Hence $B$ must contain an element whose leading term is exactly $t$; and this element is added to $G$ in step GBM5.

Should we reach step GBM6 then $t$ is an element of $Q$ because any element of $B$ whose leading term divides $t$ would have to have leading term exactly $t$ (by definition of $L$), and so such element would necessarily be of the form $t - \sum b_i t_i$ which would correspond to a linear relationship between $\mathbf{v}$ and the rows of the matrix $M$, and yet the reduction to a non-zero vector in step GBM4 proved that no such relationship exists.

We affirm that the list $L$ is updated in such a way that its $\sigma$-smallest element is always the $\sigma$-smallest power product greater than $t$ and not divisible by the leading term of some element of $G$. □

Now we know how to compute the intersection of the kernels of finitely many explicit $K$-algebra homomorphisms. For instance, the theorem applies when we have Gröbner bases of two zero-dimensional ideals $I_1$ and $I_2$ with respect to different term orderings. It also applies when we have quotient bases which are not of type $\mathbb{T}^n \setminus \mathrm{LT}_\sigma(I_i)$ but which do yield a normal form vector map $\mathrm{NFV}_{\mathcal{O}_i}$ such as in Example 2.2.

Remark 3.2. Our presentation of Algorithm GBM favoured simplicity over efficiency. In reality it is better to build the matrix $M$ in triangular form by appending the vector $\mathbf{v}^*$ in step GBM6 and maintaining a list $\mathcal{O}^*$ to which we append $t - \sum_i a_i t_i$ rather than just $t$ in step GBM6. The elements of the list $\mathcal{O}$ are then merely the leading terms of the corresponding elements in $\mathcal{O}^*$. In this way step GBM4 runs faster.

In the last part of this section we describe an optimization of Theorem 3.1 which is based on the following remark (also noted in Faugère *et al.* (1989)).

Remark 3.3.

In step GBM3 the power product $t$ is either $1$ or of the form $t = x_j t'$ for some power product $t' \in \mathcal{O}$. By storing $\mathrm{NFV}_{\mathcal{O}_i}(t')$ for each $t' \in \mathcal{O}$ as the algorithm proceeds, we may compute the vector $\mathbf{v}$ cheaply by concatenating the images of each $\mathrm{NFV}_{\mathcal{O}_i}(t')$ under the linear transformation on $K^{\mu_i}$ which corresponds to the multiplication by $x_j$ on $P/I_i$. This is simple if each $I_i$ is represented by a vector $\mathbf{w}_i$ and multiplication matrices $M_{i1}, M_{i2}, \dots, M_{in}$. Here the commutativity of the multiplication matrices given in Proposition 2.6 is crucial.

**GBM3***bis* If $t = 1$ then put $\mathbf{v} = \mathbf{w}_1 \oplus \cdots \oplus \mathbf{w}_s$. Otherwise $t = x_j t'$ for some indeterminate $x_j$ and some $t' \in \mathcal{O}$, in which case put $\mathbf{v} = M_{1j}\mathbf{v}'_1 \oplus \cdots \oplus M_{sj}\mathbf{v}'_s$ where $\mathbf{v}'_1 \oplus \cdots \oplus \mathbf{v}'_s$ is the vector stored when processing $t'$.

## 4. The General Projective BM-Algorithm

In this section we shall develop along another direction the material explained in Section 2. In order to use it to compute intersections of homogeneous ideals, we shall need to extend those methods to the following general setting. Note that throughout this section all ideals are implicitly saturated and of algebraic dimension one (i.e. corresponding to geometrical objects of projective dimension zero).

Let $K$ be a field and $P = K[x_0, \dots, x_n]$. Let $P = \oplus_{d\in\mathbb{N}} P_d$ be standard graded, i.e. graded by $\deg(x_0) = \cdots = \deg(x_n) = 1$, and let $I \subseteq P$ be a homogeneous ideal. For every $d \in \mathbb{N}$, let $\mathcal{O}_d = (t_{d1}, \dots, t_{d\mu_d})$ be a tuple of homogeneous polynomials of degree $d$ such that the residue classes $\{\bar{t}_{d1}, \dots, \bar{t}_{d\mu_d}\}$ under the canonical map $\pi_d : P_d \longrightarrow P_d/I_d$ form a basis of $P_d/I_d$ as a $K$-vector space.

As in Section 2, we introduce the notion of the **normal form vector** $\mathrm{NFV}_{\mathcal{O}_d}(f)$ for $f \in P_d$ and the map $\mathrm{NFV}_{\mathcal{O}_d} : P_d \longrightarrow K^{\mu_d}$. The tuple $(\mathrm{NFV}_{\mathcal{O}_d})_{d\in\mathbb{N}}$ will be called a **graded normal form vector map**. Clearly, such maps must be explicit if we are to perform calculations with them.

The next proposition generalizes Proposition 2.4 and can be proved in the same way (with the obvious changes).

Proposition 4.1. *For every $d \in \mathbb{N}$, let $\mu_d \geq 1$ and let $\varphi_d : P_d \longrightarrow K^{\mu_d}$ be a $K$-linear, surjective map. The following conditions are equivalent.*

- *a) The set $\underset{d\in\mathbb{N}}{\oplus} \mathrm{Ker}(\varphi_d)$ is a homogeneous ideal in $P$.*
- *b) The tuple $(\varphi_d)_{d\in\mathbb{N}}$ is a graded normal form vector map, i.e. there is a homogeneous ideal $I$ and tuples $\mathcal{O}_d$ such that for every $d \in \mathbb{N}$ the map $\varphi_d = \mathrm{NFV}_{\mathcal{O}_d}$.*
- *c) For every $d \in \mathbb{N}$, the map $\varphi_d$ is the composition of a map $\mathrm{NFV}_{\mathcal{O}_{\sigma,d}}$, where $\mathcal{O}_{\sigma,d}$ is the degree $d$ part of the complement of some leading term ideal, with a linear base change $K^{\mu_d} \longrightarrow K^{\mu_d}$.*

Let us see an example of a graded normal form vector map.

