Fast reduction of bivariate polynomials with respect to sufficiently regular Gröbner bases

Joris van der Hoeven, Robin Larrieu

Fast reduction of bivariate polynomials with respect to sufficiently regular Gröbner bases

JORIS VAN DER HOEVEN[a], ROBIN LARRIEU[b]

Laboratoire d'informatique de l'École polytechnique
LIX, UMR 7161 CNRS
Campus de l'École polytechnique
1, rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing, CS35003
91120 Palaiseau, France

a. *Email:* vdhoeven@lix.polytechnique.fr
b. *Email:* larrieu@lix.polytechnique.fr

Let G be the reduced Gröbner basis of a zero-dimensional ideal $I \subseteq \mathbb{K}[X, Y]$ of bivariate polynomials over an effective field $\mathbb{K}$. Modulo suitable regularity assumptions on G and suitable precomputations as a function of G, we prove the existence of a quasi-optimal algorithm for the reduction of polynomials in $\mathbb{K}[X, Y]$ with respect to G. Applications include fast algorithms for multiplication in the quotient algebra $\mathbb{A} = \mathbb{K}[X, Y]/I$ and for conversions due to changes of the term ordering.

1. INTRODUCTION

Let $\mathbb{K}$ be an effective field and consider an algebra $\mathbb{A} = \mathbb{K}[X_1, \ldots, X_r]/I$ where I is a finitely generated ideal. For actual computations in $\mathbb{A}$, we have three main tasks:

T1. define a non-ambiguous representation for elements in $\mathbb{A}$;

T2. design a multiplication algorithm for $\mathbb{A}$;

T3. show how to convert between different representations for elements in $\mathbb{A}$.

Fast polynomial arithmetic based on FFT-multiplication allows for a quasi-optimal solution in the univariate case. However, reduction modulo an ideal of multivariate polynomials is non-trivial.

The most common approach for computations modulo ideals of polynomials is based on Gröbner bases. This immediately solves the first task, using the fact that any polynomial admits a unique normal form modulo a given Gröbner basis [4]. The second task is solved by reducing the product of two polynomials modulo the Gröbner basis. Finally, given a Gröbner basis with respect to a first term ordering, one may use the FGLM algorithm [9] to compute a reduced Gröbner basis with respect to a second term ordering; algorithms for the corresponding conversions are obtained as a by-product.

There is an abundant literature on efficient algorithms for the computation of Gröbner bases; see for example [7, 8, 9] and references therein. Although the worst case complexity is known to be very bad [23], polynomial complexity bounds (for the number of operations in $\mathbb{K}$ in terms of the expected output size) exist for many important cases of interest. For example, for fixed r, and using naive linear algebra on Macaulay matrices, one may show [22, 14, 15] that a sufficiently regular system of r equations of

degree δ can be solved in time $O(\delta^{\omega r})$. Here $\omega < 2.3728639$ is the exponent of matrix multiplication [11]. For such a system, the Bezout bound δ^r for the number D of solutions is reached, so the running time $O(D^\omega)$ is polynomial in the expected output size D. The implicit dependency of this bound on r can be improved by using the "matrix-F5" variant [2] of Faugère's F5 algorithm [8].

The F5 algorithm and all other currently known fast algorithms for Gröbner basis computations rely on linear algebra. At this point, one may wonder whether there is an intrinsic reason for this fact, or whether fast FFT-based arithmetic might be used to accelerate Gröbner basis computations. Instead of directly addressing this difficult problem, one may investigate whether such accelerations are possible for simpler problems in this area. One good candidate for such a problem is the reduction of a polynomial P with respect to a fixed reduced Gröbner basis $G = (G_0, ..., G_n)$. In that case, the algebra $\mathbb{A}$ is given once and for all, so it becomes a matter of precomputation to obtain G and any other data that could be useful for efficient reductions modulo G.

One step in this direction was made in [20]. Using relaxed multiplication [19], it was shown that the reduction of P with respect to G can be computed in quasi-linear time in terms of the size of the equation $P = Q_0 G_0 + \cdots + Q_n G_n + R$. However, even in the case of bivariate polynomials, this is not necessarily optimal. In order to see the reason for this, consider $\mathbb{A} = \mathbb{K}[X, Y]/I$, where I is the ideal generated by two generic polynomials of total degree δ. Then $\dim_{\mathbb{K}} \mathbb{A} = \delta^2$, but the Gröbner basis for I with respect the usual total degree ordering contains $\delta + 1$ polynomials with $\Theta(\delta^2)$ coefficients. This means that we need $\Theta(\delta^3)$ space, merely to write down G. One crucial prerequisite for even faster algorithms is therefore to design a terser representation for Gröbner bases.

The main aim of this paper is to show that it is actually possible to perform polynomial reductions in quasi-linear time in some very specific cases. For simplicity, we will restrict our attention to bivariate polynomials and to ideals that satisfy suitable regularity conditions. Because of all these precautions, we do not expect our algorithms to be very useful for practical purposes, but rather regard our work as a "proof of concept" that quasi-linear complexities are not deemed impossible to achieve in this context.

More precisely, with $\mathbb{A} = \mathbb{K}[X, Y]/I$ as above, our main results are as follows. We first introduce the concept of a "vanilla Gröbner basis" that captures the regularity assumptions that are needed for our algorithms. Modulo potentially expensive precomputations, we then present a more compact description of such a Gröbner basis G that holds all necessary information in $\tilde{O}(\delta^2)$ space. We next give an algorithm for reducing a bivariate polynomial of total degree d with respect to G in quasi-linear time $\tilde{O}(d^2 + \delta^2)$. In particular, multiplication in $\mathbb{A}$ can be done in time $\tilde{O}(\delta^2)$, which is intrinsically quasi-optimal. We also present an algorithm to convert between normal forms with respect to vanilla Gröbner bases for different monomial orderings. This algorithm is based on a Gröbner walk [6] with at most $O(\log \delta)$ intermediate monomial orderings; its complexity $\tilde{O}(\delta^2)$ is again quasi-optimal.

It is instructive to compare these complexity bounds with the complexities of naive algorithms that are commonly implemented in computer algebra systems. For multiplications in $\mathbb{A}$, one may precompute the $O(\delta^2) \times O(\delta^2)$ matrix that allows us to obtain the reduction of a product of two normal forms using a matrix-vector product of cost $O(\delta^4)$. Since the product of two normal forms can be computed in quasi-linear time $\tilde{O}(\delta^2)$, it follows that multiplications in $\mathbb{A}$ take time $O(\delta^4)$. Similarly, changes of monomial orderings lead to $\delta^2 \times \delta^2$-matrices for representing the corresponding base changes. Naive conversions can then be performed in time $O(\delta^4)$.

