Computing Generic Fibres of Polynomial Ideals with FGLM and Hensel Lifting

Jérémy Berthomieu, Rafael Mohr

Computing Generic Fibres of Polynomial Ideals with FGLM and Hensel Lifting

Jérémy Berthomieu [*] Rafael Mohr [*†]

February 5, 2024

Abstract

We describe a version of the FGLM algorithm that can be used to compute generic fibers of positive-dimensional polynomial ideals. It combines the FGLM algorithm with a Hensel lifting strategy. We show that this algorithm has a complexity quasi-linear in the number of lifting steps. Some provided experimental data also demonstrates the practical efficacy of our algorithm. Additionally, we sketch a related Hensel lifting method to compute Gröbner bases using so-called tracers.

1 Introduction

Scientific Context *Gröbner bases* lie at the forefront of the algorithmic treatment of polynomial systems and ideals in symbolic computation. They are defined as special generating sets of polynomial ideals which allow to decide the ideal membership problem via a multivariate version of polynomial long division. Given a Gröbner basis for a polynomial ideal, a lot of geometric and algebraic information about the polynomial ideal at hand can be extracted, such as the degree, dimension or Hilbert function. We refer to Becker and Weispfenning, 1993 for a comprehensive treatment of the subject.

Notably, Gröbner bases depend on two parameters: The polynomial ideal which they generate and a *monomial order*, i.e. a certain kind of total order on the set of monomials of the underlying polynomial ring. Then, the geometric and ideal-theoretic information that can be extracted from a Gröbner basis depends on the chosen monomial order. For example, *elimination orders* allow, as the name suggests, to eliminate a chosen subset of variables from the given polynomial ideal (i.e. to project on an affine subspace in a geometric sense).

While Gröbner bases for elimination orders are frequently of interest, it has been observed that all algorithms to compute Gröbner bases based on the famous Buchberger algorithm (Buchberger, 1965), such as F_4 (Faugère, 1999) and F_5 (Faugère, 2002), are substantially more well-behaved when used with non-elimination orders (most notably, the *degree reverse lexicographical* $\prec_{\mathrm{drl}}$ order). This has motivated the design of numerous *change of order* algorithms: The task is to convert a given Gröbner basis w.r.t. one order into a Gröbner basis w.r.t. another order. We mention here the Hilbert-driven algorithm by Traverso, 1996, the Gröbner walk algorithm by Collart, Kalkbrener, and Mall, 1997 and, most notably for this paper, the FGLM algorithm (Faugère, Gianni, Lazard, & Mora, 1993) and its variants (Berthomieu, Neiger, & Safey El Din, 2022; Faugère, Gaudry, Huot, & Renault, 2014; Faugère & Mou, 2017; Neiger & Schost, 2020).

Furthermore, most ideal-theoretic operations in commutative algebra (such as saturation and intersection) can be performed using Gröbner bases by writing down a certain ideal associated to the given polynomial ideal, choosing a certain monomial order and computing a Gröbner basis for this associated ideal. Here, Gröbner basis computation is used as a black box. It has recently been

[*]Sorbonne Université, CNRS, LIP6, Paris, France
[†]RPTU Kaiserslautern-Landau, Germany

observed, partly by the authors of this paper, that it can be (sometimes substantially) more efficient to design *dedicated* Gröbner basis algorithms for specific ideal-theoretic tasks, see Berthomieu, Eder, and Safey El Din, 2023; Eder, Lairez, Mohr, and Safey El Din, 2023b.

Problem Statement & Contributions This paper is concerned with the algorithmic treatment of the following problem: Fix a polynomial ring $R := \mathbb{K}[\mathbf{z}, \mathbf{x}]$ in two finite sets of variables $\mathbf{x}$ and $\mathbf{z}$ over a field $\mathbb{K}$ and an ideal I in R. Assume that the map $\varphi : \mathbb{K}[\mathbf{z}] \to \mathbb{K}[\mathbf{z}, \mathbf{x}]/I$ is injective and has *generically finite fiber*, i.e. assume that the the *generic fiber* $I^{\text{gen}} := I\mathbb{K}(\mathbf{z})[\mathbf{x}]$ of I is zero-dimensional. Given a Gröbner basis of I w.r.t. a monomial order $\prec_1$, we want to compute a Gröbner basis G of I^{gen} w.r.t. another monomial order $\prec_2$. The algorithm we design to solve this problem thus relates to the two research directions previously mentioned: It is a *dedicated* algorithm to perform an ideal-theoretic operation (by computing a representation of the generic fiber of a suitably chosen map) and it performs a change of order (by going from $\prec_1$ to $\prec_2$). Our proposed solution to this problem can be seen as a combination of the previously mentioned FGLM algorithm with classical Hensel lifting techniques. As such, it can also be immediately transported to the setting where a Gröbner basis G of a zero-dimensional ideal over $\mathbb{Q}$ is required: It would extract G out of its images over $\mathbb{Z}/p^k\mathbb{Z}$, where $k \in \mathbb{N}^*$ and p is a well-chosen prime number. We show that, similar to classical Hensel lifting, our algorithm runs in arithmetic complexity quasi-linear in the precision up to which we need to lift when a "quadratic lifting strategy" is chosen, see Corollary 4.7, which implies in particular the following

Theorem 1.1. *Let $f_1, \dots, f_c$ be generic polynomials of respective degrees $d_1, \dots, d_c$ in $\mathbb{K}[x_1, \dots, x_c, z_1, \dots, z_{n-c}]$. Assume that the $\prec_{\text{drl}}$-Gröbner basis of $I = \langle f_1, \dots, f_c \rangle$ is known and that the $\prec_2$-Gröbner basis G of $I\mathbb{K}(z_1, \dots, z_{n-c})[x_1, \dots, x_c]$ has coefficients which are rational functions with degrees at most δ in the numerators and denominators. Then, one can compute G up to precision 2δ using $\widetilde{O}\left(\delta c (d_1 \cdots d_c)^3\right)$ operations in $\mathbb{K}$.*

Note that knowing G up to precision 2δ is enough to recover G by means of Padé approximants, see Lemma 3.2.

Related Work Gröbner bases of generic fibers, as defined in the previous paragraph, are classically computed using *elimination monomial orders*, see Becker and Weispfenning, 1993, Lemma 8.93. Besides that, morally similar to our algorithm, there is a rich body of literature about multi-modular Gröbner basis computations (Arnold, 2003; Ebert, 1983; Pauer, 1992; Traverso, 1989) and Hensel/modular lifting techniques for Gröbner bases (Gräbe, 1993; Schost & St-Pierre, 2023; Winkler, 1988).
Outside of the world of Gröbner bases, there are other data structures for algorithmically manipulating polynomial ideals, or the algebraic sets defined by them, which encode polynomial ideals by their generic fiber associated to a well-chosen projection. We mention in particular *geometric resolutions*, see e.g. (Giusti, Lecerf, & Salvy, 2001; Schost, 2003), and *triangular sets*, see e.g. Hubert, 2003 for a survey.
Our work also relates to specialization results for Gröbner basis, i.e. results on the question whether a Gröbner basis remains a Gröbner basis after specializing some of the variables, see e.g. Becker, 1994; Gianni, 1989.

Outline In Section 2, we give necessary preliminaries both on Gröbner bases and on the needed commutative algebra to state and prove the correctness of our algorithms in Section 3. In Section 4, we transport the complexity statements for the original FGLM algorithm to our setting. Then, in Section 5, we sketch how the concept of *Tracers*, introduced in Traverso, 1989 for the purpose of multi-modular Gröbner basis computations, can be transported to a modular lifting strategy. We discuss how this could profitably be combined with our main algorithm. Finally we give some benchmarks for a `Julia` implementation of our main algorithm in Section 6, comparing it to computing generic fibers using just elimination orders.

2 Preliminaries

2.1 Gröbner Bases

In order to be self-contained, we recall some definitions and basic properties related to Gröbner bases of polynomial ideals.
For a set of variables $\mathbf{x} := \{x_1, \ldots, x_n\}$, we denote by $\mathrm{Mon}(\mathbf{x})$ the set of monomials in $\mathbf{x}$, and for a field $\mathbb{K}$, we let $R := \mathbb{K}[\mathbf{x}]$ be the multivariate polynomial ring in $\mathbf{x}$ over $\mathbb{K}$.

Definition 2.1. A *monomial order* $\prec$ on $\mathbf{x}$ is a total order on $\mathrm{Mon}(\mathbf{x})$ which

1. extends the partial order on $\mathrm{Mon}(\mathbf{x})$ given by divisibility and
2. is compatible with multiplication i.e. we have

$$u \prec v \;\Rightarrow\; wu \prec wv \quad \forall u, v, w \in \mathrm{Mon}(\mathbf{x}).$$

Of importance for us is the *degree reverse lexicographic* order:

Definition 2.2. The *degree reverse lexicographic* $\prec_{\mathrm{drl}}$ order on $\mathrm{Mon}(\mathbf{x})$ is defined as follows for $u, v \in \mathrm{Mon}(\mathbf{x})$: $u \prec_{\mathrm{drl}} v$ iff $\deg u < \deg v$ or $\deg u = \deg v$ and the last nonzero exponent of u/v is positive.