Example 4.2. Let $P = K[x, y, z]$, and let $I \subset P$ be the ideal generated by $G = \{x^2 - 4xz + 4z^2, xy - xz - 2yz + 2z^2, y^2 - 2yz + z^2\}$, which is also the reduced `Lex`-Gröbner basis of $I$. Macaulay's Basis Theorem tells us that the residue classes of

$$\mathcal{O} = \mathbb{T}^3 \setminus \mathrm{LT}_{\mathtt{Lex}}(I) = \{z^d \mid d \in \mathbb{N}\} \cup\ x \cdot \{z^d \mid d \in \mathbb{N}\} \cup\ y \cdot \{z^d \mid d \in \mathbb{N}\}$$

form a $K$-basis of $P/I$. Therefore, if we let $\mathcal{O}_d = (z^d, xz^{d-1}, yz^{d-1})$ for all $d \in \mathbb{N}$, we get a graded normal form vector map $(\mathrm{NFV}_{\mathcal{O}_d})_{d\in\mathbb{N}}$ where $\mathrm{NFV}_{\mathcal{O}_d} : P_d \longrightarrow K^3$ is given by $\mathrm{NFV}_{\mathcal{O}_d}(f) = (a, b, c)$ such that $\mathrm{NR}_G(f) = az^d + bxz^{d-1} + cyz^{d-1}$ is the normal remainder returned by the Division Algorithm (see Kreuzer and Robbiano (2000), Section 1.6).

Now we examine the case when we have a finite number of graded normal form vector maps as before. Let $s \in \mathbb{N}$ be a positive integer, and $I_1, \dots, I_s$ be homogeneous ideals in $P$. For each ideal $I_i$ we have a collection of tuples $\mathcal{O}_{i,d}$ the residue classes of whose components form a $K$-vector space basis of $P_d/(I_i)_d$. Furthermore, we assume that $(\mathrm{NFV}_{\mathcal{O}_{i,d}})_{d\in\mathbb{N}}$ is an explicit graded normal form vector map for $i = 1, \dots, s$.

This is the setting for a generalized projective BM-algorithm. The main difference with the affine case is that we need an extra piece of information for the termination of the algorithm. Abstractly speaking, what we need is a **stopping criterion**.

Definition 4.3. For a computation which proceeds degree by degree, a **stopping criterion** is a logical condition which depends only on the data obtained in the computation up to the current degree $d$, and which, if satisfied, guarantees that the complete result has been attained.

For the computation of the homogeneous vanishing ideal of a zero-dimensional scheme in $\mathbb{P}^n_K$, the following simple stopping criterion can be used. Recall that the **Hilbert function** of a standard graded $K$-algebra $R$ is the map $\mathrm{HF}_R : \mathbb{N} \longrightarrow \mathbb{N}$ defined by $i \mapsto \dim_K(R_i)$. The basic properties of the Hilbert function $\mathrm{HF}_{P/I}$ for one-dimensional saturated homogeneous ideals $I \subset P$ needed in the proof below can be found in (Kreuzer, 1994).

Proposition 4.4. **(Naive Stopping Criterion)**
*In the setting of the generalized projective BM-algorithm, if the ideals $I_1, \ldots, I_s$ are one-dimensional and saturated, then writing $\mu_i = \mathrm{mult}\,(P/I_i)$ for each $i$, we can stop the computation of $I = I_1 \cap \cdots \cap I_s$ after we have finished computing its homogeneous generators of degree $\le \mu_1 + \cdots + \mu_s$.*

Proof. The multiplicity of the ring $P/I$ is $\mu \le \mu_1 + \cdots + \mu_s$. Since the ideals $I_1, \ldots, I_s$ are saturated, the ring $P/I$ is Cohen-Macaulay and its Hilbert function increases strictly until it reaches the value $\mu$ in some degree $d$. From there on it is constant, i.e. we have $\dim_K(P/I)_i = \mu$ for $i \ge d$. In particular, it is clear that $\mathrm{HF}_{P/I}(i) = \mu$ for $i \ge \mu - 1$. Since $\mathrm{HF}_{P/I} = \mathrm{HF}_{P/\,\mathrm{LT}_\sigma(I)}$ for every term ordering $\sigma$, it follows as in the proof of (Abbott *et al.*, 2000), Prop. 3.2, that $\mathrm{LT}_\sigma(I)$ is generated in degrees $\le \mu$. After we have finished computing the homogeneous generators of $I$ of degree $\le \mu$, the ideal $J = (I_{\le \mu})$ which they generate has therefore the leading term ideal $\mathrm{LT}_\sigma(J) = \mathrm{LT}_\sigma(I)$, and we conclude $J = I$. □

In the paper (Abbott *et al.*, 2000) we presented a stopping criterion which is usually much better and which can be used in our situation, too. For completeness we restate it here. We recall that two power products $t, t'$ are **connected** if there exist indeterminates $x, x'$ such that $x't = xt'$.