As a final remark, we notice that geometric methods provide an alternative to Gröbner basis techniques for the resolution of polynomial systems and computations in quotient algebras $\mathbb{A}$. Examples include the Kronecker solver [16] and Rouillier's RUR [25]. Such algorithms are often faster from a complexity point of view, but essentially only work for bases that correspond to lexicographical orders in the Gröbner basis setting. A similar remark applies to the elimination method by Auzinger-Stetter [1]

Notations and terminology. We assume that the reader is familiar with the theory of Gröbner basis and refer to [12, 3] for basic expositions. We denote the set of *monomials* in r variables by $\mathcal{M} := X_1^{\mathbb{N}} \cdots X_r^{\mathbb{N}} = \{X_1^{i_1} \cdots X_r^{i_r} : i_1, \ldots, i_r \in \mathbb{N}\}$. A *monomial ordering* $\prec$ on $\mathcal{M}$ is a total ordering that is compatible with multiplication. Given a polynomial in r variables $P = \sum_{M \in \mathcal{M}} P_M M \in \mathbb{K}[X_1, \ldots, X_r]$, its *support* $\operatorname{supp} P$ is the set of monomials $M \in \mathcal{M}$ with $P_M \neq 0$. If $P \neq 0$, then $\operatorname{supp} P$ admits a maximal element for $\prec$ that is called its *leading monomial* and that we denote by $\operatorname{lm}(P)$. If $M \in \operatorname{supp} P$, then we say that $P_M M$ is a *term* in P. Given a tuple $A = (A_0, \ldots, A_n)$ of polynomials in $\mathbb{K}[X_1, \ldots, X_r]$, we say that P is *reduced* with respect to A if $\operatorname{supp} P$ contains no monomial that is a multiple of the leading monomial of one of the A_i.

Unless stated otherwise, we will always work in the bivariate setting when $r = 2$, and use X and Y as our main indeterminates instead of X_1 and X_2. In particular, $\mathcal{M} := X^{\mathbb{N}} Y^{\mathbb{N}} = \{X^a Y^b : a, b \in \mathbb{N}\}$.

Acknowledgements. We thank the anonymous referees for their detailed comments and suggestions. We are aware that an example would be helpful for the intuition, unfortunately we were not able to give one because of the space constraints. Moreover, the reader should notice that a meaningful example cannot have a very small degree (say, at least 10), and that our setting requires non-structured dense polynomials, so that writing them down explicitly would hardly be readable.

2. VANILLA GRÖBNER BASES

Consider a zero-dimensional ideal I of $\mathbb{K}[X, Y]$ with Gröbner basis $G = (G_0, \ldots, G_n)$ with respect to a given monomial ordering $\prec$. We define the *degree* D of I to be the dimension of the quotient $\mathbb{K}[X, Y]/I$ as a $\mathbb{K}$-vector space. Our algorithms will only work for a special class of Gröbner bases with suitable regularity properties. For a generic ideal in the space of all zero-dimensional ideals with fixed degree D, we expect that these properties are always satisfied, although we have not proved this yet. For the time being, we define a *vanilla Gröbner basis* to be the Gröbner basis of an ideal of this type.

2.1. Monomial orderings

General monomial orderings that are suitable for Gröbner basis computations have been classified in [24]. For the purpose of this paper, it is convenient to restrict our attention to a specific type of bivariate monomial ordering that will allow us to explicitly describe certain Gröbner stairs and to explicitly compute certain dimensions.

DEFINITION 1. *Let $k \in \mathbb{N} \setminus \{0\}$. We define the **k-degree** of a monomial $X^a Y^b$ with $a, b \in \mathbb{N}$ by*

$$\deg_k(X^a Y^b) = a + kb.$$

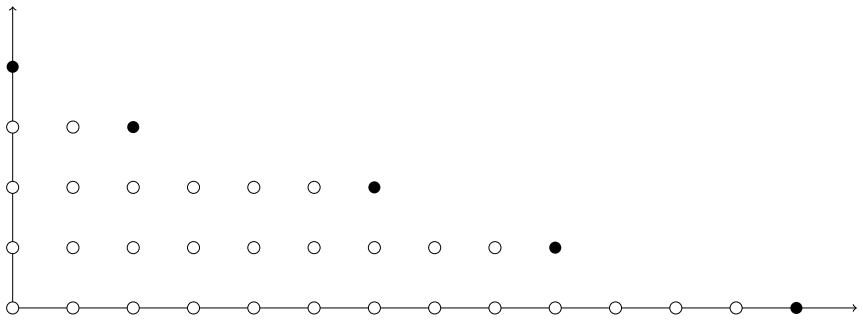

Figure 1. A vanilla Gröbner stairs with respect to $\prec_4$ ($D=30$, $n=4, q=1, r=2$).

We define the ***k-order*** *to be the monomial order $\prec_k$ such that*

$$X^a Y^b \prec_k X^u Y^v \Leftrightarrow \begin{cases} \text{either} & a+kb<u+kv \\ \text{or} & a+kb=u+kv \text{ and } a<u \end{cases}$$

The k-order $\prec_k$ is also known as the weighed degree lexicographic order for the weight vector $(1,k)$. Similarly, $\prec_1$ corresponds to the usual total degree order.

2.2. Vanilla Gröbner stairs

Consider a zero-dimensional ideal I of $\mathbb{K}[X, Y]$ of degree D with Gröbner basis $G = (G_0, \ldots, G_n)$ with respect to $\prec_k$. Let $\mathcal{N}_G$ be the set of monomials $X^a Y^b$ that are in normal form with respect to G. In other words, $\mathcal{N}_G$ corresponds to the set of D monomials "under the Gröbner stairs". For a sufficiently generic ideal of degree D, we expect $\mathcal{N}_G$ to consist exactly of the smallest D elements of $\mathcal{M}$ with respect to $\prec_k$.

DEFINITION 2. *We say that the leading monomials of G form a* ***vanilla Gröbner stairs*** *if $\mathcal{N}_G$ coincides with the set $\mathcal{M}_{k,D}$ of the D smallest elements of $\mathcal{M}$ for $\prec_k$.*

Figure 1 shows an example of a Gröbner basis whose leading monomials form a vanilla Gröbner stairs. We observe that the stair admits almost constant slope k. In fact, the set $\mathcal{M}_{k,D}$ can be described explicitly:

PROPOSITION 3. *Let I be an ideal of degree D with Gröbner basis G for $\prec_k$ with $k \geqslant 2$. Assume that the leading monomials of G form a vanilla Gröbner stairs and define*

$$\begin{aligned} n &:= \left\lceil \frac{\sqrt{8D/k+1}-1}{2} \right\rceil, \\ u &:= D - k\frac{n(n-1)}{2}, \\ q &:= u \operatorname{quo} n, \\ r &:= u \operatorname{rem} n. \end{aligned}$$

Then G has $n+1$ elements $G_0, \ldots, G_n$ and for $0 \leqslant i \leqslant n$, the leading monomial of G_i (denoted by M_i) can be expressed in terms of n, k, q, r. Assuming the basis elements are ordered such that the M_i's have increasing degree in the variable X, we have:

- $M_0 = Y^n$.
- *For all $i \in \{1, \ldots, r\}$, $M_i = X^{q+k(i-1)+1} Y^{n-i}$.*

- *For all* $i \in \{r+1, \ldots, n\}$, $M_i = X^{q+k(i-1)}\, Y^{n-i}$.

