Computing Generic Fibers of Polynomial Ideals with FGLM and Hensel Lifting

Jérémy Berthomieu, Rafael Mohr

Computing Generic Fibers of Polynomial Ideals with FGLM and Hensel Lifting

Jérémy Berthomieu
Sorbonne Université, CNRS, LIP6
F-75005 Paris, France
jeremy.berthomieu@lip6.fr

Rafael Mohr
Sorbonne Université, CNRS, LIP6
F-75005 Paris, France
Rheinland-Pfälzische Technische Universität
Kaiserslautern-Landau, Fachbereich Mathematik
G-67663 Kaiserslautern, Germany
rafael.mohr@lip6.fr

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. In analogy with Hensel lifting, we show that this algorithm has a complexity quasi-linear in the number of terms of certain $\mathfrak{m}$-adic expansions we compute. Some provided experimental data also demonstrates the practical efficacy of our algorithm.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**; • **Theory of computation** → **Design and analysis of algorithms**.

KEYWORDS

Gröbner basis; polynomial system solving; change of monomial order; Hensel lifting

ACM Reference Format:
Jérémy Berthomieu and Rafael Mohr. 20NN. Computing Generic Fibers of Polynomial Ideals with FGLM and Hensel Lifting. In *Proceedings of the 20NN International Symposium on Symbolic and Algebraic Computation (ISSAC 'NN), July XX–YY, 20NN, City, ZZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.xxxx/xxxxxxx.xxxxxxx

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 [3] 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 [10], such as F_4 [17] and F_5 [18], are substantially more well-behaved when used with non-elimination orders (most notably, the *degree reverse lexicographical* $\prec_{\text{drl}}$ 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 [38], the Gröbner walk algorithm by [11] and, most notably for this paper, the FGLM algorithm [20] and its variants [6, 19, 21, 31].

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 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 [5, 16].

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* $I^{\text{gen}} := I \cdot \mathbb{K}(\mathbf{z})[\mathbf{x}]$ of I is zero-dimensional. Given a Gröbner basis of I w.r.t. a monomial order $\prec_{\text{in}}$, we want to compute a Gröbner basis G of I^{gen} w.r.t. another monomial order $\prec_{\text{out}}$. One key motivation to solve this problem is that, when $\prec_{\text{out}}$ is a suitable elimination order, the Gröbner basis G can be used to

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 the referees for their valuable comments on the paper and Ch. Eder, P. Lairez, V. Neiger and M. Safey El Din for fruitful discussions.

ISSAC 'NN, July XX–YY, 20NN, City, ZZ, USA

ACM ISBN xxx-x-xxxx-xxxx-x/xx/xx…$yy.00
https://doi.org/10.xxxx/xxxxxxx.xxxxxxx

compute a primary decomposition of the ideal I (or an irreducible decomposition of the algebraic set defined by I), see [3] for details. Being able to compute such decompositions has numerous applications, we mention for example the algorithm presented in [27] which uses primary decompositions to compute so-called Whitney stratifications of singular varieties.

The algorithm we design to solve this problem 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 $\prec_{\text{in}}$ to $\prec_{\text{out}}$). Our proposed solution to this problem can be seen as a combination of the previously mentioned FGLM algorithm with classical Hensel lifting techniques. More precisely, if we let $\mathfrak{m} := \langle \mathbf{z} \rangle$, then, under some assumptions which are detailed in this paper, we will compute G by computing its image in $(\mathbb{K}[\mathbf{z}]/\mathfrak{m})[\mathbf{x}] \simeq \mathbb{K}[\mathbf{x}]$ and then lifting it modulo higher and higher powers of $\mathfrak{m}$. This lifting step uses the same core idea as the FGLM algorithm. With this approach, we expect that our algorithm can be transported without much difficulty to the setting where a Gröbner basis G of a zero-dimensional ideal in $\mathbb{Q}[\mathbf{x}]$ is required: Given p a well-chosen prime number and $k \in \mathbb{N}^*$ sufficiently large, it would extract G from its image in $(\mathbb{Z}/p\mathbb{Z})[\mathbf{x}]$ and then lift it modulo p^k. We show that, similar to classical Hensel lifting, our algorithm runs in arithmetic complexity quasi-linear in the number of terms of degree at most 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 $f_1, \ldots, f_c$ be generic polynomials of respective degrees $d_1, \ldots, d_c$ in $\mathbb{K}[\mathbf{z}, \mathbf{x}]$. Assume that the $\prec_{\text{drl}}$-Gröbner basis of $I = \langle f_1, \ldots, f_c \rangle$ is known and that the $\prec_{\text{out}}$-Gröbner basis G of $I \cdot \mathbb{K}(\mathbf{z})[\mathbf{x}]$ has coefficients which are rational functions with degrees at most δ in the numerators and denominators. Let $\mathrm{m}_{2\delta}$ be the number of monomials in $\mathbf{z}$ up to degree 2δ. Then, one can compute G up to precision 2δ using $\widetilde{O}\left(\mathrm{m}_{2\delta} c (d_1 \cdots d_c)^3\right)$ operations in $\mathbb{K}$.*

Note that knowing G up to precision 2δ is enough to recover G by means of Padé approximants, see Lemma 3.6.

Related Work. Gröbner bases of generic fibers, as defined in the previous paragraph, are classically computed using *block monomial orders*, see e.g. [3, Lemma 8.93]. Besides that, morally similar to our algorithm, there is a rich body of literature about multi-modular Gröbner basis computations [1, 14, 32, 37] and Hensel/modular lifting techniques for Gröbner bases [25, 34, 39].

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. [24, 33], and *triangular sets*, see e.g. [28] 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 [2, 23, 29].

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. Finally we give some benchmarks for a `Julia` implementation of our main algorithm in Section 5, 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 $\mathrm{Mon}(\mathbf{x})$

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

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

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

Definition 2.2. The *degree reverse lexicographic* $\prec_{\text{drl}}$ order on $\mathrm{Mon}(\mathbf{x})$ is defined as follows for $u, v \in \mathrm{Mon}(\mathbf{x})$: $u \prec_{\text{drl}} v$ 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 $\prec_{\mathbf{x}}$ on $\mathrm{Mon}(\mathbf{x})$ and a monomial order $\prec_{\mathbf{z}}$ on $\mathrm{Mon}(\mathbf{z})$. The corresponding *block order eliminating* $\mathbf{x}$ is defined as follows: $u \prec v$ iff $u_{\mathbf{x}} \prec_{\mathbf{x}} v_{\mathbf{x}}$ or $u_{\mathbf{x}} = v_{\mathbf{x}}$ and $u_{\mathbf{z}} \prec_{\mathbf{z}} v_{\mathbf{z}}$ 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 on $\mathrm{Mon}(\mathbf{x})$.