Theorem 4.5. **(Projective Stopping Criterion)**
*In the setting of the generalized projective BM-algorithm, let $\sigma$ be a term ordering on $\mathbb{T}^{n+1}$ and let $I = I_1 \cap \cdots \cap I_s$ where the homogeneous ideals $I_1, \ldots, I_s$ are one-dimensional and saturated. Moreover, suppose we have computed homogeneous polynomials, each one lying in the intersection and whose leading terms generate $(\mathrm{LT}_\sigma(I)_{\le d})$ for some degree $d \ge 1$, and also that the following conditions hold:*

*a)* $\mathrm{HF}_{P/\,\mathrm{LT}_\sigma(I)}(d) = \mathrm{mult}(P/I_1) + \cdots + \mathrm{mult}(P/I_s)$ *or*
$d > 0$ *and* $\mathrm{HF}_{P/\,\mathrm{LT}_\sigma(I)}(d) = \mathrm{HF}_{P/\,\mathrm{LT}_\sigma(I)}(d-1)$

*b) For each $i = 0, \ldots, n$, every power product in the connected component of $x_i^d$ in $(\mathbb{T}^{n+1} \setminus \mathrm{LT}_\sigma(I))_d$ is divisible by $x_i$.*

*Then the homogeneous polynomials computed so far are a $\sigma$-Gröbner basis of $I$.*

Now we are ready to prove the main result of this section: the Generalized Projective BM-Algorithm.

Theorem 4.6. **(Algorithm PBM)**

*In the above setting, let $\sigma$ be a term ordering on the monoid $\mathbb{T}^{n+1}$ of terms of $P = K[x_0, \dots, x_n]$. Assume that there exists a function* `StoppingCriterion`$(G)$ *which returns* `TRUE` *if the set $G$ computed so far is a $\sigma$-Gröbner basis of $I = I_1 \cap \dots \cap I_s$, and* `FALSE` *otherwise.*

*Consider the following sequence of instructions.*

**PBM1** *Start with empty lists $G = [\,]$, $H = [\,]$, a list $L = [1]$, and $d = 0$.*

**PBM2** *Apply the function* `StoppingCriterion`$(G)$*. If it returns* `TRUE`*, return the list $G$ and stop. Otherwise increase $d$ by one, set $M = (m_{ij})$ to be a matrix over $K$ with zero rows and $\ell = \sum_{i=1}^{s} \#(\mathcal{O}_{i,d})$ columns, and let $L$ be the list of all terms of degree $d$ which are not multiples of any element of $\{\mathrm{LT}_\sigma(g) \mid g \in G\}$.*

**PBM3** *If $L = \emptyset$, go to step PBM2. Otherwise choose the term $t = \min_\sigma(L)$ and remove it from $L$.*

**PBM4** *Compute the vector $\mathbf{v} = (\mathrm{NFV}_{\mathcal{O}_{1,d}}(t) \oplus \dots \oplus \mathrm{NFV}_{\mathcal{O}_{s,d}}(t)) \in K^\ell$ and reduce it against the rows of $M$ to obtain*

$$\mathbf{v}^* = \mathbf{v} - \sum_i a_i\,(m_{i1}, \dots, m_{i\ell}) \qquad\qquad \text{with } a_i \in K$$

**PBM5** *If $\mathbf{v}^* = (0, \dots, 0)$, then append the polynomial $t - \sum_i a_i h_i$ to the list $G$, where $h_i$ is the $i^{\text{th}}$ element of the list $H$. Continue with step PBM3.*

**PBM6** *Otherwise $\mathbf{v}^* \neq 0$, so append $\mathbf{v}^*$ as a new row to $M$ and $t - \sum_i a_i h_i$ as a new element to $H$. Continue with step PBM3.*

*This is an algorithm which returns the reduced $\sigma$-Gröbner basis $G$ of the ideal $I = I_1 \cap \dots \cap I_s$.*

Proof. This can be shown in the same way as Thm. 3.12 in (Abbott *et al.*, 2000). We note that $v = (0, \dots, 0)$ is equivalent to $f = t - \sum_i a_i h_i \in I_d$, since the Chinese Remainder Theorem yields that $f \in I_d$ if and only if $\mathrm{NFV}_{\mathcal{O}_{1,d}}(f) = \dots = \mathrm{NFV}_{\mathcal{O}_{s,d}}(f) = 0$. □

Remark 4.7. Let $\sigma$ be a term ordering on $\mathbb{T}^{n+1}$, let $\mathbb{X} \subset \mathbb{P}^n_K$ be a zero-dimensional subscheme, let $\mathrm{Supp}(\mathbb{X}) = \{P_1, \dots, P_s\}$, and, for $i = 1, \dots, s$, let $I_i \subset P$ be the vanishing ideal of $\mathbb{X}$ at $P_i$. Then the reduced $\sigma$-Gröbner basis of the homogeneous vanishing ideal $I_\mathbb{X} \subset P$ of $\mathbb{X}$ can be computed as follows.

1) For $i = 1, \dots, s$ and $d \in \mathbb{N}$, find a tuple $\mathcal{O}_{i,d}$ of homogeneous polynomials of degree $d$ and a map $\mathrm{NFV}_{\mathcal{O}_{i,d}} : P_d \longrightarrow K^{\mu_{i,d}}$ such that $(\mathrm{NFV}_{\mathcal{O}_{i,d}})_{d \in \mathbb{N}}$ is an effective graded normal form vector map with kernel $I_i$. (For instance, one can compute a $\sigma_i$-Gröbner basis $G_i$ of $I_i$ for some term ordering $\sigma_i$ and use the normal remainder map $\mathrm{NR}_{\sigma_i, G_i}$ as in Example 4.2 to define $\mathrm{NFV}_{\mathcal{O}_{i,d}}$.)
2) Apply algorithm PBM to this situation using the stopping criterion of Theorem 4.5.