We will also need the notion of a *block order*:

Definition 2.3. Let $\mathbf{x}$ and $\mathbf{z}$ be two finite sets of variables. Write each monomial $u \in \mathrm{Mon}(\mathbf{x} \cup \mathbf{z})$ uniquely as a product $u = u_{\mathbf{x}} u_{\mathbf{z}}$ with $u_{\mathbf{x}} \in \mathrm{Mon}(\mathbf{x})$ and $u_{\mathbf{z}} \in \mathrm{Mon}(\mathbf{z})$. Fix a monomial order $\prec_1$ on $\mathrm{Mon}(\mathbf{x})$ and a monomial order $\prec_2$ on $\mathrm{Mon}(\mathbf{z})$. The corresponding *block order eliminating* $\mathbf{x}$ is defined as follows: $u \prec v$ iff $u_{\mathbf{x}} \prec_1 v_{\mathbf{x}}$ or $u_{\mathbf{x}} = v_{\mathbf{x}}$ and $u_{\mathbf{z}} \prec_2 v_{\mathbf{z}}$ for $u, v \in \mathrm{Mon}(\mathbf{x} \cup \mathbf{z})$.

A monomial order on $\mathbf{x}$ yields a notion of *leading monomial* in R:

Definition 2.4. Let $\prec$ be a monomial order on $\mathrm{Mon}(\mathbf{x})$. For a nonzero element $f \in R$ the *leading monomial* of f w.r.t. $\prec$, denoted $\mathrm{lm}_{\prec}(f)$, is the $\prec$-largest monomial in the support of f. For a finite set F in R we define $\mathrm{lm}_{\prec}(F) := \{\mathrm{lm}_{\prec}(f) \mid f \in F\}$. For an ideal I in R we define the *leading monomial ideal* of I as $\mathrm{lm}_{\prec}(I) := \langle \mathrm{lm}_{\prec}(f) \mid f \in I \rangle$.

Fixing a monomial order gives *normal forms* for images of elements in quotient rings of R:

Definition 2.5. Let I be an ideal in R and let $\prec$ be a monomial order $\mathrm{Mon}(\mathbf{x})$.

1. The set $S_{I,\prec} := \{u \in \mathrm{Mon}(\mathbf{x}) \mid u \notin \mathrm{lm}_{\prec}(I)\}$ is called the *staircase* of I w.r.t. $\prec$. It naturally forms a $\mathbb{K}$-vector space basis of R/I.
2. The image of every element $f \in R$ in R/I can be uniquely written as a $\mathbb{K}$-linear combination of elements in $S_{I,\prec}$. This linear combination of elements in $S_{I,\prec}$ is called the *normal form* of f w.r.t. I and $\prec$. The corresponding vector of coefficients of this linear combination, with the elements in $S_{I,\prec}$ ordered by $\prec$, will be denoted $c_{I,\prec}(f)$.

We finally define the notion of Gröbner bases.

Definition 2.6. A *Gröbner basis* of an ideal $I \subset R$ w.r.t. a monomial order $\prec$ is a finite set $G \subset I$ such that $\langle \mathrm{lm}_{\prec}(G) \rangle = \mathrm{lm}_{\prec}(I)$.

A Gröbner basis G of an ideal $I \subset R$ w.r.t. a monomial order $\prec$ enables the computation of normal forms w.r.t. I and $\prec$ via a straightforward multivariate generalization of polynomial long division, see e.g. Becker and Weispfenning, 1993, Table 5.1. This, in particular, yields an ideal membership test for I. Indeed, an element $f \in R$ is contained in I if and only if its normal form w.r.t. I and $\prec$ is zero.

2.2 Points of Good Specialization

We start by fixing some notation. For a ring R and an element $f \in R$ we denote by $R[f^{-1}]$ the localization of R at the multiplicatively closed set $\left\{ f^k \;\middle|\; k \in \mathbb{N} \right\}$. For a prime ideal $\mathfrak{p} \subset R$ we denote by $R_{\mathfrak{p}}$ the localization of R at the multiplicatively closed set $R \setminus \mathfrak{p}$.
We further fix a polynomial ring $\mathbb{K}[\mathbf{z}, \mathbf{x}]$ in two finite sets of variables $\mathbf{z}$ and $\mathbf{x}$. Let $I \subseteq \mathbb{K}[\mathbf{z}, \mathbf{x}]$ be an ideal. Suppose that the map

$$\mathbb{K}[\mathbf{z}] \to \mathbb{K}[\mathbf{z}, \mathbf{x}]/I$$

is injective and has *generically finite fiber*, i.e. we assume that $I^{\text{gen}} := I\mathbb{K}(\mathbf{z})[\mathbf{x}] \neq \mathbb{K}(\mathbf{z})[\mathbf{x}]$ is a zero-dimensional ideal.

Definition 2.7. In this setting we call I^{gen} the generic fiber of I.

Let G be the reduced Gröbner basis of I^{gen} w.r.t. a monomial order $\prec$. Further, we denote for a monomial $u \in \text{Mon}(\mathbf{z})$

$$\mathfrak{m}_u := \langle v \in \text{Mon}(\mathbf{z}) \mid v \succ_{\text{drl}} u \rangle \text{ and } I_u := I + \mathfrak{m}_u,$$

$\mathfrak{m} = \langle \mathbf{z} \rangle$, as well as $\text{next}(u) = \min \{ v \in \text{Mon}(\mathbf{z}) \mid v \succ_{\text{drl}} u \}$.
Suppose that $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$. Then, each coefficient of a given $g \in G$ has a well-defined image in $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_u$ for all $u \in \text{Mon}(\mathbf{z})$. More concretely, write

$$g = \sum_{w \in \text{Mon}(\mathbf{x})} \frac{p_w}{q_w} w$$

with $q_w(0) \neq 0$ for all $w \in \text{Mon}(\mathbf{x})$ whenever $p_w \neq 0$. Then, each p_w/q_w can be written as a formal power series

$$\frac{p_w}{q_w} = \sum_{v \in \text{Mon}(\mathbf{z})} r_{w,v} v$$

and for a monomial $u \in \text{Mon}(\mathbf{z})$ we define

$$g_u = g \bmod \mathfrak{m}_u = \sum_{w \in \text{Mon}(\mathbf{x})} \sum_{\substack{v \in \text{Mon}(\mathbf{z}) \\ v \preceq_{\text{drl}} u}} r_{w,v} v w,$$

and the set $G_u := \{ g_u \mid g \in G \}$. Our algorithms will work under the assumption that all elements in G have such a well-defined image modulo $\mathfrak{m}_u$ and that, given this image of G modulo $\mathfrak{m}_u$, we can lift it uniquely to its image modulo $\mathfrak{m}_{\text{next}(u)}$. In this section we show that both required assumptions are generically satisfied, more precisely that they are satisfied if the origin in the $\mathbf{z}$-space lies outside of a certain hypersurface. Our way of showing this uses Grothendieck's Generic Freeness Lemma (Eisenbud, 1995, Theorem 14.4). We restate it here in a weaker form adapted to our situation:

Theorem 2.8 (Generic Freeness)**.** *There exists an element* $0 \neq f \in \mathbb{K}[\mathbf{z}]$ *such that* $(\mathbb{K}[\mathbf{z}, \mathbf{x}]/I)[f^{-1}]$ *is a free* $\mathbb{K}[\mathbf{z}][f^{-1}]$*-module.*

Both of the requirements of our algorithms are now enclosed by the following definition:

Definition 2.9. Let f be as in Theorem 2.8. If $f \notin \mathfrak{m}$, then $\mathfrak{m}$ is called a *point of good specialization*.

Remark 2.1. If $\mathfrak{m}$ is not a point of good specialization, or if it is not known a priori that it is, then it can be ensured with probability 1 after a random change of coordinates in the variables $\mathbf{z} = (z_1, \ldots, z_r)$ (if $\mathbb{K}$ is infinite) or with high probability (if $\mathbb{K}$ is finite of sufficiently large size) by Theorem 2.8. In particular, if the change of coordinates $z_i \leftarrow z_i - a_i$, for $1 \leq i \leq r$, suits, then $\mathfrak{m}' := \langle z_1 - a_1, \ldots, z_r - a_r \rangle$ was a point of good specialization. Equivalently, instead of performing this change of coordinates, one can use the algorithm with $\mathfrak{m}'$-adic expansions.