(1) The set $S_{I,\prec} := \{u \in \mathrm{Mon}(\mathbf{x}) \mid u \notin \mathrm{lm}_{\prec}(I)\}$ is 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 $\mathrm{nf}_{I,\prec}(f)$.

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 is called *reduced* if, for any $g \in G$, no monomial in the support of g is divisible by any element in $\mathrm{lm}_{\prec}(G \setminus \{g\})$.

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. [3, 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. Finally, recall that reduced Gröbner bases are unique for a given ideal and monomial order.

2.2 Points of Good Specialization

We start by fixing some notation. For 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 $I^{\mathrm{gen}} := I \cdot \mathbb{K}(\mathbf{z})[\mathbf{x}] \neq \mathbb{K}(\mathbf{z})[\mathbf{x}]$ is a zero-dimensional ideal.

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

Let us introduce some further notation.

Definition 2.8. We denote for a monomial $u \in \mathrm{Mon}(\mathbf{z})$

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

$\mathfrak{m} := \mathfrak{m}_1 = \langle \mathbf{z} \rangle$, as well as $\mathrm{next}(u) = \min\{v \in \mathrm{Mon}(\mathbf{z}) \mid v >_{\mathrm{drl}} u\}$.

Definition 2.9. Let $g \in \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$. Write

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

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

$$\frac{p_w}{q_w} = \sum_{v \in \mathrm{Mon}(\mathbf{z})} r_{w,v} v \in \mathbb{K}[[\mathbf{z}]]$$

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

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

For a set $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$ we define $G_u := \{g_u \mid g \in G\}$.

Let $G \subset \mathbb{K}(\mathbf{z})[\mathbf{x}]$ be the reduced Gröbner basis of I^{gen} w.r.t. a monomial order $\prec_{\mathbf{x}}$ on $\mathrm{Mon}(\mathbf{x})$. Our algorithms will work under the assumption that $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$ and that given the set G_u we can lift G_u uniquely to $G_{\mathrm{next}(u)}$. In fact the condition $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$ turns out to be sufficient. We capture this in a definition:

Definition 2.10. We say that $\mathfrak{m}$ is a *point of good specialization* (for $\prec_{\mathbf{x}}$) if $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$.

Remark 2.11. *By definition, being a point of good specialization is a Zariski-open condition, so that, if $\mathbb{K}$ is infinite, it is ensured with probability 1 after replacing each $z_i \in \mathbf{z}$ by $z_i - a_i$ for randomly chosen $a_i \in \mathbb{K}$. In Remark 4.11 we point out a situation in which an upper bound for the probability that $\mathfrak{m}$ is a point of good specialization can be given intrinsically in terms of I if $\mathbb{K}$ is finite.*

First we show

Theorem 2.12. *If the ideal $\mathfrak{m}$ is a point of good specialization, then the $\mathbb{K}[\mathbf{z}]_{\mathfrak{m}}$-module $\mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]/I$ is free of finite rank.*

Proof. Write $A := \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}$, $K = \mathbb{K}(\mathbf{z})$ for the field of fractions of A and $F := A[\mathbf{x}]/I$.
Suppose that $\mathfrak{m}$ is a point of good specialization so that $G \subset A[\mathbf{x}]$. Let $S := S_{I^{\mathrm{gen}}, \prec_{\mathbf{x}}}$, note that S is finite. We first show that S generates F as an A-module. Let $u \in \mathrm{Mon}(\mathbf{x})$ with $u \notin S$ so that $u \in \mathrm{lm}_{\prec_{\mathbf{x}}}(I^{\mathrm{gen}})$. Then there exists $g \in G$ and $v \in \mathrm{Mon}(\mathbf{x})$ such that $\mathrm{lm}_{\prec_{\mathbf{x}}}(vg) = u$ and therefore such that $\mathrm{lm}_{\prec_{\mathbf{x}}}(u - vg) \prec_{\mathbf{x}} u$.
Reducing further the expression $u - vg$ by G is done with arithmetic over A only and hence shows that, in F, we can write $u = \sum_{s \in S} r_s s$ with $r_s \in A$. This shows that S generates the A-module F. Now, to prove that F is free over A, it suffices to show that there are no non-trivial A-relations between the elements of S. Suppose that for certain $s_1, \ldots, s_t \in S$, there is a relation

$$\sum_{i=1}^{t} r_i s_i = 0$$

in F with $r_i \in A \setminus \{0\}$ for all i. This gives in particular a relation between the s_i over K. Hence, if j is such that s_j is $\prec_{\mathbf{x}}$-maximal among the s_i, then $s_j \in L$, but $L \cap S = \emptyset$, a contradiction. □

We now give some further properties regarding points of good specializations. Item 1 in the theorem below will be used to show the correctness of our lifting algorithm in Section 3 whereas Item 2 will be used for our complexity analysis in Section 4.

Theorem 2.13. *Suppose that $\mathfrak{m}$ is a point of good specialization.*

(1) For each $u \in \mathrm{Mon}(\mathbf{z})$ and $z_i \in \mathbf{z}$, the 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) Let $\prec$ be the block order eliminating $\mathbf{x}$ with $\prec = \prec_{\mathbf{x}}$ on $\mathrm{Mon}(\mathbf{x})$ and $\prec = \prec_{\mathrm{drl}}$ on $\mathrm{Mon}(\mathbf{z})$. Let M_u be the (unique) minimal generating set of $\mathfrak{m}_u$. Then, the reduced $\prec$-Gröbner basis of I_u is precisely $G_u \cup M_u$.