Remark 4.8. As is well known $\mathbb{A}^n_K$ can be readily embedded into $\mathbb{P}^n_K$, thus Algorithms GBM and PBM are closely related. We may use Algorithm GBM instead of Algorithm PBM if the chosen ordering $\sigma$ is degree compatible and all the points associated to the input ideals have non-zero coordinate corresponding to the $\sigma$-smallest indeterminate. In such special cases Algorithm GBM computes its answer doing less arithmetic than Algorithm PBM.

## 5. Modular Techniques

When we work over the base field $K = \mathbb{Q}$, algorithms for computing Gröbner bases like our algorithm GBM may perform poorly due to the problem of coefficient growth. In a previous paper Abbott *et al.* (2000) we presented some modular methods to overcome this difficulty in the BM-algorithm. More precisely, we presented a version of this algorithm which computes the desired Gröbner bases modulo different primes and then reconstructs the solution over $\mathbb{Q}$ using **Chinese Remaindering** techniques. The analogous development of algorithm GBM is impeded by the lack of a good criterion for checking the correctness of the reconstructed basis.

Therefore we shall now present a different way of exploiting modular methods in algorithm GBM — an idea similar in spirit to the method of **Gröbner traces**. The same idea applies equally to algorithm PBM, but we shall leave it to the interested reader to work out the details.

So, let $K = \mathbb{Q}$, let $n \geq 1$, let $\sigma$ be a term ordering on $P = K[x_1, \ldots, x_n]$, let $s \in \mathbb{N}$ be a positive integer, and for $i = 1, \ldots, s$ let the zero-dimensional ideal $I_i \subset P$ be given by the vector $\mathbf{w}_i \in K$, and the multiplication matrices $M_{i1}, \ldots, M_{in}$ as described in Definition 2.7. Moreover, let $\mu_i = \dim_K(P/I_i)$ for $i = 1, \ldots, s$ and $\mu = \mu_1 + \cdots + \mu_s$.

Theorem 5.1. **(Modular Version of the General BM-Algorithm)**
*In the above situation, consider the following sequence of instructions.*

**MBM1** *Pick a prime number $p \in \mathbb{N}$ which does not divide the denominator of any entry in the matrices $M_{ij}$ or in the vectors $\mathbf{w}_i$ — so that reductions modulo $p$ exist.*

**MBM2** *Apply algorithm GBM over the field $\mathbb{F}_p$ to the modular reductions of the matrices $M_{ij}$ and of the vectors $\mathbf{w}_i$. From the result we use only the tuple of power products $\mathcal{O} = (t_1, \ldots, t_\nu)$, and $\hat{G} = (\hat{g}_1, \ldots, \hat{g}_r)$, the tuple of the leading power products of the computed Gröbner basis.*

**MBM3** *Construct a $\nu \times \mu$ matrix $M$ over $\mathbb{Q}$: each element $t_i \in \mathcal{O}$ gives one row being* $\mathrm{NFV}_{\mathcal{O}_1}(t_i) \oplus \cdots \oplus \mathrm{NFV}_{\mathcal{O}_s}(t_i)$

**MBM4** *Similarly construct a matrix $R$ of size $r \times \mu$ over $\mathbb{Q}$ whose $i^{\text{th}}$ row is the concatenated normal form vectors of $\hat{g}_i \in \hat{G}$.*

**MBM5** *Now solve the linear systems $LM = R$ over $\mathbb{Q}$ to obtain a matrix $L = (\lambda_{ij})$ of size $r \times \nu$.*

**MBM6** *If the system in step MBM5 admits no solution then go back to step MBM1. Otherwise form the polynomials $g_i = \hat{g}_i - \sum_{j=1}^{\nu} \lambda_{ij} t_j$ for each power product $\hat{g}_i \in \hat{G}$, and check whether $\lambda_{ij} = 0$ for all indices $j$ having $t_j >_\sigma \hat{g}_i$. If this is not the case, go back to step MBM1. Otherwise let $G = \{g_1, \ldots, g_r\}$, and return the pair $(G, \mathcal{O})$.*

*This is an algorithm which returns a list $\mathcal{O}$ whose components are precisely the elements of $\mathbb{T}^n \setminus \mathrm{LT}_\sigma(I)$ together with the reduced $\sigma$-Gröbner basis $G$ of the ideal $I = I_1 \cap \cdots \cap I_s$.*

Proof. The basic idea behind this proof is to consider running two copies of algorithm GBM, one on the inputs over $\mathbb{Q}$, the other on the modular images in $\mathbb{F}_p$. If the two runs follow the same path then the final modular result is just the modular reduction of the result over $\mathbb{Q}$. Otherwise we consider the point where the two runs first differ, and then deduce how to detect when the modular result is not good.

Let $G_0$ and $\mathcal{O}_0$ denote the result algorithm GBM would produce over $\mathbb{Q}$; similarly let $G_p$ and $\mathcal{O}_p$ denote the result over $\mathbb{F}_p$. We shall also refer to $M_0$, the final value of the matrix $M$ used during the run of algorithm GBM over $\mathbb{Q}$. To simplify later arguments we shall suppose that all entries in $M_0$ are integer, this being achieved by multiplying each row by its least common denominator — this assumption clearly does not affect the run of algorithm GBM over $\mathbb{Q}$. Further note that the rows of $M_0$ are linearly independent by construction, and so $\operatorname{rank}(M_0) = \nu$ where $\nu$ is the number of rows.