Proof. With this expression of M_i, we first notice that this sequence $M_0, \ldots, M_n$ can indeed be the leading monomials for a reduced Gröbner basis, that is M_i does not divide M_j for any $i \neq j$. This is clear for $(i, j) \neq (1, 0)$, so let us prove that M_1 does not divide M_0. We have $D = k n' (n'+1)/2$ with $n' := \left(\sqrt{8D/k+1} - 1\right)/2$, so that

$$\frac{k n (n-1)}{2} < D \leqslant \frac{k n (n+1)}{2}.$$

In particular, this implies $u > 0$, whence $q > 0$ or $r > 0$. Remains to prove that the sequence $M_0, \ldots, M_n$ form a vanilla Gröbner stairs (for a degree D ideal) as claimed. Indeed, there are D monomials under the stairs $M_0, \ldots, M_n$ (i.e. in normal form w.r.t. G), and we notice that a monomial M is under the stairs if and only if $M \prec_k M_{r+1}$. □

COROLLARY 4. *Let* $G = (G_0, \ldots, G_n)$ *be as above, and let* M_i *be the leading monomial of* G_i *for* $0 \leqslant i \leqslant n$. *With* q, r *as in Proposition 3, the k-degree of* M_i *is given by*

$$\deg_k M_i = \begin{cases} k n & \text{if } i = 0 \\ k(n-1) + q + 1 & \text{if } 0 < i \leqslant r \\ k(n-1) + q & \text{if } r < i \leqslant n \end{cases}.$$

In particular, for all $i \in \{1, \ldots, n\}$, *we have*

$$\deg_k M_i \leqslant \deg_k M_{i-1}, \text{ and } \deg_k M_1 - 1 \leqslant \deg_k M_i \leqslant \deg_k M_1.$$

Remark 5. The results of Proposition 3 and Corollary 4 remain valid for $\prec_1$ with some precautions: if $r \geqslant 1$, one has to leave out G_r since M_r is divisible by M_{r+1} with the given formulas. Then G consists of n elements $G_0, \ldots, G_{r-1}, G_{r+1}, \ldots, G_n$.

2.3. Existence of relations

The main reduction algorithm in this paper relies on a rewriting strategy that allows us to rewrite general linear combinations $A_0 G_0 + \cdots + A_n G_n$ of elements in the Gröbner basis as linear combinations of fewer elements. In particular, it should be possible to express each G_i as a linear combination of elements in a suitable subset Σ of $\{G_0, \ldots, G_n\}$ (this subset then generates the ideal I), with degrees that can be controlled.

It turns out that such a subset S may need to contain three elements at least, but that $\Sigma := \{G_0, G_1, G_n\}$ generically works. In order to control the degrees in the linear combinations, we may also consider intermediate sets between $\{G_0, G_1, G_n\}$ and the full set $\{G_0, \ldots, G_n\}$, such as $\Sigma_\ell := \{G_0, G_1, G_\ell, G_{2\ell}, \ldots, G_{\lfloor n/\ell \rfloor \ell}, G_n\}$ for various integer "step lengths" $\ell \geqslant 2$. This leads us to the following definition:

DEFINITION 6. *Let* $\ell \geqslant 1$ *be an integer and consider the set of indices*

$$I_\ell := \{0, 1, n\} \cup \{\ell, 2\ell, \ldots, \lfloor n/\ell \rfloor \ell\}. \tag{1}$$

We say that a family of polynomials $P_0, \ldots, P_n \in \mathbb{K}[X, Y]$ *is* ***retractive*** *for step length* ℓ *and k-degree* δ *if for all* $i \in \{0, \ldots, n\}$ *we can write*

$$P_i = \sum_{j \in I_\ell} A_j P_j$$

for some $(A_j)_{j\in I_\ell} \in \mathbb{K}[X,Y]^{I_\ell}$ *with* $\deg_k A_j \leqslant \delta$.

Consider a Gröbner basis G_0, ..., G_n as in Proposition 3 and a linear combination $C = \sum_{j\in I_\ell} A_j G_j$ with $\deg_k A_j \leqslant \delta$ for all $j \in I_\ell$. Making rough estimates, the number of monomials in $\mathcal{M}$ of k-degree $\leqslant d$ is $d^2/(2k)$, whence the number of monomials of k-degree between d and $d+\delta$ is bounded by $(d+\delta)\,\delta/k$. The set $\mathcal{N}_G = \mathcal{M}_{k,D}$ roughly corresponds to the set of monomials of k-degree $\leqslant n\,k$, whence the support of C contains at most $(nk+\delta)\,\delta/k$ monomials that are *not* in $\mathcal{N}_G$. Notice that such a combination C is uniquely determined by its terms *not* in $\mathcal{N}_G$: if all the terms of $C - C' \in I$ are in $\mathcal{N}_G$, then $C - C' = 0$ by definition of a Gröbner basis.

On the other hand the polynomials A_j with $j \in I_\ell$ are determined by approximately $(n/\ell)\,\delta^2/(2k)$ coefficients. As soon as $\delta > 2k\ell$, it follows that

$$(n/\ell)\,\delta^2/(2k) > (nk+\delta)\,\delta/k,$$

and it becomes likely that non-trivial relations of the type $G_i = \sum_{j\in I_\ell} A_j G_j$ indeed exist. A refined analysis and practical experiments show that the precise threshold is located at $\delta \geqslant k(2\ell-1)-1$, although we have no formal proof of this empirical fact.

2.4. Vanilla Gröbner bases

We are now in a position to describe the class of Gröbner bases with enough regularity for our fast reduction algorithm to work.

DEFINITION 7. *Let* $G = (G_0, \ldots, G_n)$ *be the reduced Gröbner basis for an ideal* $I \subset \mathbb{K}[X,Y]$ *with respect to* $\prec_k$. *We say that* G *is a* ***vanilla Gröbner basis*** *if*

a) the leading monomials of G *form a vanilla Gröbner stairs;*

b) the family $G_0, \ldots, G_n$ *is retractive for step length* ℓ *and k-degree* $k(2\ell-1)-1$, *for* $\ell = 2, \ldots, n$.

It appears that reduced Gröbner bases of sufficiently generic ideals are always of vanilla type, although we have not been able to prove this so far. We even do not know whether vanilla Gröbner bases exist for arbitrary fields $\mathbb{K}$ (with sufficiently many elements) and degrees D. Nevertheless, practical computer experiments suggest that sufficiently random ideals of degree D admit Gröbner bases of this kind. More precisely, we have checked this (up to degrees in the range of a few hundreds) for ideals that are generated as follows by two random polynomials:

- for $I = (A(X), Y - B(X))$, where A and B are random univariate polynomials of degrees D and $D-1$, and for any ordernig $\prec_k$;
- for $I = (A,B)$, where A and B are random bivariate polynomials of total degree δ (in this case the degree of the ideal is $D = \delta^2$), and for any ordering $\prec_k$ with $k \geqslant 2$;
- for $I = (A,B)$, where A and B are random bivariate polynomials of degree δ in both variables (in this case the degree of the ideal is $D = 2\delta^2$), and for any ordering $\prec_k$ with $k \geqslant 2$.

In each of these cases, the threshold $k(2\ell-1)-1$ seems to be sharp. Nevertheless, for our complexity bounds, a threshold of the type $Kk\ell$ would suffice, for any constant $K > 0$.

3. ALGORITHMIC PREREQUISITES

In this section, we quickly review some basic complexities for fundamental operations on polynomials over a field $\mathbb{K}$. Notice that results presented in this section are not specific to the bivariate case. Running times will always be measured in terms of the required number of field operations in $\mathbb{K}$.