Theorem 2.10. *Suppose that $\mathfrak{m}$ is a point of good specialization. Then,*

1. *For each $u \in \mathrm{Mon}(\mathbf{z})$ and $z_i \in \mathbf{z}$ multiplication by z_i induces an isomorphism of $\mathbb{K}$-vector spaces:*
$$u\mathbb{K}[\mathbf{z},\mathbf{x}]/I_u \to z_i u\mathbb{K}[\mathbf{z},\mathbf{x}]/I_{z_i u}.$$
2. *We have $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$, in particular the sets G_u, $u \in \mathrm{Mon}(\mathbf{z})$, are well-defined.*
3. *Let $\prec_1$ be the block order eliminating $\mathbf{x}$ with $\prec_1 = \prec_{\mathrm{drl}}$ on $\mathrm{Mon}(\mathbf{z})$. Let M_u be the (unique) minimal generating set of $\mathfrak{m}_u$. Then, the reduced $\prec_1$-Gröbner basis of I_u is precisely $G_u \cup M_u$.*

Proof. We introduce some notation: Write $A \coloneqq \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}$, $K = \mathbb{K}(\mathbf{z})$ for the field of fractions of A and $F \coloneqq A[\mathbf{x}]/I$. Note that $\mathfrak{m}$ being a point of good specialization implies that F is a free R-module, necessarily of finite nonzero rank, since I^{gen} is zero-dimensional.
Proof of (1): It is first easy to check that now multiplication by z_i induces a surjective, well-defined map of finite-dimensional $\mathbb{K}$-vector spaces
$$u\mathbb{K}[\mathbf{z},\mathbf{x}]/I_u \to z_i u\mathbb{K}[\mathbf{z},\mathbf{x}]/I_{z_i u}.$$
Note that the structure of $V_u \coloneqq u\mathbb{K}[\mathbf{z},\mathbf{x}]/I_u$ as a vector space is induced by the canonical A-module structure of F, because $\mathfrak{m}V_u = 0$ and therefore $(V_u)_{\mathfrak{m}} = V_u$. Hence, we have
$$V_u \cong (uA/\mathfrak{m}_u)^r \cong \mathbb{K}^r.$$
Therefore multiplication by z_i induces an epimorphism between vector spaces of the same dimension, so it must be an isomorphism.
Proof of (2): Let $u \in L \coloneqq \mathrm{lm}_{\prec}(I+\mathfrak{m}) \cap \mathrm{Mon}(\mathbf{x})$ and let $S \coloneqq S_{I+\mathfrak{m},\prec}$. Now, modulo $I+\mathfrak{m}$, we have a relation of the form
$$u = \sum_{s\in S} \alpha_s s,\ \alpha_s \in \mathbb{K}.$$
By Nakayama's lemma, see Eisenbud, 1995, Corollary 4.8, this gives a relation
$$u = \sum_{s\in S} r_s s,\ r_s \in A \tag{1}$$
in F and hence also in $F \otimes_A K = \mathbb{K}(\mathbf{z})[\mathbf{x}]/I^{\mathrm{gen}}$. This implies that $L \subset \mathrm{lm}_{\prec_2}(I^{\mathrm{gen}})$. But since F is free, we have
$$\dim_K \mathbb{K}(\mathbf{z})[\mathbf{x}]/I^{\mathrm{gen}} = \dim_{\mathbb{K}} \mathbb{K}[\mathbf{z},\mathbf{x}]/(I+\mathfrak{m}),$$
which implies $L = \mathrm{lm}_{\prec_2}(I^{\mathrm{gen}})$. This shows that G is given by all relations of the form (1) and hence lies in $A[\mathbf{x}]$ as claimed.
Proof of (3): Let $S \coloneqq S_{I^{\mathrm{gen}},\prec}$. It suffices to show that
$$S_{I_u,\prec_1} = S_u \coloneqq \bigcup_{v \preceq_{\mathrm{drl}} u} vS.$$
Note that the set S_u certainly generates $\mathbb{K}[\mathbf{z},\mathbf{x}]/(I+\mathfrak{m}_u)$ as a $\mathbb{K}$-vector space. As $\mathbb{K}[\mathbf{z},\mathbf{x}]/(I+\mathfrak{m}_u) \cong F/\mathfrak{m}_u$, and since F is free, a $\mathbb{K}$-dimension count shows that the set S_u is $\mathbb{K}$-linearly independent. Now, let $s \in S$ be $\prec$-minimal such that there exists some $v \in \mathrm{Mon}(\mathbf{z})$ with $vs \in \mathrm{lm}_{\prec_1}(I_u)$. The $\prec_1$-normal form of vs w.r.t. I_u has support in $\bigcup_{v \preceq_{\mathrm{drl}} u} \{vt \mid t \prec s\}$, contradicting the linear independence of S_u and finishing the proof. □

We will want to perform finite-dimensional linear algebra akin to the FGLM algorithm in certain staircases of the ideals I_u. This will rely on the fact that I_1 is zero-dimensional.

Corollary 2.11. *Suppose $\mathfrak{m}$ is a point of good specialization. Then the ideal I_1 is zero-dimensional. Consequently, for each $u \in \mathrm{Mon}(\mathbf{z})$, the ideal I_u is zero-dimensional.*

Proof. This follows immediately from the fact that G specializes to a Gröbner basis of $J \coloneqq I\mathbb{K}[\mathbf{z},\mathbf{x}]/\mathfrak{m}$. Indeed, this implies that that $S_{J,\prec_2} = S_{I^{\mathrm{gen}},\prec_2}$ is finite and so $\mathbb{K}[\mathbf{x}]/J$ is a finite-dimensional $\mathbb{K}$-vector space. □

3 The Main Algorithm

In this section we give the main algorithm of this paper.
We reuse the notation from the last section, in particular let again $I_u := I + \mathfrak{m}_u$ for a monomial $u \in \mathrm{Mon}(\mathbf{z})$. Let G be the reduced Gröbner basis of the zero-dimensional ideal I^{gen} w.r.t. a monomial order $\prec_2$. Suppose that $\mathfrak{m}$ is a point of good specialization and that we can compute, with some black box, the reduced $\prec_1$-Gröbner basis H_u of I_u for any $u \in \mathrm{Mon}(\mathbf{z})$ where $\prec_1$ is another monomial order. Our goal is to compute the set G.

Remark 3.1. Note that we have so far required that the partition of the variables of $\mathbb{K}[\mathbf{z}, \mathbf{x}]$ is given. It can be computationally determined: From any Gröbner basis of I we can determine $\mathbf{z}$ as a *maximally independent set* of I w.r.t $\prec_1$ and let $\mathbf{x}$ be the set of remaining variables, see Becker and Weispfenning, 1993, Definition 9.22. Then, as in the last section, the map $\mathbb{K}[\mathbf{z}] \to \mathbb{K}[\mathbf{z}, \mathbf{x}]/I$ is injective with generically finite fiber, see Becker and Weispfenning, 1993, Corollary 9.28.

Recall that $\mathfrak{m}_1 = \mathfrak{m}$. We start by computing the $\prec_1$-Gröbner basis H_1 of $I_1 = I + \mathfrak{m}$. Then, we run the FGLM algorithm (Faugère, Gianni, Lazard, & Mora, 1993) with H_1 to obtain the reduced $\prec_2$-Gröbner basis of I_1. A given $g \in G$ may be written as

$$g = \sum_{w \in \mathrm{Mon}(\mathbf{x})} \frac{p_w}{q_w} w,$$

with $p_w/q_w \in \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}$. By Theorem 2.10 the reduced $\prec_2$-Gröbner basis of I_1, considered as an ideal in $\mathbb{K}[\mathbf{x}]$, will now consist of

$$G_1 := \left\{ \sum_{w \in \mathrm{Mon}(\mathbf{x})} \frac{p_w(0)}{q_w(0)} w = g \bmod \mathfrak{m} \;\middle|\; g := \sum_{w \in \mathrm{Mon}(\mathbf{x})} \frac{p_w}{q_w} w \in G \right\}.$$

Note again that each coefficient of a given $g \in G$ has a well-defined image in $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_u$ for all $u \in \mathrm{Mon}(\mathbf{z})$. For a monomial $u \in \mathrm{Mon}(\mathbf{z})$, let $v := \mathrm{next}(u)$. Starting with $u = 1$ and g_1, we will lift g_u to g_v by performing linear algebra in the finite-dimensional $\mathbb{K}$-vector space $v\mathbb{K}[\mathbf{z}, \mathbf{x}]/I_v$, using the $\prec_1$-Gröbner basis H_v. This will rely on the second statement of Theorem 2.10.

Remark 3.2. In this section we treat the computation of the required Gröbner bases H_u, $u \in \mathrm{Mon}(\mathbf{z})$ as a black box. We recall in Section 4 that these sets may be obtained free of arithmetic operations from an $\prec_1$-Gröbner basis of I when $\prec_1 = \prec_{\mathrm{drl}}$ and I is a generic complete intersection. In Section 5, we sketch a method to compute them efficiently when $\prec_1$ is a suitable block order, without any extra assumption on I.