Proof. We reuse the notation from the proof of Theorem 2.12. By Theorem 2.12, $\mathfrak{m}$ being a point of good specialization implies that F is a free R-module of finite rank.
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 $V_u := u\mathbb{K}[\mathbf{z}, \mathbf{x}]/I_u$ 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, if $F \cong A^r$, we have,

$$V_u \simeq (uA/\mathfrak{m}_u)^r \simeq \mathbb{K}^r.$$

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 $S := S_{I^{\mathrm{gen}}, \prec_{\mathbf{x}}}$. It suffices to show that

$$S_{I_u, \prec} = S_u := \bigcup_{v \preceq_{\mathrm{drl}} u} vS.$$

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 $\mathbb{K}[\mathbf{z}, \mathbf{x}]/(I + \mathfrak{m}_u) \simeq F/\mathfrak{m}_u$, 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 $\prec_{\mathbf{x}}$-minimal such that there exists some $w \in \mathrm{Mon}(\mathbf{z})$, $w \preceq_{\mathrm{drl}} u$ with $ws \in \mathrm{lm}_{\prec}(I_u)$. By minimality, the $\prec$-normal form of ws w.r.t. I_u has support in $\bigcup_{v \preceq_{\mathrm{drl}} u} \{vt \mid t \prec_{\mathbf{x}} s, t \in S\} \subset S_u$, therefore inducing a linear dependence between the elements of S_u, a contradiction. □

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

Proof. Note that Item 2 in Theorem 2.13 implies that a $\mathbb{K}$-basis of $\mathbb{K}[\mathbf{z},\mathbf{x}]/I_u$ is given by $\bigcup_{v \preceq_{\mathrm{drl}} u} vS_{I^{\mathrm{gen}},\prec_{\mathbf{x}}}$ which is a finite set. □

3 THE MAIN ALGORITHM

As in the last section, we fix an ideal $I \subset \mathbb{K}[\mathbf{z},\mathbf{x}]$ with generically finite fiber over $\mathbb{K}[\mathbf{z}]$, we will use the notation introduced in Definition 2.8 and Definition 2.9. We now further fix any monomial order $\prec_{\mathrm{out}}$ on $\mathrm{Mon}(\mathbf{x})$ and another monomial order $\prec_{\mathrm{in}}$ on $\mathrm{Mon}(\mathbf{x} \cup \mathbf{z})$. We suppose that $\mathfrak{m}$ is a point of good specialization for $\prec_{\mathrm{out}}$.
Suppose that we can compute, with some black box, the reduced $\prec_{\mathrm{in}}$-Gröbner basis H_u of I_u for any $u \in \mathrm{Mon}(\mathbf{z})$. Our goal is to use this data to compute the reduced $\prec_{\mathrm{out}}$-Gröbner basis G of I^{gen}.

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 and let $\mathbf{x}$ be the set of remaining variables, see [3, 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 [3, Corollary 9.28].*

Let us sketch our strategy. By the assumption that $\mathfrak{m}$ is a point of good specialization, we have $G \subset \mathbb{K}[\mathbf{z}]_{\mathfrak{m}}[\mathbf{x}]$. Recall that $\mathfrak{m}_1 = \mathfrak{m}$. We start by computing the $\prec_{\mathrm{in}}$-Gröbner basis H_1 of $I_1 = I + \mathfrak{m}$. Then, we run the FGLM algorithm [20] with H_1 to obtain the reduced $\prec_{\mathrm{out}}$-Gröbner basis of the image of I in $(\mathbb{K}[\mathbf{z}]/\mathfrak{m})[\mathbf{x}] \simeq \mathbb{K}[\mathbf{x}]$. By Theorem 2.13 this Gröbner basis will now precisely be the set G_1 in the notation introduced in Definition 2.9.
For a monomial $u \in \mathrm{Mon}(\mathbf{z})$, let $v := \mathrm{next}(u)$. Starting with $u = 1$ and a given $g_1 \in G_1$, we will lift g_u to g_v by performing linear algebra in the finite-dimensional, see Corollary 2.14, $\mathbb{K}$-vector space $v\mathbb{K}[\mathbf{z},\mathbf{x}]/I_v$, using the $\prec_{\mathrm{in}}$-Gröbner basis H_v. This will rely on Item 1 in Theorem 2.13.

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 $\prec_{\mathrm{in}}$-Gröbner basis of I when $\prec_{\mathrm{in}}$=$\prec_{\mathrm{drl}}$ and I satisfies a certain genericity assumption. Under the assumption that $\prec_{\mathrm{in}}$ is a suitable block order and that $\mathfrak{m}$ is also a point of good specialization for $\prec_{\mathrm{in}}$ restricted to $\mathrm{Mon}(\mathbf{x})$ one can give a* tracer-based *[37] method to compute the sets H_u. This will be the subject of a future paper.*

This lifting step is now given by Algorithm 1.

Theorem 3.3. *If $\mathfrak{m}$ is a point of good specialization for $\prec_{\mathrm{out}}$, then Algorithm 1 terminates and is correct in that it satisfies its output specification.*

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. $\prec_{\mathrm{in}}$, the set $S_{I_1,\prec_{\mathrm{out}}}$.
Output The corresponding element $g_v \in G_v$.

1 **function** *lift*$(g_u, H_v, S_{\prec_{\mathrm{out}}})$
2 $c \leftarrow \mathrm{nf}_{I_v,\prec_{\mathrm{in}}}(g_u)$ [via H_v]
3 **if** $c = 0$ **return** g_u
4 **else**
5 find $(\alpha_w)_{w \in S_{I_1,\prec_{\mathrm{out}}}}$ s.t. $c = \sum_{w \in S_{I_1,\prec_{\mathrm{out}}}} \alpha_w \,\mathrm{nf}_{I_v,\prec_{\mathrm{in}}}(uw)$ [via H_v]
6 **return** $g_u - \sum_{w \in S_{I_1,\prec_{\mathrm{out}}}} \alpha_w uw$

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 $\mathrm{nf}_{I_v,\prec_{\mathrm{in}}}(uw)$ in line 5 are linearly independent thanks to Item 1 of Theorem 2.13. Thus, there exists at most one choice of coefficients $\alpha_w, w \in S_{I_1,\prec_{\mathrm{out}}}$, such that $c = \sum_{w \in S_{I_1,\prec_{\mathrm{out}}}} \alpha_w \,\mathrm{nf}_{I_v,\prec_{\mathrm{in}}}(uw)$. Furthermore, since $\mathfrak{m}$ is a point of good specialization, 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. □