3.1. Polynomial multiplication

We denote by $\mathsf{M}(d)$ the cost of multiplying two dense univariate polynomials of degree d in $\mathbb{K}[X]$. Over general fields, one may take [27, 26, 5]

$$\mathsf{M}(d) = O(d \log d \log\log d).$$

In the case of fields of positive characteristic, one may even take $\mathsf{M}(d) = O(d \log d\, 4^{\log^* d})$, where $\log^* d$ denotes the iterated logarithm [17, 18]. We make the customary assumptions that $\mathsf{M}(d)/d$ is increasing and that $\mathsf{M}(2d) = O(\mathsf{M}(d))$, with the usual implications, such as $\mathsf{M}(d) + \mathsf{M}(e) \leqslant \mathsf{M}(d+e)$.

For multivariate polynomials, the cost of multiplication depends on the geometry of the support. The multiplication of dense bivariate "block" polynomials in $\mathbb{K}[X_1, \ldots, X_r]$ of degree $< d_i$ in each variable X_i can be reduced to multiplication of univariate polynomials of degree $< 2^{r-1} d_1 \cdots d_r$ using the well known technique of Kronecker substitution [12]. More generally, for polynomials such that the support of the product is included in an initial segment with d elements, it is possible to compute the product in time $O(\mathsf{M}(d))$. Here an initial segment of $\mathcal{M}$ is a subset $\mathcal{S}$ such that all divisors of any monomial $M \in \mathcal{S}$ are again in $\mathcal{S}$.

For the purpose of this paper, we need to consider dense polynomials P in $\mathbb{K}[X, Y]$ whose supports are contained in sets of the form $S_{l,h} := \{M \in \mathcal{M} : l \leqslant \deg_k M < h\}$. Modulo the change of variables $X^a Y^b \to T^{a+kb} U^b$, such a polynomial can be rewritten as $P(X, Y) = T^l \tilde{P}(T, U)$, where the support of $\tilde{P}$ is an initial segment with the same size as $S_{l,h}$. For a product of two polynomials of this type with a support of size d, this means that the product can again be computed in time $O(\mathsf{M}(d))$.

3.2. Relaxed multiplication

For the above polynomial multiplication algorithms, we assume that the input polynomials are entirely given from the outset. In specific settings, the input polynomials may be only partially known at some point, and it can be interesting to anticipate the computation of the partial output. This is particularly true when working with (truncated) formal power series $f = f_0 + f_1 z + \cdots \in \mathbb{K}[[z]]$ instead of polynomials, where it is common that the coefficients are given as a stream.

In this so-called "relaxed (or online) computation model", the coefficient $(fg)_d$ of a product of two series $f, g \in \mathbb{K}[[z]]$ must be output as soon as $f_0, \ldots, f_d$ and $g_0, \ldots, g_d$ are known. This model has the advantage that subsequent coefficients $f_{d+1}, f_{d+2}, \ldots$ and $g_{d+1}, g_{d+2}, \ldots$ are allowed to depend on the result $(fg)_d$. This often allows us to solve equations involving power series f by rewriting them into *recursive equations* of the form $f = \Phi(f)$, with the property that the coefficient $\Phi(f)_{d+1}$ only depends on earlier coefficients $f_0, \ldots, f_d$ for all d. For instance, in order to invert a power series of the form $1 + zg$ with $g \in \mathbb{K}[[z]]$, we may take $\Phi(f) = 1 - zfg$. Similarly, if $\mathbb{K}$ has characteristic zero, then the exponential of a power series $g \in \mathbb{K}[[z]]$ with $g_0 = 0$ can be computed by taking $\Phi(f) = 1 + \int fg'$.

From a complexity point of view, let $\mathsf{R}(d)$ denote the cost of the relaxed multiplication of two polynomials of degree $<d$. The relaxed model prevents us from directly using fast "zealous" multiplication algorithms from the previous section that are typically based on FFT-multiplication. Fortunately, it was shown in [19, 10] that

$$\mathsf{R}(d) = O(\mathsf{M}(d)\log d). \tag{2}$$

This relaxed multiplication algorithm admits the advantage that it may use any zealous multiplication as a black box. Through the direct use of FFT-based techniques, the following bound has also been established in [21]:

$$\mathsf{R}(d) = d\log d\, e^{O(\sqrt{\log\log d})}.$$

In the sequel, we will only use a suitable multivariate generalization of the algorithm from [19, 10], so we will always assume that $\mathsf{R}(d)$ is of the form (2). In particular, we have $\mathsf{R}(d)+\mathsf{R}(e)\leqslant\mathsf{R}(d+e)$.

3.3. Polynomial reduction

Let us now consider a Gröbner basis of an ideal in $\mathbb{K}[X_1, \ldots, X_r]$, or, more generally, an auto-reduced tuple $A = (A_0, \ldots, A_n)$ of polynomials in $\mathbb{K}[X_1, \ldots, X_r]$. Then for any $P\in\mathbb{K}[X_1,\ldots,X_r]$, we may compute a relation

$$P = Q_0A_0 + \cdots + Q_nA_n + R$$

such that R is reduced with respect to A. We call $(Q_0,\ldots,Q_n,R)$ an *extended reduction* of P with respect to A.

The computation of such an extended reduction is a good example of a problem that can be solved efficiently using relaxed multiplication and recursive equations. For a multivariate polynomial T with dense support of any of the types discussed in section 3.1, let $|T|$ denote a bound for the size of its support. With $\mathsf{R}(d)$ as in (2), it has been shown[1] in [20] that the *quotients* $Q_0,\ldots,Q_n$ and the *remainder* R can be computed in time

$$\mathsf{R}(|Q_0A_0|) + \cdots + \mathsf{R}(|Q_nA_n|) + O(|R|). \tag{3}$$

This implies in particular that the extended reduction can be computed in quasi-linear time in the size of the equation $P = Q_0A_0 + \cdots + Q_nA_n + R$. However, as pointed out in the introduction, this equation is in general much larger than the input polynomial P.

Extended reductions $(Q_0, \ldots, Q_n, R)$ are far from being unique (only R is unique, and only if A is a Gröbner basis). The algorithm from [20] for the computation of an extended reduction relies on a *selection strategy* that selects a particular index $i_M\in\mathcal{I}_M := \{i\in\{0,\ldots,n\}:\mathrm{lm}(A_i)\,|\,M\}$ for every monomial $M\in\mathcal{M}$ such that $\mathcal{I}_M$ is non-empty. The initial formulation [20] used the simplest such strategy by taking $i_M = \min\mathcal{I}_M$, but the complexity bound (3) holds for any selection strategy. Now the total size of all quotients $Q_0,\ldots,Q_n$ may be much larger than the size of P for a general selection strategy. One of the key ingredients of the fast reduction algorithm in this paper is the careful design of a "dichotomic selection strategy" that enables us to control the degrees of the quotients.

Remark 8. The notion of selection strategy is somewhat similar to the concept of *involutive division* introduced for the theory of involutive bases [13], although our definition is more permissive.