This lifting step is now given by Algorithm 1.

Algorithm 1 The Lifting Algorithm

Input A monomial $u \in \mathrm{Mon}(\mathbf{z})$, $g_u \in G_u$, $v := \mathrm{next}(u)$, the reduced Gröbner basis H_v of I_v w.r.t. $\prec_1$, the set $S_{\prec_2}$.

Output The corresponding element $g_v \in G_v$.

```
1  function lift(g_u, H_v, S_{≺2})
2      c ← c_{I_v,≺1}(g_u)    [computed via H_v]
3      if c = 0 then return g_u
4      compute c = Σ_{v∈S_{≺2}} α_v c_{I_v,≺1}(uv)    [computed via H_v]
5      return g_u − Σ_{v∈S_{≺2}} α_v uv
```

Theorem 3.1. *Algorithm 1 terminates and is correct in that it satisfies its output specification.*

Proof. We use the notation from the pseudocode of the algorithm.

The termination of the algorithm is clear.
For the correctness of the algorithm, note that the vectors $c_{I_v,\prec_1}(uv)$ in **??** 4 are linearly independent thanks to the second item of Theorem 2.10. Thus, there exists at most one choice of coefficients $\alpha_v, v \in S_{\prec_2}$ such that $c = \sum_{v \in S_{\prec_2}} \alpha_v c_{I_v,\prec_1}(uv)$. Furthermore, the element $g \in G$ corresponding to g_u provides such a choice of coefficients, implying that there exists at least one solution to this linear system. This proves the correctness. □

Denote again $G_u := \{g_u \mid g \in G\}$. The above algorithm is only able to compute the set G_u for a monomial $u \in \mathrm{Mon}(\mathbf{z})$, i.e. it "approximates" the set G up to order u. A natural question is then how to extract the actual set G out of G_u. For this, we may use the classical technique of Padé approximants. Having computed the set G_u, we have computed the element g_u as

$$g_u = \sum_{w \in \mathrm{Mon}(\mathbf{x})} \sum_{v \preceq_{\mathrm{drl}} u} r_{w,v} vw.$$

Now we have for the coefficient p_w/q_w of w in g

$$p_w - q_w \sum_{v \preceq_{\mathrm{drl}} u} r_{w,v} v = 0 \mod \mathfrak{m}_u, \tag{2}$$

which determines a set of linear equations in the unknown coefficients of p_w and q_w. Let $d := \deg u$. Suppose that $\deg \mathrm{next}(u) = d + 1$, so that $\mathfrak{m}_u = \mathfrak{m}^{d+1}$. Fix d_1 and d_2 with $d_1 + d_2 = d$ and let n be the cardinality of the set $\mathbf{z}$. If we impose that $\deg p_w \leq d_1$ and $\deg q_w \leq d_2$ then the linear system (2) has a finite set of unknowns and equations. Let us say that any solution to this linear system of equations is a *Padé approximant of order d of* $\lambda_w := \sum_{\deg v < d+1} r_{w,v} v$. If d_1 and d_2 are large enough then any Padé approximant of order d of λ_w is equal to p_w/q_w, see e.g. Guillaume and Huard, 2000, Proposition 2.1:

Lemma 3.2. *Let p/q be a Padé approximant of order $d = d_1 + d_2$ of λ_w. If $d_1 \geq \deg p_w$ and $d_2 \geq \deg q_w$ then $p/q = p_w/q_w$.*

By solving this linear system we obtain an algorithm $pade(g_u, d_1, d_2)$ which computes a candidate $g_{\mathrm{cand}} \in \mathbb{K}(\mathbf{z})[\mathbf{x}]$ whose coefficients are Padé approximants of the coefficients of g_u of order $d_1 + d_2$ regarded as a polynomial in the variables $\mathbf{x}$. Let us say that g_u has *stable Padé approximation* if for $v := \mathrm{next}(u)$ we have

$$g_{\mathrm{cand}} = g_v \mod \mathfrak{m}_v.$$

Based on this, we now obtain Algorithm 2 for computing the set G probabilistically. We state this algorithm in an informal way. In **??** 7 by "lifting G_{lift} to degree d" we mean that we compute the set G_u where u is the $\prec_{\mathrm{drl}}$-maximal monomial of degree d.
Clearly, by Theorem 3.1 and Lemma 3.2, this algorithm returns the correct result if the computed Padé approximants are of sufficiently large degree.

Remark 3.3. Note that Algorithm 1 works also if we replace v by any monomial larger than u: In this case we just have to write c as a linear combination of all the vectors $c_{I_v,\prec_1}(uv')$ where $u \prec_{\mathrm{drl}} v' \preceq_{\mathrm{drl}} v$.

4 Complexity Estimates

In this section, we analyze the arithmetic complexity of a version of our algorithm more akin to the original FGLM algorithm as presented in Faugère, Gianni, Lazard, and Mora, 1993. More precisely, suppose that $\prec_1$ is a block order eliminating $\mathbf{x}$. We will reuse the notation from the last section. Here, we analyze the number of arithmetic operations in $\mathbb{K}$ required to obtain the desired $\prec_2$-Gröbner basis G using the same strategy as in Algorithm 2, but with a more optimized lifting step.
Our cost analysis will require measuring the cost of performing certain linear algebra operations on structured matrices. The matrices that will appear in the analysis are *block-Toeplitz*:

Algorithm 2 Computing the generic fiber

Input A generating set F of I, a monomial order $\prec_1$, a monomial order $\prec_2$.

Output A guess for the set G.

```
function genfglm(F, ≺1, ≺2)
    H ← reduced ≺1-Gröbner basis of I1    [using F]
    Glift, S≺2 ← fglm(H, ≺2)
    Gresult ← ∅
    d ← 2
    while Glift ≠ ∅
        Glift ← lift Glift to degree d using Algorithm 1
        Compute Padé approx. for all elements in Glift
        Lift Glift one monomial higher
        add to Gresult all elements with stable Padé approx.
        remove the corresponding elements from Glift
        d ← 2d
    return Gresult
```

Definition 4.1. Let $k, D \in \mathbb{N}$. A block matrix $M = (M_{pq})_{0 \leq p,q < k} \in \mathbb{K}^{kD \times kD}$ where each M_{pq} is in $\mathbb{K}^{D \times D}$ is called *block-Toeplitz of type* (k, D) if $M_{pq} = M_{p'q'}$ whenever $p - q = p' - q'$. We say that M is *Toeplitz* when $D = 1$.

If M is block-Toeplitz of type (k, D) then we will need the arithmetic complexity of computing a matrix vector product Mv, $v \in \mathbb{K}^{kD}$, and of inverting M. For this we need the concept of *displacement rank* of a matrix, see e.g. Bostan, Jeannerod, Mouilleron, and Schost, 2017:

Definition 4.2. Let $Z \in \mathbb{K}^{n \times n}$ be the matrix defined by

$$Z \coloneqq (\delta_{i-1,j})_{1 \leq i,j \leq n},$$

where $\delta_{i-1,j}$ is the Kronecker delta and let Z^{T} be the transpose of Z. The *displacement rank* of a matrix $M \in \mathbb{K}^{n \times n}$ is

$$\alpha(M) \coloneqq \operatorname{rank}(M - ZMZ^{\mathrm{T}}).$$

Note that the displacement rank of a Toeplitz matrix is upper bounded by 2. The concept of displacement rank can be used as a general method to utilize "Toeplitz-like" structures in algorithmic linear algebra. In this vain, we have

Proposition 4.3. *Let M be block-Toeplitz of type (k, D) and let $v \in \mathbb{K}^{kD}$. Then*

1. *Mv can be computed in $\widetilde{O}\left(kD^2\right)$ arithmetic operations in $\mathbb{K}$;*
2. *M can be inverted in $\widetilde{O}\left(kD^{\omega}\right)$.*

Proof. For any matrix $M \in \mathbb{K}^{n \times n}$, according to Bostan, Jeannerod, Mouilleron, and Schost, 2017, a matrix-vector product Mv can be computed in time $\widetilde{O}\left(\alpha(M)n\right)$ and M can be inverted in time $\widetilde{O}\left(\alpha(M)^{\omega-1}n\right)$. Using a series of rows and column swaps, more precisely sending row $pD + i$ to $ik + p$ (resp. column $qD + j$ to $jk + q$), we may transform a block-Toeplitz matrix M of type (k, D) into a matrix $N = (N_{ij})_{0 \leq i,j < D} \in \mathbb{K}^{kD \times kD}$ where each N_{ij} lies in $\mathbb{K}^{k \times k}$ and is Toeplitz. Now, $N - ZNZ^{\mathrm{T}}$ has D dense columns and $(k-1)D$ columns with potentially nonzero coefficients in positions iD for all i. Only D of these latter columns can be linearly independent so that $\alpha(M) \leq 2D$, proving both claims. □

Our cost analysis follows closely the one of the original FGLM algorithm. To this end, we give the following definition:

Definition 4.4 (Multiplication Tensor)**.** Let I be a zero-dimensional ideal in a polynomial ring $\mathbb{K}[\mathbf{x}]$ and let $\prec$ be a monomial order. Let $S := S_{I,\prec}$. The multiplication tensor of I w.r.t. $\prec$ is defined as the 3-tensor

$$M(I, \prec) = (c_{I,\prec}(x_i u))_{x_i \in \mathbf{x}, u \in S},$$

where the vectors of coefficients are in the basis S.