Of those primes satisfying the conditions in step MBM1 we shall say that a prime $p$ is *good* if $\mathcal{O}_p = \mathcal{O}_0$, and otherwise the prime is *bad.* For a bad prime $p$ we shall be interested in the "first difference" between $\mathcal{O}_p$ and $\mathcal{O}_0$, i.e. the $\sigma$-smallest power product $t$ which appears in one but not both. In fact, we claim that $t \in \mathcal{O}_0$ and $t \notin \mathcal{O}_p$. For suppose instead that $t \notin \mathcal{O}_0$. This means that the row comprising the concatenated normal form vectors of $t$ is $\mathbb{Q}$-linearly dependent on the rows corresponding to those elements of $\mathcal{O}_0$ which are $\sigma$-less than $t$, equivalently there is a polynomial with rational coefficients $t - \sum_i \mu_i t_i$ lying in the intsersection where each $t_i <_\sigma t$ and $t_i \in \mathcal{O}_0$. Now $p$ does not divide the denominator of any coefficient $\mu_i$ since otherwise, multiplying by the least power of $p$ to remove all factors of $p$ from the denominators would yield an $\mathbb{F}_p$-linear dependency among the rows corresponding to those elements of $\mathcal{O}_0$ which are $\sigma$-less than $t$; yet $\mathcal{O}_p$ and $\mathcal{O}_0$ contain the same elements up to $t$, which means there can be no $\mathbb{F}_p$-linear dependency.

Now these preliminaries are over our proof has three parts: $(i)$ there are only finitely many bad primes, $(ii)$ if the prime chosen in step MBM1 is bad then the checks in step MBM6 will detect this, and $(iii)$ if the prime chosen in step MBM1 is good then the checks in step MBM6 will pass and the correct result will be returned.

$(i)$ We first show that there are only finitely many bad primes. Suppose that $\mathcal{O}_p \neq \mathcal{O}_0$, and let $t$ be the $\sigma$-smallest element of $\mathcal{O}_0$ not in $\mathcal{O}_p$. Thus over $\mathbb{F}_p$ the normal form vector of $t$ is linearly dependent on the normal form vectors of those elements of $\mathcal{O}_0$ smaller than $t$. So in particular there is a linear relation modulo $p$ between the rows of $M_0$, so $M_0$ is not of full rank modulo $p$. Thus all bad primes must divide the determinants of all $\nu \times \nu$ minors of $M_0$, and hence the bad primes are only finite in number.

$(ii)$ Now we show that if the prime chosen in step MBM1 is bad then we discover this in step MBM6. So assume that $p$ is bad. If this leads to an insoluble linear system in step MBM5 we detect this in step MBM6, and start anew with a different prime. Otherwise a solution was found in step MBM5; if there are multiple solutions we may pick any one. Let $t$ be the $\sigma$-smallest element of $\mathcal{O}_0$ not in $\mathcal{O}_p$. The solution obtained in step MBM5 has represented the normal form vector of $t$ as a linear combination of the rows of $M$. Now, the rows of $M$ and $M_0$ differ only by a non-zero scalar multiple, so we also have a representation as a linear combination of the rows of $M_0$ involving the *same* rows. Since $t \in \mathcal{O}_0$ we know its row cannot be represented as a linear combination of those rows of $M_0$ corresponding solely to power products $\sigma$-smaller than $t$. Hence the linear combination obtained for $t$ must involve at least one row corresponding to a power product $\sigma$-greater than $t$. And this is what we check for in step MBM6.

$(iii)$ Now suppose that the prime chosen in step MBM1 is good. Thus we have $\mathcal{O}_p = \mathcal{O}_0$ at the end of step MBM2. The set $\mathcal{O}_p$ uniquely determines $\hat{G}$ which, by its uniqueness, must also be the set of leading power products of $G_0$. Hence the linear systems in step MBM5 admit a solution: the coefficients of the elements of $G_0$ give one solution. Moreover these solutions are unique since the matrix $M$ is of full rank: $M$ is the same as $M_0$ up to multiplication of rows by non-zero scalars. This unique solution clearly satisfies the test in step MBM6, and so the correct result is returned. □

Remark 5.2. Here we make some observations directed at potential implementers.

(a) If step MBM1 is executed several times (because the tests in step MBM6 fail) then on each occasion we must pick a prime $p$ different from those previously chosen.

(b) In step MBM2 the number of elements of $\mathcal{O}$ may be less than $\mu$ (e.g. if the input ideals are not pairwise comaximal).

(c) In step MBM3 in our implementation we exploit the idea expounded in Remark 3.3; the same idea can be used to compute the rows of $R$ cheaply from the rows in $M$.

(d) In step MBM5 a sophisticated linear system solver will, in its turn, exploit modular techniques (e.g. Hensel methods). Indeed the Gröbner basis discarded in step MBM2 could be used here, though this is unlikely to produce a measurable improvement in speed.

(e) The order of the rows of the matrices $M$ and $R$ constructed in steps MBM3 and MBM4 is quite unimportant so long as the interpretation in step MBM6 is consistent. Their rows are indexed by power products, and we can order these power products in any convenient manner (the use of tuples in step MBM2 is intended to indicate this).

## 6. Complexity on Fat Points

In Section 3, we saw how one can compute the vanishing ideal of a zero-dimensional subscheme $\mathbb{X}$ of an affine space $\mathbb{A}^n_K$ if we are given the zero-dimensional ideals defining $\mathbb{X}$ at the points of its support. Suppose some point $\mathbf{p} = (p_1, \dots, p_n) \in \operatorname{Supp}(\mathbb{X})$ is $K$-rational. Let $\mathfrak{m} = (x_1 - p_1, \dots, x_n - p_n)$ be the corresponding maximal ideal of $K[x_1, \dots, x_n]$, and let $I \subseteq \mathfrak{m}$ be the ideal defining $\mathbb{X}$ at $\mathbf{p}$. The following situation occurs frequently.