Remark 3.4. *We want to emphasize that our algorithms never verify deterministically (and cannot verify deterministically) whether $\mathfrak{m}$ is a point of good specialization, this is a probabilistic assumption. Nonetheless, running Algorithm 1 can sometimes detect when $\mathfrak{m}$ is not a point of good specialization, namely if there exists no or more than one solution to the linear system in line 5 of Algorithm 1. In this case one would apply a random change of coordinates $z_i \leftarrow z_i - a_i$ for each $z_i \in \mathbf{z}$ and restart the computation.*

Example 3.5. Let us unroll Algorithm 1 o,n the following example. We work over the polynomial ring $\mathbb{F}_{11}[z, x_1, x_2, x_3]$, where $\mathbb{F}_{11}$ is the finite field with eleven elements. Our ideal I is generated by

$$(z+8)+x_1+x_2+x_3,\ \ (z+8)x_1+x_1x_2+x_2x_3+x_3(z+8),$$
$$(z+8)x_1x_2+x_1x_2x_3+x_2x_3(z+8)+x_3(z+8)x_1,(z+8)x_1x_2x_3-1.$$

Following Remark 3.1, one verifies that $\mathbb{F}_{11}[z] \to \mathbb{F}_{11}[z, x_1, x_2, x_3]/I$ is injective with generically finite fiber.
Readers may recognize this example as the Cyclic 4 polynomial system where we have replaced the variable z by the random choice $z + 8$ to ensure, probabilistically, that $\mathfrak{m} = \langle z \rangle$ is a point of good specialization.
For our orders we choose $\prec_{\mathrm{in}}$=$\prec_{\mathrm{drl}}$ on $\mathrm{Mon}(\mathbf{x} \cup \{z\})$ and $\prec_{\mathrm{out}}$ as the lexicographic order on $\mathrm{Mon}(\mathbf{x})$. Now, the set G_1 is given by

$$G_1 := \{x_3^2 + 6, x_2 + 8, x_1 + x_3\}.$$

Hence $S_{I_1,\prec_{\mathrm{out}}} = \{1, x_3\}$. Assuming that $g_1 := x_3^2 + 6$ is the image of some element g in the target Gröbner basis $G \subset \mathbb{F}_{11}(z)[\mathbf{x}]$, we now try to lift g_1 to g_z, i.e. the image of g modulo $\mathfrak{m}_z = \langle z^2 \rangle$, so that, in the notation of Algorithm 1, we have $u = 1$ and $v = z$.
If such a g exists, there must now exist, by Item 1 in Theorem 2.13, unique scalars $\alpha_1, \alpha_{x_3} \in \mathbb{F}_{11}$ such that

$$g_z = g_1 + \alpha_1 z + \alpha_{x_3} z x_3 = 0 \mod I_z = I + \langle z^2 \rangle,$$

and Algorithm 1 attempts to compute these scalars by finding a linear relation between the normal forms w.r.t. $\prec_{\mathrm{in}}$ of g_1, z and zx_3

modulo I_z. Using an $\prec_{\text{in}}$-Gröbner basis of I_z, we find that $S_{I_z,\prec_{\text{in}}} = \{1, z, x_3, zx_3\}$ and we compute, using normal form computations

$$\begin{aligned}\text{nf}_{I_2,\prec_{\text{in}}}(g_1) &= (0,7,0,0)\\ \text{nf}_{I_2,\prec_{\text{in}}}(z) &= (0,1,0,0)\\ \text{nf}_{I_2,\prec_{\text{in}}}(zx_3) &= (0,0,0,1)\end{aligned}$$

so that finally, $\alpha_1 = 6$ and $\alpha_{x_3} = 0$ which yields for g_z the unique candidate $g_z = x_3^2 + (4z + 6)$, finishing the example.

Algorithm 1 is only able to compute the set G_u for a monomial $u \in \text{Mon}(\mathbf{z})$, 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 image g_u of a given element $g \in G$ as

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

Now we have for the coefficient $p_w/q_w \in \mathbb{K}(\mathbf{z})$ of w in g

$$p_w - q_w \sum_{v \preceq_{\text{drl}} u} r_{w,v} v = 0 \bmod \mathfrak{m}_u, \tag{1}$$

which determines a set of linear equations in the unknown coefficients of p_w and q_w. Let $d := \deg u$. Suppose that $\deg \text{next}(u) = d + 1$, 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 $\deg p_w \le d_1$ and $\deg q_w \le d_2$, then the linear system (1) 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_1, d_2) *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_1, d_2) of λ_w is equal to p_w/q_w, see e.g. [26, Proposition 2.1]:

Lemma 3.6. *Let p/q be a Padé approximant of order (d_1, d_2) of λ_w. If $d_1 \ge \deg p_w$ and $d_2 \ge \deg q_w$ 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 $g_{\text{cand}} \in \mathbb{K}(\mathbf{z})[\mathbf{x}]$ 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 $v := \text{next}(u)$ we have

$$g_{\text{cand}} = g_v \bmod \mathfrak{m}_v.$$

Based on this, we now obtain Algorithm 2 for computing the set G probabilistically. We state this algorithm in an informal way. In Line 7 by "lifting G_{lift} to degree d" we mean that we compute the set G_u where u is the $\prec_{\text{drl}}$-maximal monomial of degree d.
Clearly, by Theorem 3.3 and Lemma 3.6, this algorithm returns the correct result if the computed Padé approximants are of sufficiently large degree and $\mathfrak{m}$ is a point of good specialization.

Remark 3.7. *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 $c_{I_v,\prec_{\text{in}}}(uv')$ where $u \prec_{\text{drl}} v' \preceq_{\text{drl}} v$.*