It turns out that computing these multiplication tensors dominates the cost of the original FGLM algorithm and similarly it dominates the cost of our algorithm. To simplify the notation, we assume from here on out that the set $\mathbf{z} = \{z\}$ consists of a single variable. It will be clear from the proofs that our theorem translates accordingly to the more general setting where $\mathbf{z}$ consists of several variables.
We denote by D the degree of I^{gen}, i.e. the $\mathbb{K}(z)$-dimension of $\mathbb{K}(z)[\mathbf{x}]/I^{\text{gen}}$. Note that this degree is upper-bounded by that of I. We also denote $I_k := I_{z^{k-1}}$ and similarly H_k and G_k. In our assumed setting, we obtain the following statement for computing multiplication tensors:

Theorem 4.5. *Let $k \in \mathbb{N}$. Let c be the cardinality of $\mathbf{x}$. Suppose that we are given the set H_{2k} and the multiplication tensor of I_k w.r.t. $\prec_1$. Then the multiplication tensor of I_{2k} w.r.t. $\prec_1$ is computed in arithmetic complexity $\widetilde{O}\left(kcD^3\right)$.*

Proof. Let $S := S_{I_1,\prec_1}$. Then, by the third item of Theorem 2.10, since $\prec_1$ eliminates $\mathbf{x}$, for any $\ell \in \mathbb{N}$,

$$S_{I_\ell,\prec_1} = \bigcup_{i=0}^{\ell-1} z^i S.$$

Let us now describe the structure of the multiplication matrices of I_{2k}, i.e. the matrices

$$M(I_{2k}, \prec_1)_y := (c_{I_{2k},\prec_1}(yu))_{u \in S_{I_{2k},\prec_1}}, \quad \text{for } y \in \mathbf{x} \cup \mathbf{z}.$$

The matrix $(c_{I_{2k},\prec_1}(zu))_{u \in S_{I_{2k},\prec_1}}$ of the multiplication by z is

$$\begin{array}{c} \\ S \\ zS \\ \vdots \\ z^{2k-1}S \end{array} \begin{array}{c} \begin{array}{cccc} S & zS & \dots & z^{2k-1}S \end{array} \\ \begin{bmatrix} & & & \\ \mathrm{Id} & & & \\ & \ddots & & \\ & & \mathrm{Id} & \end{bmatrix} \end{array}$$

and so is extracted without any arithmetic operations. Further, denote $S_0 := \bigcup_{i=0}^{k-1} z^i S$ and $S_1 := \bigcup_{i=k}^{2k-1} z^i S = z^k S_0$. Now, for $x_i \in \mathbf{x}$ the multiplication matrix M_{x_i} by x_i is determined by two matrices $M_{x_i,0}, M_{x_i,1} \in \mathbb{K}^{D \times D}$ as follows

$$M_{x_i} = \begin{array}{c} \\ S_0 \\ S_1 \end{array} \begin{array}{c} \begin{array}{cc} S_0 & S_1 \end{array} \\ \begin{bmatrix} M_{x_i,0} & \\ M_{x_i,1} & M_{x_i,0} \end{bmatrix} \end{array},$$

where each $M_{x_i,i}$ is easily seen to be block-Toeplitz of type (k, D) and $M_{x_i,0}$ is known as part of the $\prec_1$-multiplication tensor of I_k. Thanks to the block-Toeplitz structure, it now suffices to compute the columns of $M_{x_i,1}$ coming from the normal forms of the set $\mathbf{x}S$, which is of cardinality at most cD. Now, we proceed as follows: Sort the set $\mathbf{x}S$ by the monomial order $\prec_1$. Choose $u \in \mathbf{x}S$ and suppose that the normal forms of all elements less than u in $\mathbf{x}S$ are known. Two easy cases can arise:

1. $u \in S$, in which case the normal form of u is computed without any arithmetic operations;
2. $u \in \mathrm{lm}(H_{2k})$, in which case the normal form of u is computed without any arithmetic operations, it is just given by the tail of the corresponding element in H_{2k}.

Lastly, it can happen that $u \in \mathrm{lm}(I_{2k})$ but $u \notin \mathrm{lm}(H_{2k})$. In this case there exists $v \in \mathbf{x}S$ and $x_j \in \mathbf{x}$ with $u = x_j v$. By assumption the normal form of v is known and so is the normal form of each element $x_j b$ with $b \in S$ and $b \prec_1 v$. Since M_{x_j} has the same structure as M_{x_i}, we can now compute the required column of $M_{x_i,1}$ as the sum of two matrix-vector products where each of the two matrices is block-Toeplitz of type (k, D). This is done in time $\widetilde{O}\left(kD^2\right)$ thanks to Proposition 4.3, concluding the proof, since $\mathbf{x}S$ has cardinality at most cD. □

Now, we can estimate the complexity of lifting the set G_k:

Corollary 4.6. *Let $k \in \mathbb{N}$. Let c be the cardinality of $\mathbf{x}$. Suppose that we are given the set H_{2k}, the multiplication tensor of I_k w.r.t. $\prec_1$, the $\prec_1$-normal forms of the $\prec_2$-staircase of I_1 w.r.t. I_k and the $\prec_1$-normal forms of the minimal $\prec_2$-leading monomials of I_0 w.r.t. I_k. Then G_{2k} is computed in arithmetic complexity $\widetilde{O}\left(kcD^3\right)$.*

Proof. By Theorem 4.5, the $\prec_1$-multiplication tensor of I_{2k} can be computed in arithmetic complexity $O\left(kcD^3\right)$. Having computed this tensor, we proceed as follows: let S be the $\prec_1$-staircase of $I_1 = I + \langle z \rangle$, T be the $\prec_2$-staircase of I_1 and L be the set of minimal $\prec_2$-leading terms of I_1, with z removed. Denoting S_0 and S_1 as in the proof of the preceding theorem, and similarly T_0 and T_1, now we first compute the $\prec_1$-normal forms of each element in $T \cup L$ w.r.t. I_{2k}, this will yield a tableau of the form

$$C := \begin{array}{c} \\ S_0 \\ S_1 \end{array} \begin{array}{c} \begin{array}{ccc} T_0 & T_1 & L \end{array} \\ \begin{bmatrix} C_0 & & D_0 \\ C_1 & C_0 & D_1 \end{bmatrix} \end{array}.$$

Note that by assumption the matrix C_0 is already known by the $\prec_1$-normal forms of T w.r.t. I_1 and the matrix D_0 is given by the $\prec_1$-normal forms of L w.r.t. I_1. Note also that C_0 and C_1 are again block-Toeplitz of type (k, D).
The required matrices C_1 and D_1 can be computed by using the $\prec_1$-multiplication tensor of I_{2k} , enumerating the monomials in $\mathrm{Mon}(\mathbf{x})$ in order of $\prec_2$ and computing their normal forms via matrix-vector multiplications similar to the proof of the preceding theorem. Combining this with the block-Toeplitz structure of the multiplication matrices of I_{2k} w.r.t. $\prec_1$, this can again be done in time $\widetilde{O}\left(kcD^3\right)$. Finally, to compute the set G_{2k}, we have to write each column in C corresponding to an element in L as a $\mathbb{K}$-linear combination of the columns corresponding to $T_0 \cup T_1$, i.e. by solving the linear system

$$\begin{bmatrix} C_0 & \\ C_1 & C_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \end{bmatrix} = \begin{bmatrix} L_0 \\ L_1 \end{bmatrix},$$

where X_0 is known via G_k. Hence this requires

- inverting the submatrix C_0 which, by Proposition 4.3, is done in time $\widetilde{O}\left(kD^\omega\right)$;
- computing the product $C_0^{-1} D_2$.

Note that C_0^{-1} has displacement rank bounded above by $2D + 2$, see Bostan, Chyzak, Giusti, Lebreton, Lecerf, Salvy, and Schost, 2017, Proposition 10.10 and Theorem 10.11, and that the cardinality of L is upper bounded by cD. Thus, for the second step above, we have to compute at most cD matrix-vector products of the form $C_0^{-1} v$. Again, thanks to Proposition 4.3, this is done in time $\widetilde{O}\left(kcD^3\right)$. This finally yields the desired complexity. □

The following corollary gives the complexity of computing successively the sets G_{2^i} from H_{2^i} until i is large enough to recover G, like in Algorithm 2.