Definition 6.1. The point $\mathbf{p}$ is called a **fat point** of $\mathbb{X}$, if $I = \mathfrak{m}^d$ for some $d \geq 1$. The number $d$ is called the **order** of the fat point $\mathbf{p}$. We say that $\mathbb{X}$ is a **scheme of fat points** if $\operatorname{Supp}(\mathbb{X})$ is $K$-rational and every point of $\operatorname{Supp}(\mathbb{X})$ is a fat point of $\mathbb{X}$ (of some order).

Remark 6.2. Reduced points are fat points of order one. Not all non-reduced points are fat points; for instance, in $\mathbb{A}^2$ the point $(0,0)$ could be associated to the ideal $I = (x, y^2)$ which is not a power of $(x, y)$.

If the input ideals are ideals of fat points we can make step GBM3*bis* even faster. First we consider fat points located at the origin $(0, \dots, 0)$, then we observe that a simple change of coordinates allows us to handle any fat point in much the same way. In fact, the ideal of a fat point located at $(0, \dots, 0)$ is a monomial ideal, and the observations here apply to any monomial ideal.

Remark 6.3. Let $I$ be a monomial ideal, then a natural $K$-vector space basis for $P/I$ is given by the set $\mathcal{O}$ of all power products $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ not divisible by any monomial generator of $I$; if $I$ is the ideal of a fat point of order $d$ then the quotient basis is generated by all power products of degree $< d$. For this choice of basis the map $\mathrm{NFV}_{\mathcal{O}} : P \longrightarrow K^\mu$ is particularly easy to compute.

Furthermore, in step GBM3*bis* we compute $\mathrm{NFV}_{\mathcal{O}}(x_j t')$ directly from $\mathrm{NFV}_{\mathcal{O}}(t')$ by matrix multiplication. Now, using the natural quotient basis for a monomial ideal we have a special structure which permits us to replace the matrix multiplication by a simpler process. Indeed $\mathbf{v} = \mathrm{NFV}_{\mathcal{O}}(x_j t')$ may be obtained from $\mathbf{v}' = \mathrm{NFV}_{\mathcal{O}}(t')$ merely by changing the position of some coordinates and setting the rest to zero: the entries of a normal form vector are indexed by the power products in $\mathcal{O}$, so the entry of $\mathbf{v}$ indexed by $t$ is zero if $t$ is not divisible by $x_j$, otherwise it is equal to the entry of $\mathbf{v}'$ indexed by $t/x_j$. The necessary coordinate shifts can be effected optimally, in linear time, after a simple preprocessing phase (with complexity $O(n\mu^2)$).

Remark 6.4. To handle a fat point located away from the origin at $(p_1, \dots, p_n)$ we use the change of coordinates $x_i \mapsto x_i + p_i$. Note that this coordinate change affects step GBM3*bis*: the new normal form vector is now obtained by multiplying by $x_j + p_j$. We can do this multiplication in linear time by using coordinate shifts to multiply by $x_j$, and then adding $p_j$ times $\mathrm{NFV}_{\mathcal{O}}(t')$ to the result.

Remark 6.5. In general, for a fat point of order $d$ located at $(p_1, \dots, p_2)$, we have that $\mathrm{NFV}_{\mathcal{O}}(x_1^{\alpha_1} \cdots x_n^{\alpha_n})$ comprises exactly the coefficients of all terms of degree $< d$ in the polynomial $(x_1 + p_1)^{\alpha_1} \cdots (x_n + p_n)^{\alpha_n}$.

The normal form vector map described in Remark 6.4 is sufficiently well specified that we are able to analyse the complexity of Algorithm MBM in the special case of ideals of fat points with disjoint support with integer coordinates. If all the points are in fact simple then we obtain the same expected complexity as the algorithm in Abbott *et al.* (2000).

We shall express the complexity in terms of the following parameters:

$\mu$ the sum of the multiplicities of the input ideals;
$n$ the number of variables (i.e. the dimension of the ambient affine space);
$r$ the number of elements in the Gröbner basis;
$X$ a bound on the coordinates of the points supporting the ideals.

Note that since the fat points have disjoint support we know that the multiplicity of the intersection is equal to the sum of the multiplicities of the input ideals; in general, the sum is an upper bound.

Running Algorithm GBM over the finite field $\mathbb{F}_p$ has the same complexity as the classical Buchberger-Möller algorithm: the only difference is the way in which each vector $\mathbf{v}$ is computed, but the cost of reducing each vector exceeds the cost of creating it. We recall from Abbott *et al.* (2000) that the complexity is $O(\mu^2(r+\mu)(\log p)^2 + \mu^2 n^2)$ where the cost of an arithmetic operation in $\mathbb{F}_p$ is $O((\log p)^2)$.