1. The results from [20] actually apply for more general types of supports, but this will not be needed in this paper.

4. TERSE REPRESENTATIONS OF VANILLA GRÖBNER BASES

Let $G = (G_0, ..., G_n)$ be a vanilla Gröbner basis of some ideal $I \subseteq \mathbb{K}[X, Y]$ with respect to $\prec_k$ and assume the notations from Proposition 3. We recall from the introduction that a major obstruction for the design of reduction algorithms that run in quasi-linear time $\tilde{O}(k\,n^2 + d^2\,/\,k)$ is that it requires space $\Theta(k\,n^3)$ to explicitly write down the full basis G. The aim of this section is to introduce a suitable "terse representation" that can be stored in space $O(k\,n^2 \log n)$, but that still contains all necessary information for efficient computations modulo G.

4.1. Retraction coefficients

For each $\ell \geqslant 1$, let I_ℓ be as in (1). Also, for $\lambda \in \{0, ..., \lceil \log_2 n \rceil\}$, let J_λ be a shorthand for I_{2^λ}. Since $G = (G_0, ..., G_n)$ is a vanilla Gröbner basis, Definition 7 ensures in particular the existence of coefficients $C_{\lambda,i,j} \in \mathbb{K}[X, Y]$ for $\lambda \in \{0, ..., \lceil \log_2 n \rceil - 1\}$ and $i \in J_\lambda \setminus J_{\lambda+1}$ and $j \in J_{\lambda+1}$, such that

$$\begin{aligned} G_i &= \sum_{j \in J_{\lambda+1}} C_{\lambda,i,j} G_j, \\ \deg_k C_{\lambda,i,j} &\leqslant k\,(2^{\lambda+2} - 1) - 1. \end{aligned}$$

We call these $C_{\lambda,i,j}$ the *retraction coefficients* for G. For each given i, λ, the computation of the retraction coefficients $C_{\lambda,i,j}$ reduces to a linear system of size $u \times v$ with $u, v = O(k\,n\,2^\lambda)$ (for the image space, consider only the monomials that are *above* the Gröbner stairs), which is easily solved by Gaussian elimination. Notice that the space needed to write the retraction coefficients is much smaller than the Gröbner basis:

LEMMA 9. *The family of all retraction coefficients for G takes space $O(k\,n^2 \log n)$.*

Proof. For every ℓ, there are $\lceil n\,/\,\ell \rceil + 1$ indices in I_ℓ, and we notice that $I_{2\ell} \subseteq I_\ell$. For any given λ, the retraction coefficients involve at most $n\,/\,2^{\lambda+1} + 1$ indices i and $n\,/\,2^{\lambda+1} + 2$ indices j, whence at most $n^2\,/\,4^{\lambda+1} + 3\,n\,/\,2^{\lambda+1} + 2$ pairs (i, j). Since the support of $C_{\lambda,i,j}$ has size $O(k\,4^\lambda)$, it follows that all retraction coefficient together require space

$$O\left(k \sum_{\lambda \leqslant \log_2 n} (n^2 + n\,2^\lambda + 4^\lambda)\right) = O(k\,n^2 \log n). \qquad \square$$

We observe that the space needed to write all relations is about the same size as the dimension of the quotient algebra $\mathbb{K}[X, Y]\,/\,I$, up to a logarithmic factor.

4.2. Upper truncations

For vanilla Gröbner bases, it is *a priori* possible to recover G from G_0, G_1 and G_n using the retraction coefficients: with $h = \lceil \log_2 n \rceil$, first compute $G_{2^{h-1}}$, next $G_{2^{h-2}}$ and $G_{3.2^{h-2}}$, and so on. In order to compute reductions of the form $P = Q_0 G_0 + \cdots + Q_n G_n + R$ efficiently, we will need slightly more information. In particular, we wish to access some of the head terms of the G_i. More precisely, if the quotient Q_i has degree d, then we need to know the terms of G_i with degree at least $\deg G_i - d$ in order to compute the quotient Q_i using a relaxed reduction algorithm.

DEFINITION 10. *Given a polynomial $P \in \mathbb{K}[X,Y]$, we define its* ***upper truncation with k-precision p*** *as the polynomial $P^{\#}$ such that*

- *all terms of $P^{\#}$ of k-degree less than $\deg_k P - p$ are zero;*
- *all terms of $P^{\#}$ of k-degree at least $\deg_k P - p$ are equal to the corresponding terms in P.*

Notice that this upper truncation $P^{\#}$ can be written using space $O((\deg_k P)\, p/k)$. For the reduction strategy that we plan to use, we will have

$$\deg_k Q_i < 2k2^{\mathrm{val}_2 i} \tag{4}$$

for all $i = 1, \ldots, n-1$, where $\mathrm{val}_2\, i$ denotes the 2-adic valuation of i. This motivates the following definition:

DEFINITION 11. *Let $G = (G_0, \ldots, G_n)$ be a vanilla Gröbner basis for an ideal $I \subseteq \mathbb{K}[X,Y]$ with respect to $\prec_k$. The* ***terse representation*** *of G consists of the following data:*

- *the sequence of truncated elements $G_0^{\#}, \ldots, G_n^{\#}$, where*
  - *$G_i^{\#} := G_i$ for $i \in \{0, 1, n\}$;*
  - *$G_i^{\#}$ is the upper truncation of G_i at precision $2k2^{\mathrm{val}_2 i}$ for all other i;*
- *the collection of all retraction coefficients $C_{\lambda,i,j}$ as in section 4.1.*

PROPOSITION 12. *The terse representation of G fits in space $O(kn^2 \log n)$.*

Proof. The upper truncation $G_i^{\#}$ requires space $O(k\, n\, 2^{\mathrm{val}_2 i})$ for all $1 < i < n$. For each $\lambda < \log_2 n$, there are at most $n/2^{\lambda}$ indices i such that $\mathrm{val}_2\, i = \lambda$; therefore, $G_2^{\#}, \ldots, G_{n-1}^{\#}$ take $O(k\, n^2 \log n)$ space. The elements $G_0^{\#}$, $G_1^{\#}$ and $G_n^{\#}$ require $O(k\, n^2)$ additional space, whereas the coefficients $C_{\lambda,i,j}$ account for $O(kn^2 \log n)$ more space, by Lemma 9. □

5. FAST REDUCTION

Let $G = (G_0, \ldots, G_n)$ be a vanilla Gröbner basis for an ideal $I \subseteq \mathbb{K}[X,Y]$ as in the previous section and assume that its terse representation has been precomputed. The goal of this section is to present our main algorithm that computes the extended reduction $P = Q_0 G_0 + \cdots + Q_n G_n + R$ of a polynomial $P \in \mathbb{K}[X,Y]/I$ of k-degree d in quasi-linear time $\tilde{O}(kn^2 + d^2/k)$. This is quasi-optimal with respect to the dimension of the quotient algebra $\dim_{\mathbb{K}} \mathbb{K}[X,Y]/I = \Theta(kn^2)$ and the size of the support $|P| = \Theta(d^2/k)$.

The reduction algorithm proceeds in two steps: in a first stage, we compute the quotients $Q_0, \ldots, Q_n$; we next evaluate the remainder $R := P - Q_0 G_0 - \cdots - Q_n G_n$ by rewriting the linear combination $Q_0 G_0 + \cdots + Q_n G_n$ using fewer and fewer terms.