Example 3.8 (Example 3.5 continued). Let us try to see how Algorithm 2 recovers the element denoted g in Example 3.5. First, Algorithm 2 lifts the element $g_1 = x_3^2 + 6$ to degree $d = 2$, i.e. we compute the image of g modulo z^3 (line 7 in Algorithm 2). This yields, in our usual notation,

$$g_2 = x_3^2 + (2z^2 + 4z + 6).$$

Now we attempt (line 8 in Algorithm 2) to find a Padé approximation of order $(1, 1)$ for g, i.e., here, $p, q \in \mathbb{F}_{11}[z]$ of degree at most one such that $p/q = 2z^2 + 4z + 6 \bmod z^3$ by solving a linear system as outlined above. This yields the candidate

$$g_{\text{cand}} = x_3^2 + \frac{z+6}{5z+1}$$

which satisfies $g_{\text{cand}} = g_2 \bmod z^3$. Next, in line 9, we lift g one monomial higher, i.e. modulo z^4. This yields

$$g_3 = x_3^2 + (7z^3 + 2z^2 + 4z + 6),$$

But p/q has now the truncated power series $z^3 + 2z^2 + 4z + 6$, so that g_2 does not have stable Padé approximation. Hence we double d to 4 and lift g_3 to g_4, i.e. from modulo z^4 to modulo z^5, and attempt another Padé approximation. This time, computing a Padé approximation of order $(2, 2)$, this yields the candidate

$$g_{\text{cand}} = x_3^2 + \frac{1}{10z^2 + 6z + 2}.$$

Finally, we lift g_4 to g_5 and find that $g_5 = g_{\text{cand}} \bmod z^6$. So g_4 has stable Padé approximation and we terminate with $g := g_{\text{cand}}$. Computing the $\prec_{\text{out}}$-Gröbner basis G of I^{gen} using block orders as in [2, Lemma 8.93] shows that g is indeed the correct element.

Algorithm 2 Computing the generic fiber

Input A generating set F of I, a monomial order $\prec_{\text{in}}$, a monomial order $\prec_{\text{out}}$.
Output A guess for the set G.

```
1  function genfglm(F, ≺in, ≺out)
2      H1 ← reduced ≺in-Gröbner basis of I1     [using F]
3      Glift, S≺out ← fglm(H1, ≺out)
4      Gresult ← ∅
5      d ← 2
6      while Glift ≠ ∅
7          Glift ← lift Glift to degree d using Algorithm 1
8          Run pade(g, d/2, d/2) for all g ∈ Glift
9          Lift Glift one monomial higher
10         add to Gresult all elements with stable Padé approx.
11         remove the corresponding elements from Glift
12         d ← 2d
13     return Gresult
```

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 [20]. We will reuse the notation from the last section. We now add the additional assumption that $\prec_{\text{in}}$ is a block order eliminating $\mathbf{x}$ with $\prec_{\text{in}} = \prec_{\text{drl}}$ on $\text{Mon}(\mathbf{z})$ and that $\mathfrak{m}$ is a point of good specialization for both $\prec_{\text{in}}$ and $\prec_{\text{out}}$. Here, we analyze the number of arithmetic operations in $\mathbb{K}$ required to obtain the sought $\prec_{\text{out}}$-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*:

Definition 4.1. Let $k, D \in \mathbb{N}$. A block matrix $M = (M_{pq})_{0 \le p,q < k} \in \mathbb{K}^{kD \times kD}$ 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. [9]:

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

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

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 vein, 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 [9], 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 $N = (N_{ij})_{0 \le i,j < D} \in \mathbb{K}^{kD \times kD}$ 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 $\alpha(M) \le 2D$, 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 $S \coloneqq S_{I,\prec}$. The multiplication tensor of I w.r.t. $\prec$ is defined as the 3-tensor

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

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. In this section, we denote by D the degree of I^{gen}, i.e. the $\mathbb{K}(z)$-dimension of $\mathbb{K}(z)[\mathbf{x}]/I^{\mathrm{gen}}$. Note that this degree is upper-bounded by that of I. For $u \in \mathrm{Mon}(\mathbf{x})$ $\prec_{\mathrm{drl}}$-maximal of degree k, we also denote $I_k \coloneqq I_u$ and similarly $H_k \coloneqq H_u$ and $G_k \coloneqq G_u$ with these sets defined as in the last section.

To simplify the notation, we assume in the following two proofs, that the set $\mathbf{z} = \{z\}$ consists of a single variable. It will be clear from the proofs that they translate accordingly to the more general setting where $\mathbf{z}$ consists of several variables.
In our assumed setting, we obtain the following statement for computing multiplication tensors:

Theorem 4.5. *Let $k \in \mathbb{N}$ and m_k be the number of monomials in $\mathbf{z}$ up to degree k. 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. $\prec_{\mathrm{in}}$. Then the multiplication tensor of I_{2k} w.r.t. $\prec_{\mathrm{in}}$ is computed in arithmetic complexity $\widetilde{O}\left(\mathrm{m}_k c D^3\right)$.*

Proof. Let $S \coloneqq S_{I_1,\prec_{\mathrm{in}}}$. Applying the third item of Theorem 2.13 with $\prec_{\mathrm{in}}$ instead of $\prec$, since $\prec_{\mathrm{in}}$ eliminates $\mathbf{x}$, for any $\ell \in \mathbb{N}$, yields

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

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

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

The matrix $(c_{I_{2k},\prec_{\mathrm{in}}}(zu))_{u \in S_{I_{2k},\prec_{\mathrm{in}}}}$ 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 $S_0 \coloneqq \bigcup_{i=0}^{k-1} z^i S$ and $S_1 \coloneqq \bigcup_{i=k}^{2k-1} z^i S = z^k S_0$. 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 (m_k, D) and $M_{x_i,0}$ is known as part of the $\prec_{\mathrm{in}}$-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 $\prec_{\mathrm{in}}$. 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 $b \prec_{\mathrm{in}} v$. 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 (m_k, D). This is done in time $\widetilde{O}\left(\mathrm{m}_k D^2\right)$ 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 $k \in \mathbb{N}$ and m_k be the number of monomials in $\mathbf{z}$ up to degree k. 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. $\prec_{\mathrm{in}}$, the $\prec_{\mathrm{in}}$-normal forms of the $\prec_{\mathrm{out}}$-staircase of I_1 w.r.t. I_k and the $\prec_{\mathrm{in}}$-normal forms of the minimal $\prec_{\mathrm{out}}$-leading monomials of I_0 w.r.t. I_k. Then G_{2k} is computed in arithmetic complexity $\widetilde{O}\left(\mathrm{m}_k c D^3\right)$.*

Proof. By Theorem 4.5, the $\prec_{\mathrm{in}}$-multiplication tensor of I_{2k} can be computed in arithmetic complexity $\widetilde{O}\left(\mathrm{m}_k c D^3\right)$. Having computed this tensor, we proceed as follows: let S be the $\prec_{\mathrm{in}}$-staircase of $I_1 = I + \langle z \rangle$, T be the $\prec_{\mathrm{out}}$-staircase of I_1 and L be the set of minimal $\prec_{\mathrm{out}}$-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 $\prec_{\mathrm{in}}$-normal forms of each element in $T \cup L$ w.r.t. I_{2k}, this will yield a tableau of the form

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

Note that by assumption the matrix C_0 is already known by the $\prec_{\mathrm{in}}$-normal forms of T w.r.t. I_1 and the matrix D_0 is given by the $\prec_{\mathrm{in}}$-normal forms of L w.r.t. I_1. Note also that C_0 and C_1 are again block-Toeplitz of type (m_k, D).