Corollary 4.7. *Let $\delta - 1$ be the maximum degree of all numerators and denominators of all coefficients of G. Further, let ℓ be minimal such that $2^\ell \geq 2\delta$.*
Given H_{2^ℓ}, computing successively the sets G_{2^i}, for $i = 1, \ldots, \ell$, can be done in arithmetic complexity $\widetilde{O}\left(2^\ell cD^3\right) = \widetilde{O}\left(\delta cD^3\right)$.

Proof. Note that the sets H_{2^i}, for $i = 1, \ldots, \ell$, are obtained from H_{2^ℓ} free of arithmetic operations. By Corollary 4.6, the computation of G_{2^i} requires $\widetilde{O}\left(2^i C D^3\right)$ operations. Summing these complexities for i from 1 to ℓ yields the desired complexity. □

Note that going up to degree 2^ℓ suffices to recover the coefficients of G by Padé approximation thanks to Lemma 3.2.

Remark 4.1. In a follow-up paper, we plan to study the complexity of our algorithm using variants of FGLM, such as Berthomieu, Neiger, and Safey El Din, 2022; Faugère, Gaudry, Huot, and Renault, 2014; Faugère and Mou, 2017; Neiger and Schost, 2020.

We close this section by pointing out a well-known case in which $\prec_1$ is the $\prec_{\text{drl}}$ order, the required Gröbner bases H_u of $I + \mathfrak{m}_u$ are extracted without any arithmetic operations of the $\prec_{\text{drl}}$-Gröbner basis H of I and the $\prec_{\text{drl}}$-staircase of $I + \mathfrak{m}_u$ behaves the same as in the above case when $\prec_1$ is a block order. We start with

Definition 4.8. Let y be an extra variable and let $I^{\text{hom}} \subset \mathbb{K}[\mathbf{z}, \mathbf{x}, y]$ be the homogenization of I w.r.t y. Suppose that I^{hom} is Cohen-Macaulay. We say that I is in *projective generic position* if $\{y\} \cup \mathbf{z}$ is a maximal homogeneous regular sequence in $\mathbb{K}[\mathbf{z}, \mathbf{x}, y]/I^{\text{hom}}$.

Supposing that I^{hom} is Cohen-Macaulay we now have the following statement, see e.g. Lejeune-Jalabert, 1986. This statement has frequently been used in the complexity analysis of Gröbner basis algorithms.

Lemma 4.9. *Let I be in projective generic position with I^{hom} Cohen-Macaulay. Let H be the reduced $\prec_{\text{drl}}$-Gröbner basis of I (with the variables in $\mathbf{z}$ considered smaller as those in $\mathbf{x}$). Then*

$$\operatorname{lm}(H) \subset \operatorname{Mon}(\mathbf{x}).$$

In particular, if S is the $\prec_{\text{drl}}$-staircase of $I_1 \coloneqq I + \mathfrak{m}$, then the $\preceq_{\text{drl}}$-staircase of $I + \mathfrak{m}_u$ is given by

$$S_u \coloneqq \bigcup_{v \preceq_{\text{drl}} u} vS.$$

Proof. Note that the reduced $\prec_{\text{drl}}$-Gröbner basis of I^{gen} is precisely given by the homogenization of H, with the variable y considered the smallest. All statements made in this lemma now follow easily from Eisenbud, 1995, Theorem 15.13. □

This implies that when I is such that I^{hom} is Cohen-Macaulay and is in projective generic position then we can replace $\prec_1$ with $\prec_{\text{drl}}$ and H_u with H in the statements of Theorem 4.5 and Corollary 4.6. Now we are ready to prove:

Proof of Theorem 1.1. The genericity assumption on $f_1, \ldots, f_c$ implies that they form a Cohen-Macaulay ideal in projective generic position and that the ideal has degree $D = d_1 \cdots d_c$. Thus, Algorithm 2 can be called on $\{f_1, \ldots, f_c\}$, $\prec_{\text{drl}}$ and $\prec_{\text{lex}}$ in order to compute the reduced $\prec_{\text{drl}}$-Gröbner basis of $I = \langle f_1, \ldots, f_c \rangle$ and then the reduced $\prec_2$-Gröbner basis of $I\mathbb{K}(z_1, \ldots, z_{n-c})[x_1, \ldots, x_c]$. Finally, using Corollaries 4.6 and 4.7, we obtain the desired complexity. □

5 Using Tracers for Gröbner Basis Lifting

Again reusing the notation from Sections 2 and 3, recall that Algorithm 2 requires the $\prec_1$-Gröbner bases H_u of the ideals $I + \mathfrak{m}_u$. In this section, we will show how these Gröbner bases can be computed more optimally when $\prec_1$ is a block order on $\operatorname{Mon}(\mathbf{x} \cup \mathbf{z})$ eliminating $\mathbf{x}$. Recall that under the additional assumption that $\mathfrak{m}$ is a point of good specialization, by Theorem 2.10, the reduced $\prec_1$-Gröbner basis of I_u is given by $H_u \cup M_u$, where H is the reduced $\prec_1$-Gröbner basis of I^{gen} and M_u is the minimal generating set of $\mathfrak{m}_u$. If, in addition, the origin in the $\mathbf{z}$-space lies outside a suitably chosen Zariski-closed subset in the $\mathbf{z}$-space, then all coefficients that appear when computing H with an algorithm like F_4 lie in $\mathbb{K}[\mathbf{z}]_{\mathfrak{m}}$. This means, conversely, that when we

compute the Gröbner basis H_1 over $\mathbb{K}[\mathbf{z}]/\mathfrak{m} \cong \mathbb{K}$ with F_4, we can "remember" the computations that where performed and repeat the exact same computations over $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_u$ for any choice of $u \in \mathrm{Mon}(\mathbf{z})$ to obtain the set H_u.
This is formalized by the the concept of *tracers*, introduced by Traverso, 1989 for the purpose of multi-modular Gröbner basis computations.

5.1 Tracers

We start by recalling the definition of Macaulay matrices.

Definition 5.1. Let $F = \{f_1, \ldots, f_r\}$ be a finite set of polynomials in $\mathbb{K}[\mathbf{x}]$. Let $U_1, \ldots, U_r$ be finite sets of monomials in $\mathrm{Mon}(\mathbf{x})$ and denote $\mathbf{U} := \{U_1, \ldots, U_r\}$. Write $U_i = \{u_{i1}, \ldots, u_{ir_i}\}$. The *Macaulay matrix* $M_{F,\mathbf{U}}$ determined by F and $\mathbf{U}$ is the matrix with rows indexed by $\bigcup_{i=1}^{r} \bigcup_{j=1}^{r_i} u_{ij} f_i$ and columns indexed by the union of all supports of $u_{ij} f_i$ whose entry in row $u_{ij} f_i$ and column v is the coefficient of v in $u_{ij} f_i$.

The previously mentioned F_4 algorithm, introduced in Faugère, 1999, uses echelonization of Macaulay matrices to compute Gröbner bases. In this algorithm, the columns in each appearing Macaulay matrix are sorted descendingly by the monomial order for which one is computing a Gröbner basis and swapping columns is not allowed. During the computation, the rows of a given Macaulay matrix whose leading term changes during the echelonization are added to the Gröbner basis *in spe*. We can store a run of the F_4 algorithm using the following data structure:

Definition 5.2. A *tracer* T for a system of polynomials $f_1, \ldots, f_r$ consists in a finite sequence of finite sets of integers $I_1, \ldots, I_s$ together with a finite sequence of finite sets of monomials $\mathbf{U}_1, \ldots, \mathbf{U}_s$.

Given such a tracer T, we can try to follow it to recover a Gröbner basis of $\langle f_1, \ldots, f_r \rangle$ using Algorithm 3.

Algorithm 3 Following a Tracer

Input A system of polynomials $F := \{f_1, \ldots, f_r\}$, a tracer T for F, a monomial order $\prec$.

Output A finite set of polynomials G.

```
1 function trace(F, T, ≺)
2     G ← F
3     for k from 1 up to s
4         G_k ← {G[i] | i ∈ I_k}
5         M_k ← M_{G_k,U_k}   [with decreasing col. labels w.r.t. ≺]
6         M'_k ← echelonization of M_k
7         add all rows with changed leading term to G
8     return G
```

Definition 5.3. A tracer T for $F := \{f_1, \ldots, f_r\}$ and $\prec$ is called *good* if running Algorithm 3 is well-defined (i.e. there are no out-of-bounds accesses during its run) and if the output G of $\mathit{trace}(F, T, \prec)$ is a $\prec$-Gröbner basis of $\langle F \rangle$.