Lemma 6.6. *Let $p_1, \dots, p_n \in \mathbb{R}$ and $\tau_1, \dots, \tau_n \in \mathbb{N}$. Put $\tau = \tau_1 + \cdots + \tau_n$ and $f = \prod (x_i + p_i)^{\tau_i}$. Then the coefficient in $f$ of any term of degree $d$ has magnitude bounded by $B_d$ the coefficient of $x^d$ in $(x+B)^\tau$ for any $B \ge \max\{|p_1|, \dots, |p_n|\}$. In particular, we have $B_d \le (B+1)^\tau$ whenever $\tau > 0$.*

Proof. Let $t$ be any power product. Then the magnitude of the coefficient of $t$ in $f$ is clearly bounded by the coefficient of $t$ in $\prod_{i=1}^n (x_i + |p_i|)^{\tau_i}$, and this in turn is clearly bounded by the coefficient of $t$ in $g = \prod_{i=1}^n (x_i + B)^{\tau_i}$. For any degree $d$ it is an elementary induction on $n$ to show that the coefficient of $x^d$ in $(x+B)^\tau$ is the sum of all coefficients of terms of degree $d$ in $g$. Finally, every coefficient in $(x+B)^\tau$ is non-negative and the sum of these coefficients is $(B+1)^\tau$, and therefore obviously an upper bound. □

Theorem 6.7. *Let $I_1, I_2, \dots, I_s$ be ideals in $\mathbb{Q}[x_1, \dots, x_n]$ of fat points with disjoint support at points with integer coordinates bounded by $X$. Let the multiplicity of each $I_j$ be $\mu_j$, and put $\mu = \sum \mu_j$. Supposing that the ideals are presented using the bases of Remark 6.4 then Algorithm MBM has expected bit complexity:*

$$O(\mu^5(r+\mu)\log^2(X+1) + \mu^2 n^2).$$

*Furthermore, for "generic" fat points with the* `DegRevLex` *ordering, Algorithm MBM has bit complexity*

$$O(\mu^4 d(r+\mu)\log^2(X+1) + \mu^2 n^2)$$

*where $d$ is such that $\binom{n+d-1}{d} = r + \mu$; this is better than the general expected complexity by a factor of about $\mu/d$.*

Proof. Recall from the proof of Theorem 5.1 that a prime is bad if and only if it divides a certain determinant whose value is independent of the choice of prime. Consequently, the probability of returning to step MBM1 at least $n$ times is bounded above by $2^{-n}$ (an upper bound for the probability that the determinant is divisible by each of the $n$ primes chosen). Furthermore for a fixed input and a given prime $p$ the cost of an iteration is in $O(\log^2 p)$. Thus provided the primes chosen do not vary wildly in size the expected cost is bounded by a constant times the cost of a single iteration. Our proof is completed by assuming the prime chosen in step MBM1 is good and then showing that each step of Algorithm MBM has cost not exceeding the complexity claimed above.

Steps MBM1 and MBM6 have negligible cost. The cost of the computation over $\mathbb{F}_p$ in step MBM2 is also dominated by the claimed complexity provided $\log p \le \mu\sqrt{\mu}\log X$, i.e. we must not pick huge primes.

We now estimate the cost of steps MBM3 and MBM4. We first estimate the cost of creating a single row of $M$ (or $R$). Using the method of Remark 6.4 we do at most $\mu$ multiplications between an integer of size $O(\log X)$ and another of size at most $O(\log(X^\mu))$, giving a total cost of $O(\mu^2\log^2 X)$ per row. Altogether there are $r+\mu$ rows, so their combined cost remains less than the claimed complexity.

Finally, we estimate the cost of solving the linear systems in step MBM5. We use Remark 6.5 to bound the size of the entries of the matrices $M$ and $R$ which are necessarily integer. The entries in $M$ come from normal form vectors of power products of total degree not exceeding $\mu-1$; similarly for the entries in $R$ except that the degree may be as high as $\mu$. We shall use Cramer's rule and Hadamard's bound on determinants to estimate the size of solutions to the linear system.

To use Hadamard's bound we need to calculate the Euclidean length of each row in $M$ and in $R$. Consider a row corresponding to a power product $t$ of degree $\tau > 0$. By Remark 6.5 and Lemma 6.6 the normal form vector of $t$ with respect to any of the ideals $I_j$ contains entries of magnitude at most $(X+1)^\tau$. Hence the Euclidean length of the entire row is at most $(X+1)^\tau\sqrt{\mu}$. For a row in $M$ we can take $\tau = \mu-1$, and a for a row in $R$ we take instead $\tau=\mu$.

By Cramer's rule every coordinate of any solution vector in the system solved in step MBM5 has numerator bounded by $\sqrt{\mu^\mu}(X+1)^{\mu^2-\mu+1} \in O((X+1)^{\mu^2})$, and denominator bounded by $\sqrt{\mu^\mu}(X+1)^{\mu^2-\mu} \in O((X+1)^{\mu^2})$. Hence the system can be solved using Chinese Remaindering techniques in time $O(\mu^5(r+\mu)\log^2(X+1))$ which is bounded by the claimed complexity.

In the generic case with the ordering `DegRevLex` we can take advantage of the fact that the highest degree power product appearing in the algorithm is typically far smaller than $\mu$. In fact, we need to go only as far as degree $d$ where $d$ is the smallest integer for which $\binom{n+d}{d} \ge r+\mu$. The reasoning above then allows us to reduce the complexity by a factor of $d/\mu$, except for the contribution $\mu^2 n^2$ (arising from the generation of power products within the call to Algorithm GBM). □

Remark 6.8. The worst case complexity is $\max(X^2,\mu^2)$ times the expected complexity, but is realised with exceedingly low probability. The product of the first $k$ primes clearly exceeds $k!$ whose logarithm lies in $O(k\log k)$. Hence there can be at most $\max(X^2,\mu^2)$ bad primes.

Remark 6.9. A more general analysis is difficult to manage because of the freedom of choice of the representation of an ideal using a normal form vector map. However, the formula of Theorem 6.7 still applies in the case where the ideals are given by integer vectors $\mathbf{w}_i$ of bounded length, and multiplication matrices $M_{ij}$ with integer entries and all of whose rows have 1-norm bounded by $X + 1$. This is because the crucial part of the proof is the estimation of the sizes of the entries in the matrices $M$ and $R$, and we can easily obtain the same estimates in this case.