5.1. Computing the quotients

To compute the quotients, we reduce P as in section 3.3 against the tuple $(A_0, \ldots, A_n) := (G_0^{\#}, \ldots, G_n^{\#})$, in such a way that the degrees of the quotients are bounded as in equation (4). This is done using the algorithm from [20], but with the following *dichotomic selection strategy*. Given a monomial $M \in \mathcal{M}$, we reduce M against A_{i_M}, where $i_M \in \mathcal{I}_M := \{i \in \{0, \ldots, n\} : \mathrm{lm}(A_i) \mid M\}$ is determined as follows:

- if $\mathrm{lm}(A_0)$ divides M, then take $i_M := 0$;

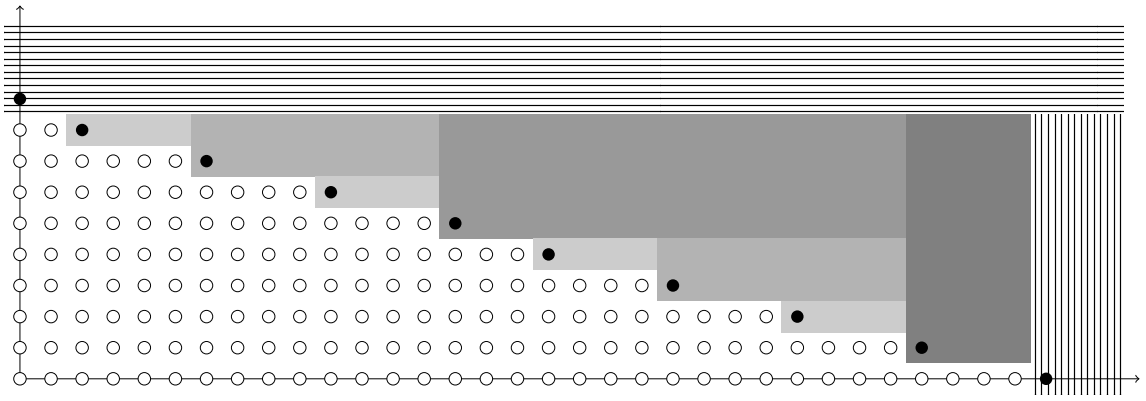

Figure 2. The dichotomic selection strategy: monomials falling in each area are reduced against the corresponding basis element.

- else if $\operatorname{lm}(A_n)$ divides M, then take $i_M := n$;
- else we take i_M to be the unique element in $\mathcal{I}_M$ with $\operatorname{val}_2 i_M = \max\{\operatorname{val}_2 i : i \in \mathcal{I}_M\}$.

This selection strategy is illustrated in Figure 2.

LEMMA 13. *Let $Q_0, \ldots, Q_n$ be the quotients obtained for the reduction of P with respect to $(G_0^\#, \ldots, G_n^\#)$ using the dichotomic selection strategy. Then the bound*

$$\deg_k(Q_i) < 2k2^{\operatorname{val}_2(i)}$$

holds for all $0 < i < n$, so that $|Q_0| + \cdots + |Q_n| = O(k\,n^2 + d^2/k)$, and the extended reduction $P = Q_0 G_0^\# + \cdots + Q_n G_n^\# + R^\#$ can be computed in time

$$O(\mathsf{R}(kn^2)\log n + \mathsf{R}(d^2/k)).$$

Proof. Let $X^a\,Y^b \in \operatorname{supp} Q_i$ with $0 < i < n$, so that $i = i_M$ for $M = X^a\,Y^b\,\operatorname{lm}(A_i)$, and denote $\ell := 2^{\operatorname{val}_2 i}$. Then we observe that $b < \ell$: if not, then $\operatorname{lm}(A_{i-\ell})$ would divide M, whereas $\operatorname{val}_2(i-\ell) > \operatorname{val}_2 i$. A similar reasoning with $A_{i+\ell}$ (or A_n, whenever $i + \ell > n$) shows that $a < k\ell$. It follows that $\deg_k(X^a Y^b) < 2k\ell$.

This also proves that $|Q_i| < 2\,\ell\,(2\,k\,\ell + 1) = O(k\,\ell^2)$ and $|Q_i\,G_i^\#| = O(k\,n\,\ell)$, for any $0 < i < n$. Since the number of indices $0 < i < n$ with $\ell = 2^{\operatorname{val}_2 i}$ is bounded by n/ℓ, we get

$$\begin{aligned} |Q_1| + \cdots + |Q_{n-1}| &= O(2kn + 4kn + \cdots + 2^{\lfloor \log_2 n \rfloor} kn) = O(kn^2) \\ \mathsf{R}(|Q_1 G_1^\#|) + \cdots + \mathsf{R}(|Q_{n-1} G_{n-1}^\#|) &= O(\mathsf{R}(kn^2)\log n). \end{aligned}$$

On the other hand, $\deg_k(Q_0 G_0^\#) \leqslant \deg_k P$ and $\deg_k(Q_n G_n^\#) \leqslant \deg_k P$, whence $|Q_0| + |Q_n| = O(d^2/k)$ and $\mathsf{R}(|Q_0\,G_0^\#|) + \mathsf{R}(|Q_n\,G_n^\#|) = O(\mathsf{R}(d^2/k))$. We conclude by applying the bound (3) for the complexity of polynomial reduction. □

The next important observation is that the quotients $Q_0, \ldots, Q_n$ obtained in the above way can actually be used as quotients for the extended reduction of P with respect to G:

PROPOSITION 14. *Let $Q_0, \ldots, Q_n$ be as in Lemma 13 and consider*

$$R := P - Q_0 G_0 - \cdots - Q_n G_n.$$

Then R is reduced with respect to G.

Proof. Let $R^\# := P - Q_0\,G_0^\# - \cdots - Q_n\,G_n^\#$. By construction, $R^\#$ is reduced with respect to $G^\# = (G_0^\#, \ldots, G_n^\#)$ and whence with respect to G since $\mathrm{lm}(G_i) = \mathrm{lm}(G_i^\#)$ for all i. For any $0 < i < n$, we also have $\deg_k(G_i - G_i^\#) < \deg_k G_i - 2k2^{\mathrm{val}_2 i}$, whence

$$\deg_k(Q_i G_i - Q_i G_i^\#) < \deg_k G_i - 1 \leqslant \min_{0 \leqslant j \leqslant n} \deg_k G_j$$

by Lemma 13 and Corollary 4. Since $G_0 = G_0^\#$ and $G_n = G_n^\#$, this means that

$$\deg_k(R - R^\#) < \deg_k G_i \text{ for all } 0 \leqslant i \leqslant n.$$

In other words, the polynomials $R^\#$, $R - R^\#$, and therefore R are all reduced with respect to G. □

5.2. Computing the remainder

Once the quotients Q_0, ..., Q_n are known, we need to compute the remainder $R := P - Q_0 G_0 - \cdots - Q_n G_n$. We do this by rewriting (or *retracting*) the linear combination $Q_0\,G_0 + \cdots + Q_n\,G_n$ into a linear combination $S_0\,G_0 + S_1\,G_1 + S_n\,G_n$ using the following algorithm:

Algorithm 1
Input: the quotients $Q_0, \ldots, Q_n \in \mathbb{K}[X, Y]$ of the dichotomic extended reduction of P by G
Output: $S_0, S_1, S_n \in \mathbb{K}[X, Y]$ with $Q_0 G_0 + \cdots + Q_n G_n = S_0 G_0 + S_1 G_1 + S_n G_n$

For $j = 0, \ldots, n$, set $Q_{0,j} := Q_j$
For $\lambda = 1, \ldots, \lceil \log_2 n \rceil - 1$ do
  For $j = 0, \ldots, n$ do
    If $1 < j < n$ and $\mathrm{val}_2 j \leqslant \lambda$, then set $Q_{\lambda+1,j} := 0$
    Otherwise, set $Q_{\lambda+1,j} := Q_{\lambda,j} + \sum_{i \in J_\lambda \setminus J_{\lambda+1}} Q_{\lambda,i} C_{\lambda,i,j}$
For $j = 0, 1, n$, define $S_j := Q_{\lceil \log_2 n \rceil, j}$, and return S_0, S_1, S_n

LEMMA 15. *Algorithm 1 is correct and runs in time* $O(\mathsf{M}(kn^2) \log n)$.