Going back to our setting, as in the beginning of this section, Let us assume that the origin in the $\mathbf{z}$-space lies outside of a suitably chosen Zariski-closed subset. Let us assume further that T is a good tracer for $\prec_1$ and the image in $(\mathbb{K}[\mathbf{z}]/\mathfrak{m})[\mathbf{x}] \simeq \mathbb{K}[\mathbf{x}]$ of a given generating set of I. Then, applying T to $\prec_1$ and the image in $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_u$ of the same generating set, for any $u \in \mathrm{Mon}(\mathbf{z})$, yields the set H_u.
Going further, if we have computed H_u using a good tracer and Algorithm 2 requires us to compute H_v for some $v \in \mathrm{Mon}(\mathbf{z})$ with $u \prec_{\mathrm{drl}} v$, then we can obtain the needed echelonizations of the Macaulay matrices over $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_v$ by lifting the ones previously computed over $\mathbb{K}[\mathbf{z}]/\mathfrak{m}_u$. Let us sketch how to do this in the next subsection.

5.2 Lifting of LU-factorization

To simplify the presentation we again assume, as in the proofs in Section 4, that $\mathbf{z} = \{z\}$ is a single variable. The method proposed here will however immediately transport to the setting where $\mathbf{z}$ consists in several variables. Let us denote $R := \mathbb{K}[\mathbf{z}]$ and $Q := \mathbb{K}(\mathbf{z})$. In this section, we want to lift an echelonization of a matrix, or in other words a LU-like decomposition (say for instance $PLUQ$, $PLEQ$, ...), from the field $R/\mathfrak{m}$ to $R/\mathfrak{m}^{k+1}$ for some $k \geq 0$. As previously, we assume that the steps of the computation of this decomposition over $R/\mathfrak{m}$ are exactly those over Q projected onto $R/\mathfrak{m}$, in particular this implies that no nonzero entry $p/q \in Q$ with $p, q \in R$ is such that $p \notin \mathfrak{m}$ and $q \in \mathfrak{m}$.
Again to simplify the presentation, we restrict ourselves to an exact LU-factorization of a full-rank matrix. Let $A \in Q^{n\times m}$ be a matrix with LU-factorization $A = LU$ with $L \in Q^{n\times n}$ and $U \in Q^{n\times m}$. We assume that A, L and U have coefficients in $R_{\mathfrak{m}}$. In that case we have matrices A_k, L_k and U_k with coefficients in $R/\mathfrak{m}^{k+1}$ and

$$A = A_k \bmod \mathfrak{m}^{k+1}, \qquad L = L_k \bmod \mathfrak{m}^{k+1}, \qquad U = U_k \bmod \mathfrak{m}^{k+1}.$$

We also have $A_k = L_k U_k \bmod \mathfrak{m}^{k+1}$.

Theorem 5.4. *Given A, L_0^{-1} and L_k, U_k there is an algorithm which computes L_{k+1} and U_{k+1} in $O\left(mn^2\right)$ arithmetic operations in $\mathbb{K}$.*

Proof. Recall that L_k is invertible and that $L_k^{-1} = L_0^{-1} \bmod \mathfrak{m}$. Thus,

$$\begin{aligned} A &= L_k U_k + f^{k+1}\delta_A \bmod \mathfrak{m}^{k+2} \\ &= L_k \left(U_k + f^{k+1} L_k^{-1}\delta_A\right) \bmod \mathfrak{m}^{k+2} \\ &= L_k \left(U_k + f^{k+1} L_0^{-1}\delta_A\right) \bmod \mathfrak{m}^{k+2}, \end{aligned}$$

and we define $B := U_k + f^{k+1}L_0^{-1}\delta_A$. As $L_0^{-1} \bmod \mathfrak{m}$ has already been computed, $L_0^{-1}\delta_A$ is computed in $O\left(mn^2\right)$ arithmetic operations in $\mathbb{K}$. It now suffices to prove that we can compute an LU-factorization of B over $R/\mathfrak{m}^{k+2}$ in $O\left(mn^2\right)$ operations in $\mathbb{K}$. Note that we can write

$$B = U_0 + fU_1' + \cdots + f^k U_k' + f^{k+1}V,$$

where $U_i' = \frac{1}{f^i}(U_i - U_{i-1})$ is upper triangular with coefficients in $R/\mathfrak{m}$, for all $1 \leq i \leq k$, and $V = \frac{1}{f^{k+1}}(B - U_k) \in \mathbb{K}^{n\times n}$. Echelonizing B comes down to reducing the rows of V with the rows of B above. In particular, since $b_{i,1} = f^{k+1}v_{i,1}$ for $i > 1$, it can be reduced using the pivot $b_{1,1}$, which is invertible in $R/\mathfrak{m}^{k+1}$ by assumption. Thus, the row operation

$$\begin{aligned} b_{i,j} &\leftarrow b_{i,j} - f^{k+1}\frac{b_{i,1}}{b_{1,1}} b_{1,j} \bmod \mathfrak{m}^{k+2} \\ &= f^{k+1}(v_{i,j} - \frac{v_{i,1}}{u'_{0,1,1}} u'_{0,1,j} \bmod \mathfrak{m}) \end{aligned}$$

consists only in performing row operations on the layer of valuation $k+1$ in B.
In other words, if

$$L' = \begin{bmatrix} 1 & & \\ -v_{2,1}/u'_{0,1,1}f^{k+1} & 1 & \\ \vdots & & \ddots \\ -v_{n,1}/u'_{0,1,1}f^{k+1} & & 1 \end{bmatrix}$$

then $L'B$ can be written

$$L'B = U_0' + fU_1' + \cdots + f^k U_k' + f^{k+1}V'$$

where we have $v'_{i,1} = 0$ for all $i \geq 2$. Proceeding with further row operations like this, we thus triangularize B. This clearly has the same complexity as echelonizing a matrix in $\mathbb{K}^{n\times m}$. □

Remark 5.1. Theorem 5.4 and its proof are also valid for $R = \mathbb{Z}$, $Q = \mathbb{Q}$, $\mathfrak{m} = p$ a good prime and $\mathbb{K} = \mathbb{F}_p$. We refer to Arnold, 2003; Vaccon, 2014, 2017; Winkler, 1988 where lifting Gröbner bases over p-adic numbers is performed and e.g. to Dixon, 1982; Haramoto and Matsumoto, 2009; Pan, 2011 for methods in p-adic linear algebra.

6 Benchmarks

In this section, we provide benchmarks for a proof-of-concept implementation of Algorithm 2. We first give a brief description thereof.
This implementation is written using the computer algebra system `OSCAR` (The OSCAR team, 2022) which itself is written in `Julia` (Bezanson, Edelman, Karpinski, & Shah, 2017). All required Gröbner basis computations use the Gröbner basis library `Groebner.jl`, also written in `Julia`, see Demin and Gowda, 2023. The main step in Algorithm 2, Algorithm 1, was implemented naively, close to the provided pseudocode, i.e. without the use of multiplication tensors to compute normal forms as described in Section 4.
For the below benchmarks, the following computations were performed, keeping the notation from the last sections:

1. Compute a $\prec_{\text{drl}}$-Gröbner basis for the polynomial ideal I in question.
2. Use this $\prec_{\text{drl}}$-Gröbner basis to compute a maximally independent set of variables modulo I, this gives us the partition of the variables into the subsets $\mathbf{x}$ and $\mathbf{z}$ as above.
3. If $\mathbf{z} = \{z_1, \dots, z_{n-c}\}$, choose random $a_1, \dots, a_{n-c} \in \mathbb{K}$ and make the coordinate substitution $z_i \leftarrow z_i - a_i$, as in Remark 2.1.
4. Choose $\prec_1$ as the block order eliminating $\mathbf{x}$ with $\prec_{\text{drl}}$ on both blocks of variables.
5. If $\mathbf{x} = \{x_1, \dots, x_c\}$, choose $\prec_2$ as the block order on $\text{Mon}(\mathbf{x})$ eliminating $\{x_1, \dots, x_{c-1}\}$ with $\prec_{\text{drl}}$ on this block.
6. Then the target Gröbner basis G contains a single polynomial g_{x_c} in the univariate polynomial ring $\mathbb{K}(\mathbf{z})[x_c]$.
7. Use Algorithm 2 to compute g_{x_c} (rather than the full set G).