The required matrices C_1 and D_1 can be computed by using the $\prec_{\mathrm{in}}$-multiplication tensor of I_{2k} , enumerating the monomials in $\mathrm{Mon}(\mathbf{x})$ in order of $\prec_{\mathrm{out}}$ 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. $\prec_{\mathrm{in}}$, this can again be done in time $\widetilde{O}\left(\mathrm{m}_k c D^3\right)$. 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

$$\begin{bmatrix} C_0 & \\ C_1 & C_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} D_0 \\ D_1 \end{bmatrix},$$

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 $\widetilde{O}\left(\mathrm{m}_k D^\omega\right)$;
- computing the product $C_0^{-1}(D_1 - C_1 X_0)$.

Note that C_0^{-1} has displacement rank bounded above by $2D + 2$, see [8, 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 $C_0^{-1} v$ and cD matrix-vector products of the form $C_1 v$. Again, thanks to Proposition 4.3, this is done in time $\widetilde{O}\left(\mathrm{m}_k c D^3\right)$. This finally yields the desired complexity. □

The following corollary now 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. *For $k \in \mathbb{N}$, let m_k be the number of monomials in $\mathbf{z}$ up to degree k. Let c be the cardinality of $\mathbf{x}$. 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 $\widetilde{O}\left(\mathrm{m}_{2^\ell} c D^3\right) = \widetilde{O}\left(\mathrm{m}_\delta c D^3\right)$.

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 $\widetilde{O}\left(\mathrm{m}_{2^i} c D^3\right)$ operations. Since $\mathrm{m}_{2^i} = \binom{n-c+2^i}{n-c} = O\left(2^{(n-c)i}\right)$, where n is the total number of variables, and thus $n-c$ the cardinality of $\mathbf{z}$, 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 3.6.

Remark 4.8. *In a follow-up paper, we plan to study the complexity of our algorithm using variants of FGLM, such as [6, 19, 21, 31].*

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

Definition 4.9. Let y be an extra variable and let $I^{\mathrm{hom}} \subset \mathbb{K}[\mathbf{z}, \mathbf{x}, y]$ 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 $\mathbb{K}[\mathbf{z}, \mathbf{x}, y]/I^{\mathrm{hom}}$.

Supposing that I^{hom} is Cohen-Macaulay we now have the following statement, see e.g. [30]. This statement has frequently been used in the complexity analysis of Gröbner basis algorithms.

Lemma 4.10. *Let I be in projective generic position with I^{hom} Cohen-Macaulay. Let H be the reduced $\prec_{\mathrm{drl}}$-Gröbner basis of I (with the variables in $\mathbf{z}$ considered smaller as those in $\mathbf{x}$). Then*

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

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

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

This implies that when I is such that I^{hom} is Cohen-Macaulay and is in projective generic position then we can replace $\prec_{\mathrm{in}}$ with $\prec_{\mathrm{drl}}$ 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 $\{f_1, \ldots, f_c\}$, $\prec_{\mathrm{drl}}$ and the chosen $\prec_{\mathrm{out}}$ in order to compute the the reduced $\prec_{\mathrm{out}}$-Gröbner basis of $I \cdot \mathbb{K}(\mathbf{z})[\mathbf{x}]$ up to precision 2δ. Finally, using Corollaries 4.6 and 4.7, we obtain the desired complexity. □

Remark 4.11. *Let us close this section with a remark on the probability of $\mathfrak{m}$ being a point of good specialization in the situation of*

Theorem 1.1 if $\prec_{\text{out}}=\prec_{\text{lex}}$*, the* lexicographic *order on* $\text{Mon}(\mathbf{x})$*. If* I^{gen} *is in* shape position*, then the reduced* $\prec_{\text{lex}}$*-Gröbner basis of* I^{gen} *is of the form*

$$\{g_c(\mathbf{z}, x_c), x_1 - g_1(\mathbf{z}, x_c), \ldots, x_{c-1} - g_{c-1}(\mathbf{z}, x_c)\}$$

with $g_c(\mathbf{z}, x_c) \in \mathbb{K}[\mathbf{z}, x_c]$ *of total degree* D*. One can then show, using [33], that the degree of the lcm of the denominators of the coefficients of* $g_1, \ldots, g_{c-1}$ *is bounded by* D^2 *where* $D := d_1 \cdots d_c$ *is the Bézout bound of our system. Then, if* $\mathbb{K} = \mathbb{F}_q$ *for a prime power* q*, the Schwartz-Zippel lemma [35] implies that the probability of* $\mathfrak{m}$ *not being a point of good specialization is bounded by* D^2/q *which goes to zero as* q *increases.*

5 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` [36] which itself is written in `Julia` [7]. All required Gröbner basis computations use the Gröbner basis libraries `msolve` [4] via its `Julia`-interface `AlgebraicSolving.jl` or `Groebner.jl` [12], also written in `Julia`. 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. The implementation is available at https://gitlab.lip6.fr/mohr/genfglm. 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 $\mathbf{z} = \{z_1, \ldots, z_{n-c}\}$, choose random $a_1, \ldots, a_{n-c} \in \mathbb{K}$ and make the coordinate substitution $z_i \leftarrow z_i - a_i$.
(4) Choose $\prec_{\text{in}}$ as the block order on $\text{Mon}(\mathbf{z}, \mathbf{x})$ eliminating $\mathbf{x}$ with $\prec_{\text{drl}}$ on both blocks of variables.
(5) If $\mathbf{x} = \{x_1, \ldots, x_c\}$, choose $\prec_{\text{out}}$ as a block order on $\text{Mon}(\mathbf{x})$ eliminating $\mathbf{x}' := \{x_1, \ldots, x_{c-1}\}$ with $\prec_{\text{drl}}$ on $\mathbf{x}'$ and $\prec$ on $\{x_c\}$.
(6) By the elimination property of block orders, the target Gröbner basis G contains a single polynomial g_c in the univariate polynomial ring $\mathbb{K}(\mathbf{z})[x_c]$.
(7) Use Algorithm 2 to compute only the polynomial g_c, ignoring the rest of the set G. Note that this is indeed possible, in line 7 of Algorithm 2 we may choose which of the elements in G_{lift} to actually keep and lift and ignore the rest.