Proof. By construction, we notice that $Q_{\lambda,j} = 0$ if $1 < j < n$ and $\mathrm{val}_2 j < \lambda$ (that is $j \notin J_\lambda$). Let us now show by induction over λ that

$$Q_{\lambda,0} G_0 + \cdots + Q_{\lambda,n} G_n = Q_0 G_0 + \cdots + Q_n G_n.$$

This is clearly true for $\lambda = 0$. We have

$$\begin{aligned}
\sum_{j \in J_{\lambda+1}} Q_{\lambda+1,j} G_j &= \sum_{j \in J_{\lambda+1}} \left(Q_{\lambda,j} + \sum_{i \in J_\lambda \setminus J_{\lambda+1}} Q_{\lambda,i} C_{\lambda,i,j} \right) G_j \\
&= \sum_{j \in J_{\lambda+1}} Q_{\lambda,j} G_j + \sum_{i \in J_\lambda \setminus J_{\lambda+1}} Q_{\lambda,i} \sum_{j \in J_{\lambda+1}} C_{\lambda,i,j} G_j \\
&= \sum_{j \in J_{\lambda+1}} Q_{\lambda,i} G_i + \sum_{i \in J_\lambda \setminus J_{\lambda+1}} Q_{\lambda,i} G_i \\
&= \sum_{j \in J_\lambda} Q_{\lambda,i} G_i,
\end{aligned}$$

which proves the correctness of Algorithm 1. Again by induction over λ, it is not hard to see that the bound $\deg_k C_{\lambda,i,j} < 4k2^\lambda$ implies

$$\deg_k(Q_{\lambda,i}) \leqslant \max(4k2^\lambda, 2k2^{\mathrm{val}_2(i)}) \text{ for } 1 < i < n. \tag{5}$$

Now, for $i \in J_\lambda \setminus J_{\lambda+1}$ and $j \in J_\lambda$, the product $Q_{\lambda,i} C_{\lambda,i,j}$ is computed in time $O(k 4^\lambda)$, and there are $O(n^2/4^\lambda)$ such products (see the proof of Lemma 9). Using that $\mathsf{M}(d)/d$ is non-decreasing, we conclude that each step can be computed in time $O(\mathsf{M}(k n^2))$. □

Combining our subalgorithms, we obtain our algorithm for extended reduction.

Algorithm 2
Input: A tersely represented vanilla Gröbner basis $G = (G_0, \ldots, G_n)$ and $P \in \mathbb{K}[X, Y]$
Output: An extended reduction $(Q_0, \ldots, Q_n, R)$ of P modulo G

Compute the extended reduction $(Q_0, \ldots, Q_n, R^\#)$ with respect to $G^\#$
Compute $S_0, S_1, S_2 \in \mathbb{K}[X, Y]$ as a function of $Q_0, \ldots, Q_n$ using Algorithm 1
Compute $R := P - S_0 G_0^\# - S_1 G_1^\# - S_n G_n^\# = P - S_0 G_0 - S_1 G_1 - S_n G_n$
Return $(Q_0, \ldots, Q_n, R)$.

THEOREM 16. *Algorithm 2 is correct and runs in time*

$$O(\mathsf{R}(k n^2) \log n + \mathsf{R}(d^2/k)).$$

Proof. Because of Lemma 13, the extended reduction with respect to $G_0^\#, \ldots, G_n^\#$ is computed in time

$$O(\mathsf{R}(k n^2) \log n + \mathsf{R}(d^2/k)).$$

Proposition 14 ensures that the quotients are also valid with respect to $G_0, \ldots, G_n$. The next step is to evaluate the remainder $R := P - Q_0 G_0 - \cdots - Q_n G_n$. The S_i's are computed in time $O(\mathsf{M}(k n^2) \log n)$ using Lemma 15 and we have

$$Q_0 G_0 + \cdots + Q_n G_n = S_0 G_0 + S_1 G_1 + S_n G_n.$$

For $i \in \{0, 1, n\}$, it follows from (5) that $\deg_k(S_i G_i) \leqslant \max(d, 5 k n)$. Consequently, the evaluation of R takes time

$$O(\mathsf{M}(d^2/k) + \mathsf{M}(k n^2)).$$ □

6. APPLICATIONS

6.1. Multiplications in the quotient algebra

Let $G = (G_0, \ldots, G_n)$ be a vanilla Gröbner basis for an ideal $I \subseteq \mathbb{K}[X, Y]$ with respect to $\prec_k$ and assume that we we have precomputed a terse representation for G. Elements in the quotient algebra $\mathbb{A} = \mathbb{K}[X, Y]/I$ can naturally be represented as polynomials in $\mathbb{K}[X, Y]$ that are reduced with respect to G. An immediate application of Theorem 16 is a multiplication algorithm for $\mathbb{A}$ that runs in quasi-linear time.

More precisely, with the notations from Proposition 3, given two polynomials $P, Q \in \mathbb{K}[X, Y]$ that are reduced with respect to G, we have $\deg_k P \leqslant k n$ and $\deg_k Q \leqslant k n$, whence $\deg_k P Q \leqslant 2 k n$ and $|P Q| = O(k n^2) = O(D)$. It follows that $P Q$ can be computed in time $O(\mathsf{M}(D))$, whereas the reduction of $P Q$ with respect to G takes time $O(\mathsf{R}(D) \log D)$. This yields:

THEOREM 17. *For I as above, multiplication in the quotient algebra $\mathbb{A} = \mathbb{K}[X, Y]/I$ can be performed in time*

$$O(\mathsf{R}(D) \log D).$$

6.2. Changing the monomial ordering

Let us now assume that our ideal $I \subseteq \mathbb{K}[X,Y]$ admits a vanilla Gröbner basis $G^{[k]}$ with respect to the ordering $\prec_k$ for all k. We will write $\mathbb{A}^{[k]} = K[X,Y]/I$ for the quotient algebra when representing elements using normal forms with respect to $G^{[k]}$. If $k > D = \dim_{\mathbb{K}} \mathbb{A}$, then we notice that $G^{[k]}$ is also a Gröbner basis with respect to the lexicographical monomial ordering $\prec_\infty$. In order to efficiently convert between $\mathbb{A}^{[k]}$ and $\mathbb{A}^{[\ell]}$ with $k < \ell$, we first consider the case when $\ell \leqslant 2k$:

LEMMA 18. *With the above notations and $k < \ell \leqslant 2\,k$, assume that we have precomputed terse representations for $G^{[k]}$ and $G^{[\ell]}$. Then back and forth conversions between $\mathbb{A}^{[k]}$ and $\mathbb{A}^{[\ell]}$ can be computed in time*

$$O(\mathsf{R}(D)\log D).$$

Proof. Assume that $G^{[k]}$ has $n+1$ elements $G_0^{[k]}$, ..., $G_n^{[k]}$ and $G^{[\ell]}$ has $m+1$ elements $G_0^{[\ell]}, \ldots, G_m^{[\ell]}$. We know from Proposition 3 that $k\,n\,(n-1) < 2\,D \leqslant k\,n\,(n+1)$ and similarly $\ell\,(m-1)\,m < 2\,D \leqslant \ell\,m\,(m+1)$. Now given $P \in \mathbb{K}[X,Y]$ that is reduced with respect to $G^{[k]}$, we have $\deg_k P \leqslant k\,n$, whence $\deg_\ell P \leqslant \ell\,n$ and $(\deg_\ell P)^2/\ell \leqslant \ell\,n^2 \leqslant 2\,k\,n^2 = O(D)$. Theorem 16 therefore implies that normal form of P w.r.t. $G^{[\ell]}$ can be computed in time

$$O(\mathsf{R}(\ell\,m^2)\log m + \mathsf{R}(D))$$

and we conclude using $\ell\,m^2 = O(D)$. The proof for the backward conversion is similar. □

For general $k < \ell$, let $a \leqslant b$ be such that $2^{a-1} < k \leqslant 2^a$ and $2^{b-1} < \ell \leqslant 2^b$. Then we may perform conversions between $\mathbb{A}^{[k]}$ and $\mathbb{A}^{[\ell]}$ using a Gröbner walk

$$\mathbb{A}^{[k]} \leftrightarrow \mathbb{A}^{[2^a]} \leftrightarrow \cdots \leftrightarrow \mathbb{A}^{[2^{b-1}]} \leftrightarrow \mathbb{A}^{[\ell]}.$$

All $G^{[k]}$ coincide for $k > D$, so we can assume that $1 \leqslant k < \ell \leqslant D+1$. Then there are at most $\log D$ conversions as above, so that:

THEOREM 19. *With the above notations and $k < \ell \leqslant D+1$, assume that we have precomputed terse representations for $G^{[k]}, G^{[2^a]}, \ldots, G^{[2^b]}, G^{[\ell]}$. Then back and forth conversions between $\mathbb{A}^{[k]}$ and $\mathbb{A}^{[\ell]}$ can be computed in time*

$$O(\mathsf{R}(D)\log^2 D).$$

7. CONCLUSION AND PERSPECTIVES

As explained in the introduction, we deliberately chose to present our results in the simplest possible setting. As a future work, it would be interesting to generalize our algorithms. The following two extensions should be rather straightforward:

- The consideration of general monomial orderings, starting with $\prec_k$ for $k \in \mathbb{Q}^>$.
- Generalizations to multivariate polynomials in $\mathbb{K}[X_1, \ldots, X_r]$. We expect no essential problems for fixed r. However, the dependence of the complexity on r is likely to be polynomial in $r!$.

Some of the more challenging problems are as follows:

- Is it true that a "sufficiently generic" zero-dimensional ideal I of fixed degree $D = \dim_{\mathbb{K}} \mathbb{K}[X,Y]/I$ necessarily admits a vanilla Gröbner basis?

- Given a vanilla Gröbner basis, what is the actual complexity of computing its terse representation? Our first analysis suggests a bound $\tilde{O}(D^\omega)$, but we suspect that the computation of the retraction coefficients $C_{\lambda,i,j}$ can be accelerated by using the sygyzies that result from reducing the S-polynomials of basis elements to zero.
- Can our results be generalized to the degenerate case of non-vanilla Gröbner bases G?

On the long run, one might also wonder whether some of the new techniques can be used for the efficient computation of Gröbner bases themselves. For the moment, this seems far beyond reach. Nevertheless, a quasi-optimal algorithm does exist for the particular case of an ideal I generated by two generic polynomials $P, Q \in \mathbb{K}[X, Y]$ of total degree δ, when working with respect to the monomial ordering $\prec_1$. We intend to report on the details in a forthcoming paper.

BIBLIOGRAPHY

[1] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In Ravi P. Agarwal, Y. M. Chow, and S. J. Wilson, editors, *Proceedings of the International Conference on Numerical Mathematics*, pages 11–30. Basel, 1988. Birkhäuser Basel.

[2] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, pages 1–24, sep 2014.

[3] Thomas Becker and Volker Weispfenning. *Gröbner bases: a computational approach to commutative algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1993.

[4] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, Universitat Innsbruck, Austria, 1965.

[5] David G Cantor and Erich Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[6] Stéphane Collart, Michael Kalkbrener, and Daniel Mall. Converting bases with the gröbner walk. *Journal of Symbolic Computation*, 24(3-4):465–469, 1997.

[7] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, 1999.

[8] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83. New York, NY, USA, 2002. ACM.

[9] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[10] M. J. Fischer and L. J. Stockmeyer. Fast on-line integer multiplication. *Proc. 5th ACM Symposium on Theory of Computing*, 9:67–72, 1974.

[11] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proc. ISSAC 2014*, pages 296–303. Kobe, Japan, July 23–25 2014.

[12] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

[13] Vladimir P. Gerdt and Yuri A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998.

[14] M. Giusti. Some effectivity problems in polynomial ideal theory. In *Proc. Eurosam '84*, volume 174 of *Lecture Notes in Computer Science*, pages 159–171. Cambridge, 1984. Springer, Berlin.

[15] M. Giusti. A note on the complexity of constructing standard bases. In *Proc. Eurocal '85*, volume 204 of *Lecture Notes in Computer Science*, pages 411–412. Springer-Verlag, 1985.

[16] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.

[17] D. Harvey and J. van der Hoeven. Faster integer and polynomial multiplication using cyclotomic coefficient rings. Technical Report, ArXiv, 2017. http://arxiv.org/abs/1712.03693.

[18] David Harvey, Joris van der Hoeven, and Grégoire Lecerf. Faster polynomial multiplication over finite fields. Technical Report, ArXiv, 2014. http://arxiv.org/abs/1407.3361.

[19] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[20] J. van der Hoeven. On the complexity of polynomial reduction. In I. Kotsireas and E. Martínez-Moro, editors, *Proc. Applications of Computer Algebra 2015*, volume 198 of *Springer Proceedings in Mathematics and Statistics*, pages 447–458. Cham, 2015. Springer.

[21] Joris van der Hoeven. Faster relaxed multiplication. In *Proc. ISSAC '14*, pages 405–412. Kobe, Japan, Jul 2014.

[22] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In J. A. van Hulzen, editor, *Proc. EUROCAL'83*, number 162 in Lect. Notes in Computer Sc., pages 146–156. Springer Berlin Heidelberg, 1983.

[23] Ernst Mayr. Membership in polynomial ideals over Q is exponential space complete. *STACS 89*, pages 400–406, 1989.

[24] L. Robbiano. Term orderings on the polynominal ring. In *European Conference on Computer Algebra (2)*, pages 513–517. 1985.

[25] Fabrice Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, May 1999.

[26] A. Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Infor.*, 7:395–398, 1977.

[27] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.