In a certain generic situation (more precisely, when the variable x_c is "generic"), the computed polynomial g_{x_c} can be used for a primary decomposition of I, see e.g. Becker and Weispfenning, 1993, Sections 8.6 and 8.7, this motivates our choice of $\prec_2$. We never directly computed the reduced Gröbner basis H of I w.r.t. $\prec_1$, but only the reduced $\prec_1$-Gröbner basis H_u of the ideals $I + \mathfrak{m}_u$. When doing this, we found that the computations were better-behaved when choosing $\prec_1$ as above rather than $\prec_1 = \prec_{\text{drl}}$. All computations were performed with $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$ where p was a randomly chosen prime of 16 bits.
We compared the time this computation took with the computation of the set G using the `C` library `msolve` (Berthomieu, Eder, & Safey El Din, 2021) (which just runs the F4 algorithm with $\prec_2$). These timings are given in Table 1 in the Appendix. In this table, "OOM" means that the computation used more than the 100 GB memory limit that we set.
The polynomial systems used for these benchmarks are:

- ED(3,3) encodes the parametric *euclidean distance* problem for a hypersurface of degree 3 in 3 variables, see Draisma, Horobeţ, Ottaviani, Sturmfels, and Thomas, 2016;
- R1, R2, R3 come from a problem in Robotics, see García Fontán, Nayak, Briot, and Safey El Din, 2022;
- M2 and M3 are certain jacobian ideals of single multivariate polynomials which define singular hypersurfaces;

- The "PS", "Sing" and "SOS" systems are all critical loci of certain projections, see Eder, Lairez, Mohr, and Safey El Din, 2023a for a more detailed description;
- The RD(d) systems are randomly generated sequences of 3 polynomials of degree d in 4 variables.

All computations were performed on an Intel Xeon Gold 6244 CPU @ 3.60 GHz with 1.5 TB of memory.

Appendix

Table 1: Benchmarks for Algorithm 2

Polynomial System	Algorithm 2 Timing (in s)	`msolve` using $\prec_2$ Timing (in s)
ED(3,3)	121.34	OOM
R1	1.84	0.01
R2	2.74	0.01
R3	2.74	0.01
M2	152.81	6.54
M3	3.11	OOM
PS(2,10)	5.56	1251.94
PS(2,12)	120.10	OOM
Sing(2,10)	3.04	4.25
SOS(5,4)	21.56	18.98
SOS(6,4)	114.88	11366.36
SOS(6,5)	120.1	OOM
RD(3)	3.31	0.11
RD(4)	9.77	28.72
RD(5)	385.31	2277.56

Acknowledgments

The authors are supported by the joint ANR-FWF ANR-19-CE48-0015 ECARP and ANR-22-CE91-0007 EAGLES projects, ANR-19-CE40-0018 De Rerum Natura project, DFG Sonderforschungsbereich TRR 195 project and grants DIMRFSI 2021-02–C21/1131 of the Paris Île-de-France Region, FA8665-20-1-7029 of the EOARD-AFOSR, and Forschungsinitiative Rheinland-Pfalz. We thank Ch. Eder, P. Lairez, V. Neiger and M. Safey El Din for fruitful discussions.

References

Arnold, E. A. (2003). Modular algorithms for computing Gröbner bases. *Journal of Symbolic Computation*, *35*(4), 403–419.

Becker, T. (1994). On Gröbner bases under specialization. *Applicable Algebra in Engineering, Communication and Computing*, *5*(1), 1–8.

Becker, T., & Weispfenning, V. (1993). *Gröbner bases* (Vol. 141). Springer-Verlag, New York.

Berthomieu, J., Eder, C., & Safey El Din, M. (2021). msolve: A Library for Solving Polynomial Systems. *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*, 51–58.

Berthomieu, J., Eder, C., & Safey El Din, M. (2023). New efficient algorithms for computing Gröbner bases of saturation ideals (F4SAT) and colon ideals (Sparse-FGLM-colon). arXiv: 2202.13387 [cs, math]

Berthomieu, J., Neiger, V., & Safey El Din, M. (2022). Faster Change of Order Algorithm for Gröbner Bases under Shape and Stability Assumptions. *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, 409–418.
Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, *59*(1), 65–98.
Bostan, A., Jeannerod, C.-P., Mouilleron, C., & Schost, É. (2017). On Matrices With Displacement Structure: Generalized Operators and Faster Algorithms. *SIAM Journal on Matrix Analysis and Applications*, *38*(3), 733–775.
Bostan, A., Chyzak, F., Giusti, M., Lebreton, R., Lecerf, G., Salvy, B., & Schost, É. (2017). *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.)
Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem Nulldimensionalen Polynomideal* [Doctoral dissertation, Universität Innsbruck].
Collart, S., Kalkbrener, M., & Mall, D. (1997). Converting Bases with the Gröbner Walk. *Journal of Symbolic Computation*, *24*(3), 465–469.
Demin, A., & Gowda, S. (2023). Groebner.jl: A Package for Gröbner Bases Computations in Julia.
Dixon, J. D. (1982). Exact solution of linear equations using p-adic expansions. *Numerische Mathematik*, *40*(1), 137–141.
Draisma, J., Horobeţ, E., Ottaviani, G., Sturmfels, B., & Thomas, R. R. (2016). The Euclidean distance degree of an algebraic variety. *Found. Comput. Math.*, *16*(1), 99–149.
Ebert, G. L. (1983). Some comments on the modular approach to gröbner-bases. *SIGSAM Bull.*, *17*(2), 28–32
Eder, C., Lairez, P., Mohr, R., & Safey El Din, M. (2023a). A Direttissimo Algorithm for Equidimensional Decomposition. *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, 260–269.
Eder, C., Lairez, P., Mohr, R., & Safey El Din, M. (2023b). A signature-based algorithm for computing the nondegenerate locus of a polynomial system. *Journal of Symbolic Computation*, *119*, 1–21.
Eisenbud, D. (1995). *Commutative algebra: With a view toward algebraic geometry*. Springer New York.
Faugère, J.-C. (1999). A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, *139*(1), 61–88.
Faugère, J.-C. (2002). A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, 75–83.
Faugère, J.-C., Gaudry, P., Huot, L., & Renault, G. (2014). Sub-cubic change of ordering for Gröbner basis: A probabilistic approach. *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, 170–177.
Faugère, J.-C., Gianni, P., Lazard, D., & Mora, T. (1993). Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation*, *16*(4), 329–344.
Faugère, J.-C., & Mou, C. (2017). Sparse FGLM algorithms. *Journal of Symbolic Computation*, *80*, 538–569.
García Fontán, J., Nayak, A., Briot, S., & Safey El Din, M. (2022). Singularity Analysis for the Perspective-Four and Five-Line Problems. *International Journal of Computer Vision*, *130*(4), 909–932.
Gianni, P. (1989). Properties of Gröbner bases under specializations. In J. H. Davenport (Ed.), *Eurocal '87* (pp. 293–297). Springer.
Giusti, M., Lecerf, G., & Salvy, B. (2001). A Gröbner Free Alternative for Polynomial System Solving. *Journal of Complexity*, *17*(1), 154–211.
Gräbe, H.-G. (1993). On Lucky Primes. *Journal of Symbolic Computation*, *15*(2), 199–209.
Guillaume, P., & Huard, A. (2000). Multivariate Padé approximation. *Journal of Computational and Applied Mathematics*, *121*(1), 197–219.
Haramoto, H., & Matsumoto, M. (2009). A p-adic algorithm for computing the inverse of integer matrices. *Journal of Computational and Applied Mathematics*, *225*(1), 320–322
Hubert, E. (2003). Notes on Triangular Sets and Triangulation-Decomposition Algorithms I. *Symbolic and Numerical Scientific Computation*, 1–39.

Lejeune-Jalabert, M. (1986). *Effectivité de calculs polynomiaux. Cours de D.E.A.* Laboratoire de Mathématiques associé au C. N. R. S.
Neiger, V., & Schost, É. (2020). Computing syzygies in finite dimension using fast linear algebra. *Journal of Complexity*, *60*, 101502.
Pan, V. Y. (2011). Nearly optimal solution of rational linear systems of equations with symbolic lifting and numerical initialization. *Computers & Mathematics with Applications*, *62*(4), 1685–1706
Pauer, F. (1992). On lucky ideals for Gröbner basis computations. *Journal of Symbolic Computation*, *14*(5), 471–482.
Schost, É. (2003). Computing parametric geometric resolutions. *Appl. Algebra Engrg. Comm. Comput.*, *13*(5), 349–393
Schost, É., & St-Pierre, C. (2023). P-adic algorithm for bivariate Gröbner bases. *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, 508–516.
The OSCAR team. (2022). OSCAR – Open Source Computer Algebra Research System.
Traverso, C. (1989). Gröbner trace algorithms. In P. Gianni (Ed.), *Symbolic and Algebraic Computation* (pp. 125–138). Springer.
Traverso, C. (1996). Hilbert Functions and the Buchberger Algorithm. *Journal of Symbolic Computation*, *22*(4), 355–376.
Vaccon, T. (2014). Matrix-f5 algorithms over finite-precision complete discrete valuation fields. *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, 397–404.
Vaccon, T. (2017). Matrix-f5 algorithms over finite-precision complete discrete valuation fields. *Journal of Symbolic Computation*, *80*, 329–350.
Winkler, F. (1988). A p-adic approach to the computation of Gröbner bases. *Journal of Symbolic Computation*, *6*(2), 287–304.