In a certain generic situation (more precisely, when the variable x_c is "generic"), the computed polynomial g_c can be used for a primary decomposition of I, see e.g. [3, Sections 8.6 and 8.7], this motivates our choice of $\prec_{\text{out}}$. We should emphasize however that we did not verify whether this generic situation is met in the examples below. In the context of primary decomposition, it suffices to know just the polynomial g_c in G, the rest of the Gröbner basis can be ignored. This is a potential advantage our algorithm has over the classical way of computing Gröbner bases of generic fibers using elimination orders for which there is no way of getting around computing the entire set G.

Table 1: Benchmarks for Algorithm 2

	Algorithm 2		`msolve` with $\prec_{\text{out}}$
System	Timing (in s)	Memory	Timing (in s)
ED(3,3)	237.9	95.47 GB	43521.43
R1	0.01	140.49 GB	0.01
R2	0.01	251.95 MB	0.01
R3	0.01	248.93 MB	0.01
M2	2.75	1.56 GB	0.03
M3	0.19	410.09 MB	0.01
PS(2,10)	0.8	417.63 MB	0.3
PS(2,12)	44.82	1.38 GB	7.3
Sing(2,10)	0.2	275.17 MB	0.1
SOS(5,4)	1.2	1.2 GB	0.3
SOS(6,4)	11.97	1.07 GB	30.35
SOS(6,5)	22.19	954.41 MB	26.61
RD(3)	4.29	650.98 MB	0.11
RD(4)	33.42	10.46 GB	13.43
RD(5)	729.51	185.87 GB	780.92

We never directly computed the reduced Gröbner basis H of I w.r.t. $\prec_{\text{in}}$, but only the reduced $\prec_{\text{in}}$-Gröbner basis H_u of the ideals $I + \mathfrak{m}_u$. When doing this, we found that the computations were better-behaved when choosing $\prec_{\text{in}}$ as above rather than $\prec_{\text{in}}=\prec_{\text{drl}}$. 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 `msolve` and which, in this case, just runs the F_4 algorithm with a suitable block order on $\text{Mon}(\mathbf{z}, \mathbf{x})$. These timings are given in Table 1.
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 [13];
- R1, R2, R3 come from a problem in Robotics, see [22];
- 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 [15] 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. To illustrate the memory usage of our algorithm, we give in addition the total memory allocated by our implementation.
On most small examples in Table 1, we achieve a comparable timing with `msolve`, with the exception of PS(2,12). On the larger example RD(5) we have a small improvement, while on ED(3,3) we achieve a much better timing. We should mention that our implementation is not yet optimized (for example, we observed that the linear systems to be solved in Algorithm 1 are very sparse, yet our implementation relies on dense representations and arithmetic) and does not incorporate the idea mentioned in Remark 3.2.

Example 5.1. Continuing Example 3.8, we provide a detailed log file with explanatory comments for running our implementation on the Cyclic 8 polynomial system at https://polsys.lip6.fr/~mohr/assets/fglm_log.txt.

REFERENCES

[1] Elizabeth A. Arnold. 2003. Modular Algorithms for Computing Gröbner Bases. *Journal of Symbolic Computation* 35, 4 (2003), 403–419. https://doi.org/10.1016/S0747-7171(02)00140-2

[2] Thomas Becker. 1994. On Gröbner Bases under Specialization. *Applicable Algebra in Engineering, Communication and Computing* 5, 1 (1994), 1–8. https://doi.org/10.1007/BF01196621

[3] Thomas Becker and Volker Weispfenning. 1993. *Gröbner Bases.* Graduate Texts in Mathematics, Vol. 141. Springer-Verlag, New York. https://doi.org/10.1007/978-1-4612-0913-3

[4] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. 2021. msolve: A Library for Solving Polynomial Systems. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation* (Virtual Event, Russian Federation) *(ISSAC '21)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/3452143.3465545

[5] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. 2023. New Efficient Algorithms for Computing Gröbner Bases of Saturation Ideals (F4SAT) and Colon Ideals (Sparse-FGLM-colon). https://doi.org/10.48550/arXiv.2202.13387 arXiv:2202.13387 [cs, math]

[6] Jérémy Berthomieu, Vincent Neiger, and Mohab Safey El Din. 2022. Faster Change of Order Algorithm for Gröbner Bases under Shape and Stability Assumptions. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation (ISSAC '22)*. Association for Computing Machinery, New York, NY, USA, 409–418. https://doi.org/10.1145/3476446.3535484

[7] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM review* 59, 1 (2017), 65–98.

[8] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. 2017. *Algorithmes Efficaces en Calcul Formel.* Frédéric Chyzak (auto-édit.), Palaiseau.

[9] Alin Bostan, Claude-Pierre Jeannerod, Christophe Mouilleron, and Éric Schost. 2017. On Matrices With Displacement Structure: Generalized Operators and Faster Algorithms. *SIAM J. Matrix Anal. Appl.* 38, 3 (2017), 733–775. https://doi.org/10.1137/16M1062855

[10] Bruno Buchberger. 1965. *Ein Algorithmus Zum Auffinden Der Basiselemente Des Restklassenringes Nach Einem Nulldimensionalen Polynomideal.* Ph. D. Dissertation. Universität Innsbruck.