## 7. Implementation Issues and Timing

Here we comment briefly on the differences between the new algorithm MBM and the old algorithm M we gave in Abbott *et al.* (2000). Foremost, the new algorithm is more general than the old one since it can be applied to any intersection of zero-dimensional ideals. Another notable difference is that in the new algorithm the "lifting" of the modular result is implicit in the process of solving the linear systems in step MBM5 whereas the "lifting" was an integral part of the old algorithm. Furthermore, the new algorithm is able to benefit directly and immediately from any improvement to algorithms for solving linear systems; instead the old algorithm could benefit only if such improvements can be fitted into its scheme.

The following tables give the timings for computing the intersections of various random ideals of non-simple points. The times reported are in seconds and represent averages of ten cases run on a 433 MHz Digital Alpha with 192Mb RAM; the code was compiled with `gcc -O2`. The timings reported in the tables below are a small selection; there are too many parameters to give comprehensive results. We have arbitrarily fixed the term ordering to be `DegRevLex`, the ambient spaces were chosen to be $\mathbb{A}^3_{\mathbb{Q}}$ and $\mathbb{P}^3_{\mathbb{Q}}$, and the coordinates of the points are random integers between $-99$ and $99$.

We do not give a table comparing the new implementation in the reduced case with that described in Abbott *et al.* (2000) since on all examples tried the times were virtually identical.

The table rows labelled "Order 2" give times for computing the intersection of ideals of fat points of order $2$; those labelled "Order 3" give times for computing the intersection of ideals of fat points of order $3$; and the row labelled "non-monomial" gives the times for computing the intersection of non-monomial zero-dimensional ideals of multiplicity $10$ (the same multiplicity as for a fat point of order $3$). The last column gives the size of the largest coefficient in the resulting Gröbner basis for the intersection of 15 ideals. In all cases the time spent constructing the multiplication matrices was excluded.

The Affine Case

| **DegRevLex** | 5 pts | 10 pts | 15 pts | 15 pts |
|---|---|---|---|---|
| Order 2 | 0.1 s | 1.3 s | 5.5 s | 330 digits |
| Order 3 | 1.4 s | 28 s | 140 s | 1100 digits |
| non-monomial | 3.4 s | 43 s | 210 s | 1250 digits |

The Projective Case

| **DegRevLex** | 5 pts | 10 pts | 15 pts | 15 pts |
|---|---|---|---|---|
| Order 2 | 0.6 s | 4.2 s | 15.3 s | 500 digits |
| Order 3 | 3.9 s | 68 s | 340 s | 1600 digits |

In section 6 we observed that the ideal of a fat point is a monomial ideal under some change of coordinates. The non-monomial ideals used in these tests are ideals which cannot be obtained by applying a change of coordinates to a monomial ideal.

## References

Abbott, J., Bigatti, A., Kreuzer, M., Robbiano, L. (2000) *Computing Ideals of Points.* J. Symb. Comput., 30: 341–356.

Bigatti, A., La Scala, R., Robbiano, L. (1999) *Computing Toric Ideals.* J. Symb. Comput., 27: 351–365.

Buchberger, B. (1985). *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, In Bose, N.K., editor, Multidimensional Systems Theory , pp. 184–232. Dordrecht. D. Reidel Publishing Co.

Buchberger, B., Möller, H. M. (1982). *The construction of multivariate polynomials with preassigned zeros.* In Calmet, J., editor, Proceedings of the European Computer Algebra Conference (EUROCAM '82), Lecture Notes in Comp. Sci., Vol. 144, pp. 24–31. Springer.

Caboara, M., Robbiano, L., (2001) Families of Estimable terms Proceedings of ISSAC 2001, pp.56–63, ACM Press

Capani, A., Niesi, G., Robbiano, L. (1998). *CoCoA, a system for doing Computations in Commutative Algebra.* Available via anonymous ftp from `cocoa.dima.unige.it`, version 4

Cioffi, F. (1998) Minimally generating ideals of points in polynomial time using linear algebra. Ricerche di Matematica, XLVII fasc. 2 (1998)

Cioffi, F., Orecchia, F. (2001) Computation of Minimal Generators of Ideals of Fat Points. Proceedings of ISSAC 2001, pp.72–76, ACM Press

Faugère, J. C., Gianni, P., Lazard, D., Mora, T. (1989). *Efficient computation of zero-dimensional Gröbner bases by change of ordering.* J. Symb. Comput. , 16(4): 329–344.

Kreuzer, M. (1994). *On the canonical module of a 0-dimensional scheme.* Can. J. Math., 46: 357–379.

Kreuzer, M., Robbiano, L. (2000) *Computational Commutative Algebra 1.* Heidelberg. Springer.

Lakshman, Y. (1991). *A single exponential bound on the complexity of computing Gröbner bases of zero dimensional ideals.* In Mora, T., Traverso, C., editors, Proceedings of MEGA-90 (Castiglioncello/Italy), Progress in Math., Vol. 94, pp. 227–234, Boston. Birkhäuser.

Marinari, M., Moeller, H., Mora, T. (1993). *Gröbner bases of ideals defined by functionals with an application to ideals of projective points.* Appl. Alg. in Eng., Comm. and Comp., 4(2): 103–145.

Mourrain, B. (1999). *A new criterion for normal form algorithms.* In Fossier, M., Imai, H., Lin, S., Poli, A., editors, Proceedings of the 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Comp. Sci., Vol. 1719, pp. 439–443. Springer.

Robbiano, L. (2001). *Zero-dimensional ideals or the inestimable value of estimable terms.* Preprint.