[11] S. Collart, M. Kalkbrener, and D. Mall. 1997. Converting Bases with the Gröbner Walk. *Journal of Symbolic Computation* 24, 3 (1997), 465–469. https://doi.org/10.1006/jsco.1996.0145

[12] Alexander Demin and Shashi Gowda. 2023. Groebner.jl: A Package for Gröbner Bases Computations in Julia. https://doi.org/10.48550/arXiv.2304.06935

[13] Jan Draisma, Emil Horobeţ, Giorgio Ottaviani, Bernd Sturmfels, and Rekha R. Thomas. 2016. The Euclidean distance degree of an algebraic variety. *Found. Comput. Math.* 16, 1 (2016), 99–149. https://doi.org/10.1007/s10208-014-9240-x

[14] Gary L. Ebert. 1983. Some comments on the modular approach to Gröbner-bases. *SIGSAM Bull.* 17, 2 (1983), 28–32. https://doi.org/10.1145/1089330.1089336

[15] Christian Eder, Pierre Lairez, Rafael Mohr, and Mohab Safey El Din. 2023. A Direttissimo Algorithm for Equidimensional Decomposition. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation (ISSAC '23)*. Association for Computing Machinery, New York, NY, USA, 260–269. https://doi.org/10.1145/3597066.3597069

[16] Christian Eder, Pierre Lairez, Rafael Mohr, and Mohab Safey El Din. 2023. A Signature-Based Algorithm for Computing the Nondegenerate Locus of a Polynomial System. *Journal of Symbolic Computation* 119 (2023), 1–21. https://doi.org/10.1016/j.jsc.2023.02.001

[17] Jean-Charles Faugère. 1999. A New Efficient Algorithm for Computing Gröbner Bases (F4). *Journal of Pure and Applied Algebra* 139, 1 (1999), 61–88. https://doi.org/10.1016/S0022-4049(99)00005-5

[18] Jean-Charles Faugère. 2002. 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* (Lille, France) *(ISSAC '02)*. Association for Computing Machinery, New York, NY, USA, 75–83. https://doi.org/10.1145/780506.780516

[19] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. 2014. Sub-Cubic Change of Ordering for Gröbner Basis: A Probabilistic Approach. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC '14)*. Association for Computing Machinery, New York, NY, USA, 170–177. https://doi.org/10.1145/2608628.2608669

[20] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. 1993. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation* 16, 4 (1993), 329–344. https://doi.org/10.1006/jsco.1993.1051

[21] Jean-Charles Faugère and Chenqi Mou. 2017. Sparse FGLM Algorithms. *Journal of Symbolic Computation* 80 (2017), 538–569. https://doi.org/10.1016/j.jsc.2016.07.025

[22] Jorge García Fontán, Abhilash Nayak, Sébastien Briot, and Mohab Safey El Din. 2022. Singularity Analysis for the Perspective-Four and Five-Line Problems. *International Journal of Computer Vision* 130, 4 (2022), 909–932. https://doi.org/10.1007/s11263-021-01567-4

[23] Patrizia Gianni. 1989. Properties of Gröbner Bases under Specializations. In *Eurocal '87 (Lecture Notes in Computer Science)*, James H. Davenport (Ed.). Springer, Berlin, Heidelberg, 293–297. https://doi.org/10.1007/3-540-51517-8_128

[24] Marc Giusti, Grégoire Lecerf, and Bruno Salvy. 2001. A Gröbner Free Alternative for Polynomial System Solving. *Journal of Complexity* 17, 1 (2001), 154–211. https://doi.org/10.1006/jcom.2000.0571

[25] Hans-Gert Gräbe. 1993. On Lucky Primes. *Journal of Symbolic Computation* 15, 2 (1993), 199–209. https://doi.org/10.1006/jsco.1993.1014

[26] Philippe Guillaume and Alain Huard. 2000. Multivariate Padé Approximation. *J. Comput. Appl. Math.* 121, 1 (2000), 197–219. https://doi.org/10.1016/S0377-0427(00)00337-X

[27] Martin Helmer and Vidit Nanda. 2023. Conormal Spaces and Whitney Stratifications. *Foundations of Computational Mathematics* 23, 5 (Oct. 2023), 1745–1780. https://doi.org/10.1007/s10208-022-09574-8 arXiv:2106.14555 [cs, math]

[28] Evelyne Hubert. 2003. Notes on Triangular Sets and Triangulation-Decomposition Algorithms I. In *Symbolic and Numerical Scientific Computation (Lecture Notes in Computer Science)*. Springer, 1–39. https://doi.org/10.1007/3-540-45084-X_1

[29] Michael Kalkbrener. 1989. Solving Systems of Algebraic Equations by Using Gröbner Bases. In *Eurocal '87*, James H. Davenport (Ed.). Springer, Berlin, Heidelberg, 282–292. https://doi.org/10.1007/3-540-51517-8_127

[30] Monique Lejeune-Jalabert. 1986. *Effectivité de calculs polynomiaux. Cours de D.E.A.* Laboratoire de Mathématiques associé au C. N. R. S.

[31] Vincent Neiger and Éric Schost. 2020. Computing Syzygies in Finite Dimension Using Fast Linear Algebra. *Journal of Complexity* 60 (2020), 101502. https://doi.org/10.1016/j.jco.2020.101502

[32] Franz Pauer. 1992. On Lucky Ideals for Gröbner Basis Computations. *Journal of Symbolic Computation* 14, 5 (1992), 471–482. https://doi.org/10.1016/0747-7171(92)90018-Y

[33] Éric Schost. 2003. Computing parametric geometric resolutions. *Appl. Algebra Engrg. Comm. Comput.* 13, 5 (2003), 349–393. https://doi.org/10.1007/s00200-002-0109-x

[34] Éric Schost and Catherine St-Pierre. 2023. P-Adic Algorithm for Bivariate Gröbner Bases. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation (ISSAC '23)*. Association for Computing Machinery, New York, NY, USA, 508–516. https://doi.org/10.1145/3597066.3597086

[35] J. T. Schwartz. 1980. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* 27, 4 (Oct. 1980), 701–717. https://doi.org/10.1145/322217.322225

[36] The OSCAR team. 2022. OSCAR – Open Source Computer Algebra Research System.

[37] Carlo Traverso. 1989. Gröbner Trace Algorithms. In *Symbolic and Algebraic Computation (Lecture Notes in Computer Science)*, P. Gianni (Ed.). Springer, Berlin, Heidelberg, 125–138. https://doi.org/10.1007/3-540-51084-2_12

[38] Carlo Traverso. 1996. Hilbert Functions and the Buchberger Algorithm. *Journal of Symbolic Computation* 22, 4 (1996), 355–376. https://doi.org/10.1006/jsco.1996.0056

[39] Franz Winkler. 1988. A p-adic Approach to the Computation of Gröbner Bases. *Journal of Symbolic Computation* 6, 2 (1988), 287–304. https://doi.org/10.1016/S0747-7171(88)80049-X